Computational Complexity of Strong Admissibility for Abstract Dialectical Frameworks

Keshavarzi Zafarghandi, Atefeh; Dvorak, Wolfgang; Verbrugge, Rineke; Verheij, Bart

Published in:
19 th International Workshop on Non-Monotonic Reasoning

Publication date:
2021

19th International Workshop on Non-Monotonic Reasoning

November 3 – 5, 2021
Hanoi, Vietnam

Leila Amgoud
CNRS – IRIT, France

Richard Booth
Cardiff University, United Kingdom

Preface

NMR is the premier forum for results in the area of non-monotonic reasoning. Its aim is to bring together active researchers in this broad field within knowledge representation and reasoning (KR), including belief revision, uncertain reasoning, reasoning about actions, planning, logic programming, preferences, argumentation, causality, and many other related topics including systems and applications. NMR has a long history - it started in 1984, and has been held every two years since then.

This volume contains the papers accepted for presentation at the 19^{th} edition of the workshop, held virtually on November 3–5, 2021, and collocated with the 18^{th} International Conference on Principles of Knowledge Representation and Reasoning (KR 2021). There were 37 submissions, each of which has been reviewed by two program committee members. The committee has decided to accept 33 papers. The program also includes three invited talks by Nina Gierasimczuk (Technical University of Denmark), Vered Shwartz (University of British Columbia, Canada) and Tran Cao Son (New Mexico State University, USA).

Program Committee

Ofer Arieli, Academic College of Tel-Aviv, Israel
Christoph Beierle, University of Hagen, Germany
Alexander Bochman, Holon Institute of Technology, Israel
James Delgrande, Simon Fraser University, Canada
Juergen Dix, Clausthal University of Technology, Germany
Wolfgang Faber, Alpen-Adria-Universitaet Klagenfurt, Austria
Lluis Godo Lacas, IIIA-CSIC, Spain
Eduardo Fermé, Universidade da Madeira, Portugal
Laura Giordano, Università del Piemonte Orientale, Italy
Aaron Hunter, British Columbia Institute of Technology, Canada
Haythem Ismail, Cairo University and German University of Cairo, Egypt
Tomi Janhunen, Aalto University, Finland
Souhila Kaci, Université Montpellier 2, France
Antonis Kakas, University of Cyprus
Gabriele Kern-Isberner, Technische Universitaet Dortmund, Germany
Sébastien Konieczny, CRIL-CNRS, France
Maria Vanina Martinez, Universidad de Buenos Aires and CONICET, Argentina
Thomas Meyer, University of Cape Town, South Africa
Nir Oren, University of Aberdeen, United Kingdom
Xavier Parent, Vienna University of Technology, Austria
Laurent Perrussel, Université de Toulouse, France
Ramon Pino Perez, Universidad de Los Andes, Venezuela
Sylwia Polberg, Cardiff University, United Kingdom
Ricardo O. Rodriguez, Universidad de Buenos Aires, Argentina
Ken Satoh, National Institute of Informatics and Sokendai, Japan
Gerardo Simari, Universidad Nacional del Sur and CONICET, Argentina
Guillermo R. Simari, Universidad de Sur in Bahia Blanca, Argentina
Christian Strasser, Ruhr-Universitaet Bochum, Germany
Matthias Thimm, Universitaet Koblenz-Landau, Germany
Leon van der Torre, Université du Luxembourg
Ivan Varzinczak, Université d'Artois and CNRS, France
Renata Wassermann, Universidade de Sao Paulo, Brazil
Emil Weydert, University of Luxembourg
Stefan Woltran, Vienna University of Technology, Austria

Additional Reviewers

Fred Freitas (Federal University of Pernambuco, Brazil)
Merna Hatem Shafiee (German University in Cairo, Egypt)
Markus Ulbricht, 2TU Wien, Institute of Logic and Computation
Jonas Haldimann, University of Hagen, Germany
Ammar Abbas Hassan, German University in Cairo, Egypt
Nourhan Ehab, German University in Cairo, Egypt
Zeynep G. Saribatur, Vienna University of Technology, Austria

Contents

Multiple Selective Revision

Fillipe Resina[*] , **Renata Wassermann**
Universidade de São Paulo
{fmresina, renata}@ime.usp.br

Abstract

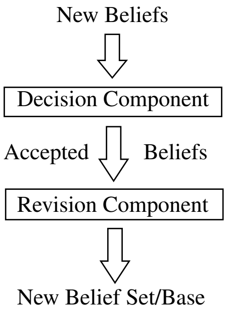
One of AGM revision's main properties is success, which guarantees that new information is always accepted by a rational agent, even when it has to give up a reasonable belief previously held. However, in more realistic scenarios, when dealing with a new belief that contradicts previous ones, an agent has the option to reject it. Selective Revision, then, came up as a third possibility, allowing the agent to accept only part of a new belief. Selective Revision was initially defined for single sentences as inputs but, in many situations, new pieces of information come simultaneously (a set of beliefs). This paper proposes a generalization of Selective Revision to the multiple case for both belief sets (theories) and belief bases. We provide constructions, postulates and representation theorems for different classes of Multiple Selective Revision.

1 Introduction

The belief corpus of an agent is usually not static, and, in this context, a rational agent needs to know how to deal with its dynamics. That is the purpose of the studies in Belief Revision, in which the most used framework is the AGM paradigm (Alchourrón, Gärdenfors, and Makinson 1985). A revision occurs when an agent receives a new piece of information possibly inconsistent with its previous beliefs and has to accommodate the new one consistently while preserving as much of the old beliefs as possible.

One of the AGM revision assumptions is that a new belief is always accepted, a property known as *success*. Revision operations of this kind are classified as *prioritized* revision. However, sometimes the agent should have the option to reject a piece of incoming information, either because of possible low reliability of the new belief (or of its source) or because of strong confidence in the beliefs previously held. That is why the field of *non-prioritized* revision (Hansson 1999a) started to be explored, in which the success property is not guaranteed.

Among the different varieties of non-prioritized revision[1], *Selective Revision* (Fermé and Hansson 1999) came up as a third possibility for the agent, since this operation allows it not only to accept or reject a new belief but also to accept just a part of it, that is, a weakening of the input sentence may be applied. This weakening is performed by a transformation function, to which the incoming information is submitted to perform an evaluation. Then the agent applies a traditional (prioritized) revision of its beliefs by the outcome of that function. The following figure summarizes the general behaviour of selective revision:



As well as AGM revision, selective revision was initially defined for single sentences as inputs. Nevertheless, in many situations, an agent receives not only a single new belief but a set of them and has to make decisions in the face of it, a problem known as *Multiple Revision* (Fuhrmann 1997; Fuhrmann and Hansson 1994). Barber and Kim (2001), for example, state that in the real world, an agent is in contact with several information sources and deals with limited, incomplete, unsure or even wrong knowledge. Therefore, they developed a belief revision process[2] which assesses the reputation of information sources and use it to define the next decision steps. Another example of the importance of multiple contexts is explored in (Pantoja et al. 2016) and (Stabile, Pantoja, and Sichman 2018), in which the authors analyze the application of perception filters in agents. They consider simulation systems and robotic domains and observe that agents may be overwhelmed by unnecessary information without any goal control, thus generating a needless increase in processing time. The more sensors an agent has (to perceive an environment), the more perceptions it has to

[*] Supported by the Brazilian funding agency CAPES

[1] For an overview see (Fermé and Hansson 2018, Chapter 8).

[2] The belief revision process defined by them is a numerical formalism, unlike AGM, which is a logical formalism.

process, which becomes a bottleneck. Hence, using a kind of pre-processing of the information may decrease the cost effects of processing everything and advance an agent's performance. In order to illustrate the idea, consider the following example.

Example 1. *Imagine that three robots (including a coordinator C) are in a house that needs to be cleaned up. Initially, C knows that the bedroom is organized but full of dust, the bathroom is flooded, the kitchen is full of food scraps, but the living room is neat. Before starting the job, robots A and B collected some perceptions. After some time, they jointly report to C. A said that (i) the beds in the bedroom need to be made and that (ii) there is a silver tap in the bathroom that is open. B told that (iii) a dinosaur broke a vase in the living room and that (iv) the kitchen is clean and organized. Before revising its beliefs, C applies a filter, which accepts (i) and (ii), except for the information about the tap's material, as there is no silver in that country. From (iii), the filter accepts that there is a broken vase in the living room but rejects the dinosaur part. (iv) is fully rejected. After that, C performs a prioritized multiple revision of its beliefs.*

Selective revision was also initially defined for sets of sentences closed under logical consequence, known as *belief sets* or *theories*. Due to their usually infinite nature, they are more suitable for idealized agents and, as a consequence, challenging to be handled computationally. As an alternative, one can represent knowledge using *belief bases* - sets of sentences not necessarily closed, an approach closer to realistic scenarios.

This article proposes a generalization of Selective Revision to the multiple change context for both belief sets and belief bases. We provide constructions, postulates for the operators, properties for the transformation function and representation theorems to link everything for different classes of *Multiple Selective Revision*. The model of belief bases considered in this work is the one defined by Hansson (1991).

It is essential to observe that, in a multiple-revision context, differently from the case of singleton inputs, the partial acceptance characteristic of selective revision can have two different meanings: either the simple choice of a subset of the input set or the logical weakening of a chosen subset (from the input). We are going to address both cases.

This paper proceeds as follows. Section 2 provides the necessary background. Multiple Selective Revision for belief sets is presented in Section 3, while for belief bases is given in Section 4. Analysis of related work comes in Section 5 and conclusion and future work come in Section 6.

2 Background

In this section, we briefly present the necessary background on selective and multiple revision.

2.1 Formal Preliminaries

We will assume that a logic is a language $\mathcal{L}$ provided with a consequence operator Cn. $\mathcal{L}$ contains all the available sentences of the logic. Cn is a function that maps sets of sentences to sets of sentences and that satisfies the standard Tarskian axioms, namely *iteration, inclusion* and *monotony*, and also *compactness*. For $A, B \subseteq \mathcal{L}$, we say that A implies B iff $B \subseteq Cn(A)$. We will sometimes use $K \vdash \alpha$ for $\alpha \in Cn(K)$, $\vdash \alpha$ for $\alpha \in Cn(\emptyset)$, $K \not\vdash \alpha$ for $\alpha \notin Cn(K)$ and $\not\vdash \alpha$ for $\alpha \notin Cn(\emptyset)$. For formulas and sentences, we will use lowercase greek letters (such as α, β). For sets of sentences, uppercase Latin letters (such as A, B, C, K). $\perp$ is the falsity constant, and $K_\perp$ is the inconsistent belief set.

2.2 AGM Revision

The revision of a belief set aims to absorb a new belief in that set. In a revision mechanism, some previous beliefs may be given up in order to achieve, as result, a consistent belief set. The postulates below are usually known as the basic AGM postulates for revision:

(closure) $K * \alpha$ is a belief set

(success) $K * \alpha \vdash \alpha$

(inclusion) $K * \alpha \subseteq Cn(K \cup \{\alpha\})$

(consistency) If $\nvdash \neg\alpha$ then $K * \alpha \neq K_\perp$

(vacuity) If $K \nvdash \neg\alpha$, then $Cn(K \cup \{\alpha\}) \subseteq K * \alpha$

(extensionality) If $\vdash \alpha \leftrightarrow \beta$, then $K * \alpha = K * \beta$

The authors also provided constructions and a representation theorem. For more details, see (Alchourrón, Gärdenfors, and Makinson 1985).

2.3 Selective Revision

Here we present Selective Revision for singleton inputs in its two variants: belief sets and belief bases. Both of them were axiomatically characterized in their respective works.

Selective Theory Revision Fermé and Hansson (1999) introduced a new operator named selective revision to deal with the partial acceptance of new information.

Among the six basic AGM revision postulates, four are also plausible for selective revision: *closure*, *inclusion*, *consistency* and *extensionality*. Three new postulates were proposed - two weakening versions of success and one to control minimality of change:

(proxy success) There is a sentence β, such that $K \circ \alpha \vdash \beta$, $\vdash \alpha \rightarrow \beta$ and $K \circ \alpha = K \circ \beta$

(weak proxy success) There is a sentence β, such that $K \circ \alpha \vdash \beta$ and $K \circ \alpha = K \circ \beta$

(consistent expansion) If $K \nsubseteq K \circ \alpha$, then $K \cup (K \circ \alpha) \vdash \perp$

Selective theory revision is defined as follows:

Definition 1. *(Fermé and Hansson 1999) Let K be a belief set, $*$ a basic AGM revision operator for K and f a function from $\mathcal{L}$ to $\mathcal{L}$. The selective revision $\circ$ based on $*$ and f is the operation such that for all sentences α: $K \circ \alpha = K * f(\alpha)$. f is called the transformation function on which $\circ$ is based.*

Roughly speaking, a transformation function f selects the reliable part of every sentence. A natural constraint is that $f(\alpha)$ should not return more information than what is expressed in α (*i.e.*, $\vdash \alpha \rightarrow f(\alpha)$). Nevertheless, it is possible to apply a function without this constraint. Some of the proposed properties for transformation functions are:

(implication) $\vdash \alpha \rightarrow f(\alpha)$

(weak implication) If $K \nvdash \neg\alpha$, then $\vdash \alpha \rightarrow f(\alpha)$

(idempotence) $\vdash f(f(\alpha)) \leftrightarrow f(\alpha)$

(extensionality) If $\vdash \alpha \leftrightarrow \beta$, then $\vdash f(\alpha) \leftrightarrow f(\beta)$

(consistency preservation) If $\nvdash \neg\alpha$, then $\nvdash \neg f(\alpha)$

(weak maximality) If $K \nvdash \neg\alpha$, then $\vdash f(\alpha) \leftrightarrow \alpha$

Fermé and Hansson (1999) provided representation theorems for three classes of selective theory revision. What distinguishes the three classes are the properties required for the transformation f, such as f being idempotent or being a proper weakening in the logical sense (satisfying implication).

Selective Base Revision Resina *et al.* (2020) extended selective revision to belief bases. Some postulates were adapted from the ones for belief sets and others were newly proposed:

(consistency) If $\alpha \nvdash \bot$, then $B \circledast \alpha \nvdash \bot$

(vacuity) If $B \nvdash \neg\alpha$, then $B \cup \{\alpha\} \subseteq B \circledast \alpha$

(proxy success) There is a sentence β, such that $\beta \in B \circledast \alpha$, $\vdash \alpha \rightarrow \beta$ and $B \circledast \alpha = B \circledast \beta$

(weak proxy success) There is a sentence $\beta \in B \circledast \alpha$ and $B \circledast \alpha = B \circledast \beta$

(stability) If $\alpha \in B$, then $\alpha \in B \circledast \alpha$

(uniform success) If for all subsets $B' \subseteq B$, $B' \cup \{\alpha\} \vdash \bot$ iff $B' \cup \{\beta\} \vdash \bot$, then $\alpha \in B \circledast \alpha$ iff $\beta \in B \circledast \beta$

(weak inclusion) If $\alpha \in B \circledast \alpha$, then $B \circledast \alpha \subseteq B \cup \{\alpha\}$

(conditional uniformity) If $\alpha \in B \circledast \alpha$ and for all subsets B' of B it holds that $B' \cup \{\alpha\} \vdash \bot$ iff $B' \cup \{\beta\} \vdash \bot$, then $B \cap (B \circledast \alpha) = B \cap (B \circledast \beta)$

(weak relevance) If $\alpha \in B \circledast \alpha$, $\beta \in B$ and $\beta \notin B \circledast \alpha$, then there is some B' such that $B \circledast \alpha \subseteq B' \subseteq B \cup \{\alpha\}$, $B' \nvdash \bot$ but $B' \cup \{\beta\} \vdash \bot$

Differently from Definition 1, the revision operator $*$ applied in the definition was one for bases (Hansson 1999b). New and adapted properties for f were also defined:

(idempotence) $f(f(\alpha)) = f(\alpha)$

(weak maximality) If $A \nvdash \neg\alpha$, then $f(\alpha) = \alpha$

(lower boundary) If $\alpha \in A$, then $f(\alpha) = \alpha$

(uniform identity) If for all $A' \subseteq A$, $A' \cup \{\alpha\} \vdash \bot$ iff $A' \cup \{\beta\} \vdash \bot$, then $f(\alpha) = \alpha$ iff $f(\beta) = \beta$

After the definition of two kinds of construction, four representation theorems were obtained for two classes of selective base revision (Resina et al. 2020).

2.4 Multiple Revision

In this section, we will present two variants of multiple revision[3], one for theories and one for belief bases.

Multiple Theory Revision Fuhrmann (1988; 1997) generalized the AGM revision operation for dealing with sets instead of a single formula as input:

Definition 2. *(Fuhrmann 1997) An operator $*_p$ is called a multiple theory revision iff $*_p$ satisfies*

(closure) *$K *_p A$ is a belief set*

(success) *$A \subseteq K *_p A$*

(inclusion) *$K *_p A \subseteq Cn(K \cup A)$*

(weak consistency) *If $A \nvdash \bot$ then $K *_p A \nvdash \bot$*

(relevance) *If $\beta \in K$ but $\beta \notin K *_p A$, then there is some K' such that $(K *_p A) \cap K \subseteq K' \subseteq K$, $K' \cup A \nvdash \bot$ but $K' \cup \{\beta\} \cup A \vdash \bot$.*

(extensionality) *If A and B are pairwise equivalent[4], then $K *_p A = K *_p B$*

Observation 1. *If an operator $*_p$ for a belief set K satisfies success, inclusion and relevance, then it satisfies*

(vacuity) *If $K \cup A \nvdash \bot$ then $K *_p A = Cn(K \cup A)$*

Proof. This proof can be straightforwardly adapted from the version for sentence revision in (Hansson 1999b). □

Multiple Base Revision Hansson (1993) also generalized base revision to multiple revision of belief bases:

Definition 3. *(Hansson 1993) An operator $*_p$ is called a multiple base revision if and only if $*_p$ satisfies*

(inclusion) *$B *_p A \subseteq B \cup A$.*

(success) *$A \subseteq B *_p A$.*

(weak consistency) *If $A \nvdash \bot$ then $B *_p A \nvdash \bot$.*

(uniformity) *If, for all subsets B' of B, $B' \cup A \vdash \bot$ iff $(B' \cup C) \vdash \bot$, then $B \cap (B *_p A) = B \cap (B *_p C)$.*

(relevance) *If $\beta \in B \setminus (B *_p A)$ then there is a set B' such that $B *_p A \subseteq B' \subseteq (B \cup A)$, $B' \nvdash \bot$ but $B' \cup \{\beta\} \vdash \bot$.*

Fallapa *et al.* (2012) defined two different constructions for multiple base revision and provided representation theorems for both.

3 Multiple Selective Theory Revision

In this section, we will show how to define and axiomatically characterize multiple selective revision for belief sets (theories).

3.1 Properties

Some of the postulates for Multiple Theory Revision (Section 2.4) remain the same: closure, inclusion, weak consistency, extensionality and vacuity. However, vacuity is a questionable postulate because, although intuitive in some sense, the agent may decide not to accept parts of the incoming beliefs.

As we already discussed, due to the non-prioritized nature of Selective Revision the success postulate is not suitable in this context. So we generalized the two weaker versions for success presented in (Fermé and Hansson 1999) in order to

[3] In the literature, this kind of multiple revision is also known as *package* revision, in which the whole input set is incorporated.

[4] Two sets of sentences A and B are pairwise equivalent (modulo Cn) just in case: $\forall\alpha \in A : \exists\beta \in B$ s.t. $Cn(\alpha) = Cn(\beta)$ and $\forall\beta \in B : \exists\alpha \in A$ s.t. $Cn(\beta) = Cn(\alpha)$.

consider sets of sentences as input. The same was done for consistent expansion and a weaker version of relevance.

Let K be a belief set, A and C be sets of sentences and $\odot$ be a binary selective revision operator that takes a belief set and a set of sentences as input. We propose the following reasonable postulates for multiple selective theory revision:

(choice success) There is a set B such that $B \subseteq Cn(K \odot A)$, $B \subseteq A$ and $K \odot A = K \odot B$

(proxy success) There is a set B such that $B \subseteq Cn(K \odot A)$, $B \subseteq Cn(A)$ and $K \odot A = K \odot B$

(weak proxy success) There is a set B such that $B \subseteq Cn(K \odot A)$ and $K \odot A = K \odot B$

(stability) If $A \subseteq K$, then $A \subseteq K \odot A$.

(consistency) $K \odot A \nvdash \bot$

(weak consistency) If $A \nvdash \bot$ then $K \odot A \nvdash \bot$

(consistent expansion) If $K \nsubseteq K \odot A$ then $K \cup (K \odot A) \vdash \bot$

(weak relevance) If $A \subseteq K \odot A$, $\beta \in K$ and $\beta \notin K \odot A$, then there is some K' such that $(K \odot A) \cap K \subseteq K' \subseteq K$, $K' \cup A \nvdash \bot$ but $K' \cup \{\beta\} \cup A \vdash \bot$.

Choice success states that the selective revision should incorporate a subset of the input set. *Proxy success* establishes that the selective revision should accept some of the input's logical consequences, while *weak proxy success* is a weaker version of it. *Stability* brings that if the input set is already part of the agent's beliefs, it should be kept by the selective revision. *Consistency* guarantees an always consistent result, while *weak consistency* demands a consistent input for that. *Consistent expansion* and *weak relevance* express the idea that nothing is given up from the original set unless it leads the new belief set to consistency. Except for *weak consistency* and *consistent expansion* (already presented in (Krümpelmann et al. 2011)), the other postulates are new.

3.2 Constructing the Operation

Definition 4. *Let K be a belief set, $*_p$ a multiple theory revision for K and f a function from $2^{\mathcal{L}}$ to $2^{\mathcal{L}}$. The* multiple selective theory revision $\odot$*, based on $*_p$ and f, is the operation such that for all sets A: $K \odot A = K *_p f(A)$. f is the* transformation function *on which $\odot$ is based.*

Selective theory revision becomes, then, a particular case of this new operator $\odot$. One can question why a multiple revision operator is needed to construct the operation, perhaps suggesting a sequence of singleton input revisions by the elements of $f(A)$. Nonetheless, multiple revision is different from iterated revision (Darwiche and Pearl 1997) as the sequence in which you process the sentences can result in different outcomes. Thus, we want here to treat all the sentences with equal priority, processing them simultaneously.

The following is a list of properties that the transformation function may satisfy:

(choice) $f(A) \subseteq A$

(implication) $f(A) \subseteq Cn(A)$

(weak implication) If $K \cup A \nvdash \bot$, then $f(A) \subseteq Cn(A)$

(lower boundary) if $A \subseteq K$, then $Cn(f(A)) = Cn(A)$

(idempotence) $Cn(f(f(A))) = Cn(f(A))$

(consistency preservation) If $A \nvdash \bot$, then $f(A) \nvdash \bot$

(consistency) $f(A) \nvdash \bot$

(maximality) $Cn(f(A)) = Cn(A)$

(weak maximality) If $K \cup A \nvdash \bot$, then $Cn(f(A)) = Cn(A)$

(extensionality) If A and B are pairwise equivalent, then $f(A)$ and $f(B)$ are pairwise equivalent.

Choice sets the transformation function to simply choose a subset of the input set A. *Implication* allows the function to choose from the logical consequences of the input, while *weak implication* restricts to the the consequences of the input only if the input is consistent with the previous beliefs. *Lower boundary* states that if an input A is already part of the previous beliefs, then $f(A)$ and A have the same consequences. While *consistency preservation* demands consistency from $f(A)$ only if A is consistent, *consistency* always guarantees a consistent $f(A)$. *Maximality* states that $f(A)$ and A are logically equivalent, while *weak maximality* states a precondition for that. Finally, *extensionality* guarantees a coherent behavior of f when different inputs are pairwise equivalent. *Choice, consistency preservation, consistency* and *extensionality* had already been suggested in (Krümpelmann et al. 2011). The observation below establishes some links between the properties for f and the postulates for $\odot$:

Observation 2. *Let K be a belief set in a language $\mathcal{L}$, $*_p$ be a multiple theory revision operator for K that satisfies the six postulates referred in Definition 2, and f be a transformation function. Let $\odot$ be the multiple selective revision function on K based on $*_p$ and f. Then $\odot$ satisfies closure and consistent expansion. In addition, if f satisfies:*

1. *weak implication then $\odot$ satisfies inclusion.*
2. *consistency then $\odot$ satisfies consistency.*
3. *maximality then $\odot$ satisfies success.*
4. *implication then $\odot$ satisfies weak consistency.*

Proof. 1. We prove by cases: (a) If $K \cup A \vdash \bot$, then $Cn(K \cup A) = K_{\bot}$ and, therefore, $K \odot A \subseteq Cn(K \cup A)$. (b) If $K \cup A \nvdash \bot$, then $K \odot A = K *_p f(A)$ and, by $*_p$-inclusion, $K *_p f(A) \subseteq Cn(K \cup f(A))$. By weak implication we have that $Cn(K \cup f(A)) \subseteq Cn(K \cup A)$. Hence, $K \odot A \subseteq Cn(K \cup A)$.

2. Since $f(A) \nvdash \bot$, by $*_p$-*weak consistency* we have that $K *_p f(A) \nvdash \bot$. Thus, $K \odot A \nvdash \bot$.
3. Trivial, since by definition $*_p$ satisfies success and by maximality $K \odot A = K *_p f(A) = K *_p A$.
4. If $A \nvdash \bot$ then $f(A) \nvdash \bot$ and, by $*_p$-*weak consistency*, $K \odot A = K *_p f(A) \nvdash \bot$.

□

The following representation theorems have been obtained for three classes of multiple selective theory revision functions. Once more, the differences between them lie on the properties of the transformation function (which directly influences the properties of the operation.

Theorem 1. *Let $\mathcal{L}$ be a finite language, K a belief set in $\mathcal{L}$ and $\odot$ an operator on K. The following conditions are equivalent:*

1. *$\odot$ satisfies closure, inclusion, vacuity, weak consistency, extensionality, stability, weak relevance and weak proxy success.*
2. *There exists a multiple theory revision $*_p$ for K that satisfies the six postulates referred in Definition 2, and a transformation function f that satisfies extensionality, lower boundary, consistency preservation, weak maximality and idempotence, such that $K \odot A = K *_p f(A)$ for all A.*

Proof. (1) *implies* (2): we first define f and $*_p$:

$$f(A) = \begin{cases} A & \text{if } K \cup A \nvdash \bot \text{ or } A \subseteq K \odot A; \\ r(A) & \text{otherwise, where } r \text{ is a (well-defined) function from } 2^{\mathcal{L}} \text{ to } 2^{\mathcal{L}} \text{ such that } r(A) \subseteq Cn(K \odot A),\ K \odot A = K \odot r(A) \text{ and for all } A \text{ and } A' \text{ such that } K \odot A = K \odot A',\ r(A) = r(A'). \end{cases}$$

This definition is possible since $\odot$ satisfies *weak proxy success*.

$$K{*}A = \begin{cases} K \odot A & \text{if } A \subseteq Cn(K \odot A); \\ K *'_p A & \text{otherwise, where } *'_p \text{ is any operation that satisfies the six axioms from Definition 2.} \end{cases}$$

We need to show that:

(a) f is a (well-defined) transformation function;

(b) f satisfies the properties;

(c) $*$ is a multiple theory revision (see Definition 2);

(d) $K \odot A = K * f(A)$, for all A.

The proofs are given below:

(a) To prove that f is a (well defined) function we must show that for all $A \subseteq \mathcal{L}$ there exists $A' \subseteq \mathcal{L}$ such that $f(A) = A'$ and that, if $A_1 = A_2$, then $f(A_1) = f(A_2)$.
Let $A \subseteq \mathcal{L}$. If $K \cup A \nvdash \bot$ or $A \subseteq K \odot A$, then $f(A) = A$. Otherwise, $f(A) = r(A) = A'$, for some A' such that $A' \subseteq K \odot A = K \odot A'$. Such A' exists since $\odot$ satisfies *weak proxy success* and *closure*. Assume now that $A_1 = A_2$. If $K \cup A_1 \nvdash \bot$, then $K \cup A_2 \nvdash \bot$, or if $A_1 \subseteq K \odot A_1$, then $A_2 \subseteq K \odot A_2$. Thus, in both cases $f(A_1) = A_1 = A_2 = f(A_2)$. If $K \cup A_1 \vdash \bot$ and $A_1 \nsubseteq K \odot A_1$, then $K \cup A_2 \vdash \bot$ and $A_2 \nsubseteq K \odot A_2$. Thus, $f(A_1) = r(A_1)$ and $f(A_2) = r(A_2)$. r is a (well-defined) function. Hence, from $A_1 = A_2$ it follows that $r(A_1) = r(A_2)$. Therefore, $f(A_1) = f(A_2)$.

(b) That f satisfies *weak maximality* follows from the definition of f. To show that f satisfies *consistency preservation*, let $A \nvdash \bot$; if $K \cup A \nvdash \bot$ or $A \subseteq K \odot A$, then $f(A) = A$ and $f(A) \nvdash \bot$; otherwise, then $f(A) = r(A)$ and $r(A) \subseteq Cn(K \odot A)$; since $A \nvdash \bot$, by $\odot$-*consistency* $K \odot A \nvdash \bot$, which implies that $r(A) \nvdash \bot$ and, finally, that $f(A) \nvdash \bot$. To show that f satisfies *extensionality* suppose that A and B are pairwise equivalent belief sets; if $K \cup A \nvdash \bot$ then $K \cup B \nvdash \bot$, or if $A \subseteq K \odot A$ then $B \subseteq K \odot B$ and in both cases we have that $f(A) = A$ and $f(B) = B$; hence, $f(A)$ and $f(B)$ are pairwise equivalent. If $K \cup A \vdash \bot$ and $A \nsubseteq K \odot A$, then $f(A) = r(A)$, $K \cup B \vdash \bot$ and $B \nsubseteq K \odot B$. Then $f(B) = r(B)$. By $\odot$-*extensionality*, $K \odot A = K \odot B$ and from the definition of r it follows that $r(A) = r(B)$. Therefore, $f(A)$ and $f(B)$ are pairwise equivalent. To show that f satisfies *lower boundary*, assume that $A \subseteq K$. From $\odot$-*stability* it follows that $A \subseteq K \odot A$ and, by the definition of f, $f(A) = A$. Finally, we show that f satisfies *idempotence*. If $K \cup A \nvdash \bot$ or $A \subseteq K \odot A$ then, from the definition of f, $f(f(A)) = f(A)$ follows directly. Otherwise, $f(A) = r(A)$ and, by the definition of r, $r(A) \subseteq K \odot r(A)$. From the definition of f, since $r(A) \subseteq K \odot r(A)$, we have that $f(r(A)) = r(A)$ and, given that $f(A) = r(A)$, we have that $f(f(A)) = f(r(A) = r(A) = f(A)$.

(c) In order to show that $*$ is a multiple theory revision, we need to prove that it satisfies the six axioms from Definition 2. That $*$ satisfies closure, inclusion, extensionality and weak consistency is trivial, since both $\odot$ and $*'$ satisfy these four postulates. That $*$ satisfies success also follows directly from the definition. In order to show that $*$ satisfies relevance, if $A \nsubseteq Cn(K \odot A)$, we are done (given that $*'_p$ satisfies relevance). If $A \subseteq Cn(K \odot A)$, then $K * A = K \odot A$. Suppose that $\exists \beta \in K$ such that $\beta \notin K \odot A$. By $\odot$-*vacuity*, $K \cup A \vdash \bot$. By $\odot$-*weak relevance*, $\exists K'$ such that $(K \odot A) \cap K \subseteq K' \subseteq K$, $K' \cup A \nvdash \bot$ and $K' \cup \{\beta\} \cup A \vdash \bot$. Therefore, as $K * A = K \odot A$, we can conclude that *relevance* is satisfied.

(d) We need to prove that $K \odot A = K * f(A)$. If $K \cup A \nvdash \bot$ or $A \subseteq K \odot A$, $f(A) = A$ and $K \odot f(A) = K \odot A$. In the case of $K \cup A \nvdash \bot$, by $\odot$-*vacuity* it follows that $A \subseteq K \odot A$ and, then, $f(A) \subseteq K \odot f(A)$. By the definition of $*$, $K * f(A) = K \odot f(A)$. Hence, $K * f(A) = K \odot A$. If $K \cup A \vdash \bot$ and $A \nsubseteq K \odot A$ then it follows from the definitions of f and r that $f(A) \subseteq Cn(K \odot A)$ and $K \odot A = K \odot f(A)$, from which follows that $f(A) \subseteq Cn(K \odot f(A))$. Then, from the definition of $*$ it follows that $K * f(A) = K \odot f(A) = K \odot A$.

(2) implies (1): That $\odot$ satisfies closure is trivial since by Definition 2 $*_p$ satisfies closure.

In order to prove *extensionality*, let A and B be pairwise equivalent. Then, by f-*extensionality*, $f(A)$ and $f(B)$ are pairwise equivalent and, by $*_p$-*extensionality*, $K *_p f(A) = K *_p f(B)$ or, equivalently, $K \odot A = K \odot B$.

In order to prove *inclusion*, weak maximality implies weak implication; then, inclusion follows from item 1 of Observation 2.

For vacuity, suppose that $K \cup A \nvdash \bot$. Then, by f-*weak maximality*, $Cn(f(A)) = Cn(A)$. Since $K \odot A = K *_p f(A)$, by $*_p$-*success* $f(A) \subseteq K *_p f(A)$ and, by $*_p$-*closure*, $Cn(f(A)) \subseteq K *_p f(A)$. Since $Cn(f(A)) = Cn(A)$, we also have that $Cn(A) \subseteq K *_p f(A)$. From $K \cup A \nvdash \bot$ and

$Cn(f(A)) = Cn(A)$, we have that $K \cup f(A) \nvdash \bot$. Thus, by $*_p$-*vacuity*, $K *_p f(A) = Cn(K \cup f(A)) = Cn(K \cup A)$. Therefore, $K \odot A = Cn(K \cup A)$.

For weak consistency, let $A \nvdash \bot$. Then, by consistency preservation we have that $f(A) \nvdash \bot$ and, by $*_p$-*weak consistency*, we have that $K *_p f(A) \nvdash \bot$. Thus, $K \odot A \nvdash \bot$.

In order to prove weak proxy success, we have that by Definition 4 and idempotence, $K \odot A = K *_p f(A) = K *_p f(f(A)) = K \odot f(A)$. We therefore have $f(A) \subseteq Cn(K *_p f(A))$, $K \odot A = K \odot f(A)$ and $f(A) \subseteq Cn(K \odot A)$, which is sufficient to prove that $\odot$ satisfies weak proxy success.

For stability, let $A \subseteq K$. Then, since f satisfies lower boundary we have that $Cn(f(A)) = Cn(A)$. Since $K \odot A = K *_p f(A)$, by $*_p$-*extensionality* $K *_p f(A) = K *_p A$. Hence $K \odot A = K *_p A$. From $*_p$-*success* it follows that $A \subseteq K *_p A = K \odot A$.

It only remains to prove that $\odot$ satisfies *weak relevance*. Suppose that $A \subseteq K \odot A$ and $\exists \beta \in K$ such that $\beta \notin K \odot A$. As $K \odot A = K *_p f(A)$, we have that $\beta \notin K *_p f(A)$. By $*_p$-*relevance*, $\exists K'$ such that $(K *_p f(A)) \cap K \subseteq K' \subseteq K$, $K' \cup f(A) \nvdash \bot$ but $K' \cup \{\beta\} \cup f(A) \vdash \bot$. Since $K \odot A = K *_p f(A)$, $(K \odot A) \cap K \subseteq K' \subseteq K$. So, it remains to prove that $K' \cup A \nvdash \bot$. Given that $A \subseteq K \odot A$, then $A \subseteq K *_p f(A)$ and, by $*_p$-*inclusion* we have three possibilities. If $A \subseteq K$, by f-*lower boundary* we have that $Cn(f(A)) = Cn(A)$, and also if $A \subseteq f(A)$. If $((K \odot A) \cap K) \cap A \neq \emptyset \neq A \cap f(A)$, we can conclude that, since $(K \odot A) \cap K \subseteq K'$, $A \subseteq K' \cup f(A)$. Therefore, K' is such that $K' \cup A \nvdash \bot$ but $K' \cup \{\beta\} \cup A \vdash \bot$. Thus, $\odot$-*weak relevance* is satisfied. □

Theorem 2. *Let $\mathcal{L}$ be a finite language, K a belief set in $\mathcal{L}$ and $\odot$ an operator on K. Then the following conditions are equivalent:*

1. *$\odot$ satisfies closure, inclusion, vacuity, weak consistency, extensionality, stability, weak relevance and proxy success.*
2. *There exists a multiple theory revision $*_p$ for K that satisfies the six postulates from Definition 2, and a transformation function f that satisfies extensionality, lower boundary, consistency preservation, weak maximality, idempotence and implication, such that $K \odot A = K *_p f(A)$ for all A.*

Proof. This proof is quite similar to that of Theorem 1. To show that (1) implies (2), we define f to be a function like that of the previous proof but with an additional restriction when $K \cup A \vdash \bot$ and $A \nsubseteq K \odot A$: $f(A) \subseteq Cn(A)$. The existence of such a function follows from proxy success. The proofs for f are essentially the same, and the implication property follows trivially. To show that (2) implies (1) we only have to add a proof for proxy success, which we obtain from Theorem 1 and f-*implication*. □

Theorem 3. *Let $\mathcal{L}$ be a finite language, K a belief set in $\mathcal{L}$ and $\odot$ an operator on K. Then the following conditions are equivalent:*

1. *$\odot$ satisfies closure, weak inclusion, vacuity, weak consistency, extensionality, stability, weak relevance and choice success.*
2. *There exists a multiple theory revision $*_p$ for K that satisfies the six postulates from Definition 2, and a transformation function f that satisfies extensionality, lower boundary, consistency preservation, weak maximality, idempotence and choice, such that $K \odot A = K *_p f(A)$ for all A.*

Proof. This proof is quite similar to that of Theorem 1. To show that (1) implies (2), we define f to be a function like that of the previous proof but with an additional restriction when $K \cup A \vdash \bot$ and $A \nsubseteq K \odot A$: $f(A) \subseteq A$. The existence of such a function follows from choice success. The proofs for f are essentially the same, and the choice property follows trivially. To show that (2) implies (1) we only have to add a proof of choice success, which we obtain from Theorem 1 and f-*choice*. □

Theorem 1 embraces very general operations which do not demand $f(A)$ to be derived from A. Theorem 2 looks at the operations in which $f(A)$ does not return more information than what is expressed in A. Finally, Theorem 3, although more restrictive, represents the most intuitive procedure when $f(A)$ selects a subset of A.

4 Multiple Selective Base Revision

As an alternative to the previous section's approach, we now show how to define and characterize axiomatically multiple selective revision for belief bases.

4.1 Postulates

In comparison to what was defined for belief sets, we have the exclusion of the closure postulate, an exchange of extensionality for uniformity, adaptations in inclusion and vacuity (removing the logical closure) and new versions for the success and inclusion postulates (due to the context of belief bases). From Multiple Base Revision (Section 2.4), inclusion, vacuity 1 and weak consistency remain the same.

Let B be a belief set, A and C be sets of sentences and $\odot$ be a binary selective revision operator. We bring the following reasonable postulates for multiple selective base revision:

(choice success) There is a set C such that $C \subseteq B \odot A$, $C \subseteq A$ and $B \odot A = B \odot C$.

(proxy success) There is a set C such that $C \subseteq B \odot A$, $C \subseteq Cn(A)$ and $B \odot A = B \odot C$.

(weak proxy success) There is a set C such that $C \subseteq B \odot A$ and $B \odot A = B \odot C$.

(conditional success) If $A \backslash B \subseteq B \odot A$, then $A \subseteq B \odot A$.

(uniform success) If for all subsets $B' \subseteq B$, $B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$, then $A \subseteq B \odot A$ iff $C \subseteq B \odot C$.

(weak inclusion) $B \odot A \subseteq B \cup Cn(A)$

(very weak inclusion) $B \odot A \subseteq Cn(B \cup A)$

(conditional inclusion) If $A \subseteq B \odot A$, then $B \odot A \subseteq B \cup A$.

(vacuity 2) If $B \cup A \nvdash \bot$, then $B \odot A = B \cup A$

(consistency) $B \odot A \nvdash \bot$

(weak relevance) If $A \subseteq B \odot A$, $\beta \in B$ and $\beta \notin B \odot A$, then there is some B' such that $B \odot A \subseteq B' \subseteq B \cup A$, $B' \nvdash \bot$ but $B' \cup \{\beta\} \vdash \bot$.

(conditional uniformity) If $A \subseteq B \odot A$ and for all subsets B' of B it holds that $B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$, then $B \cap (B \odot A) = B \cap (B \odot C)$.

The last two postulates are weakened by establishing the precondition $A \subseteq B \odot A$.

Conditional success guarantees that if the difference between A and B is part of the final result, then so is the whole A, which means that if $A \setminus B$ is accepted the intersection between them will not be rejected. *Uniform success* says that if two sets are inconsistent with exactly the same subsets of B, then one of them should be absorbed in the selective revision by it iff the same happens to the other one. *Weak inclusion* shows that the selective revision of B by A is contained in the union of B and the logical consequences of A, while *very weak inclusion* says that it is contained in the logical consequences of the union of B and A. *Conditional inclusion* is a weakening of the traditional inclusion postulate for multiple by preconditioning A in the final result. For *conditional uniformity*, if A is in the outcome and two consistent sets are inconsistent with the same subsets of the original base, then the respective retained sentences of B should be identical. The other postulates' intuition is the same for Multiple Selective Theory Revision (Section 3.1).

Still about the rationale of *inclusion, weak inclusion* and *very weak inclusion*, it is possible to associate them to some of the success postulates. *Inclusion* makes sense when an agent simply chooses a subset of the input set, which links it to *choice success*. *Weak inclusion* represents the possibility for an agent to weaken a subset of the input, which links it to *proxy success*. Finally, *very weak inclusion* is related to a very general context in which the decision of the agent in relation to the input set is not restricted to the set itself, which links the postulate to *weak proxy success*.

Except for the new postulates *choice success, conditional success, weak inclusion* and *very weak inclusion*, the other ones are straightforward generalizations of postulates for Selective Base Revision (Resina et al. 2020).

4.2 Constructing the Operation

Definition 5. *Let B be a belief base, $*_p$ be multiple base revision for B and f be a function from $2^{\mathcal{L}}$ to $2^{\mathcal{L}}$. The* multiple selective base revision $\odot$*, based on $*_p$ and f, is the operation such that for all sets A: $B \odot A = B *_p f(A)$. f is the* transformation function *on which $\odot$ is based.*

Similarly to the previous section, selective base revision becomes a particular case of this new operator $\odot$.

From Multiple Selective Theory Revision, some potential properties for f remain the same: *choice, implication, weak implication, consistency* and *consistency preservation*. Some others needed to be adapted for belief bases. Extensionality was substituted by another with similar intuition.

(maximality) $f(A) = A$

(weak maximality) If $B \cup A \nvdash \bot$, then $f(A) = A$

(conditional maximality) if $A \setminus B \subseteq f(A)$, then $f(A) = A$

(idempotence) $f(f(A)) = f(A)$

(uniform identity) if for all $B' \subseteq B, B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$, then $f(A) = A$ iff $f(C) = C$.

Uniform identity is a version of uniformity for f. *Conditional maximality* states that if the difference between A and B is chosen by the transformation function, then actually the function chose the whole A. *Maximality* and *weak maximality* were suggested in (Krümpelmann et al. 2011).

Observation 3. *Let B be a belief base, $*_p$ be a multiple base revision operator on B that satisfies the postulates described in Definition 3 and f be a transformation function. Let $\odot$ be the multiple selective base revision operator on B based on $*_p$ and f. Then if f satisfies:*

1. *weak implication, then $\odot$ satisfies very weak inclusion.*
2. *consistency preservation then $\odot$ satisfies weak consistency.*
3. *implication, then $\odot$ satisfies weak inclusion and weak consistency.*
4. *choice, then $\odot$ satisfies inclusion.*
5. *weak maximality, then $\odot$ satisfies very weak inclusion and vacuity 2.*

Proof. 1. We prove by cases: (a) If $B \cup A \vdash \bot$, then $Cn(B \cup A) = K_\bot$ and, therefore, $B \odot A \subseteq Cn(B \cup A)$. (b) If $B \cup A \nvdash \bot$, then $B \odot A = B *_p f(A), B *_p f(A) \subseteq B \cup f(A)$ ($*_p$-inclusion), $B \cup f(A) \subseteq B \cup Cn(A)$ (weak implication). Hence, $B \odot A \subseteq Cn(B \cup A)$.

2. Let $A \nvdash \bot$. Then, by *f-consistency preservation*, $f(A) \nvdash \bot$ and, by *$*_p$-weak consistency*, $B *_p f(A) \nvdash \bot$. Therefore, $B \odot A \nvdash \bot$ and *weak consistency* is satisfied.

3. We have that $B \odot A = B *_p f(A)$; then by *$*_p$-inclusion* $B *_p f(A) \subseteq B \cup f(A)$ and, since f satisfies implication, $B *_p f(A) \subseteq B \cup Cn(A)$. Thus, $B \odot A \subseteq B \cup Cn(A)$ and *weak inclusion* is satisfied. That $\odot$ satisfies weak consistency follows from item 2, since implication implies consistency preservation.

4. We have that $B \odot A = B *_p f(A)$; then by *$*_p$-inclusion* $B *_p f(A) \subseteq B \cup f(A)$ and, since f satisfies choice, $B *_p f(A) \subseteq B \cup A$. Thus, $B \odot A \subseteq B \cup A$ and *inclusion* is satisfied.

5. Weak maximality implies weak implication; then, very weak inclusion follows from item 1. For vacuity, suppose that $B \cup A \nvdash \bot$. Then, by weak maximality, $A = f(A)$ so that $B \odot A = B *_p A$ and, by $*_p$-vacuity, $B \odot A = B *_p A = B \cup A$.

□

The observation below clarifies an important property:

Observation 4. *Let B be a belief base, $*_p$ be a multiple base revision operator on B that satisfies* $*$-inclusion *and f be a transformation function. Let $\odot$ be the multiple selective base revision operator on B based on $*_p$ and f. If $A \subseteq B \odot A$ and f satisfies conditional maximality, then $f(A) = A$.*

Proof. Assume that $A \subseteq B \odot A$. Since $B \odot A = B *_p f(A)$, we have that $A \subseteq B *_p f(A)$ and, by $*$-*inclusion*, $A \subseteq B \cup f(A)$. Then we have three possibilities: (i) $A \subseteq B$, which implies that $A \setminus B = \emptyset \subseteq f(A)$. (ii) $A \subseteq f(A)$, which also implies that $A \setminus B \subseteq f(A)$. (iii) $A \cap B \neq \emptyset$ and $A \setminus B \subseteq f(A)$. By f-*conditional maximality*, in all of the three cases we have that $f(A) = A$. □

The following representation theorems have been obtained for three classes of multiple selective base revision functions.

Theorem 4. *Let B be a belief base in $\mathcal{L}$ and $\odot$ be an operator on B. Then the following conditions are equivalent:*

1. *$\odot$ satisfies conditional inclusion, weak consistency, conditional uniformity, weak proxy success, conditional success, uniform success and weak relevance.*
2. *There exists a multiple base revision $*_p$ for B and a transformation function f that satisfies conditional maximality, consistency preservation, idempotence, choice, uniform identity and such that $B \odot A = B *_p f(A)$, for every A.*

Proof. (1) *implies* (2): we first define f and $*$:

$$f(A) = \begin{cases} A & \text{if } A \subseteq B \odot A; \\ r(A) & \text{otherwise, where } r \text{ is a (well defined) function from } 2^{\mathcal{L}} \text{ to } 2^{\mathcal{L}} \text{ such that } r(A) \subseteq B \odot A \text{ and } B \odot A = B \odot r(A). \end{cases}$$

This definition is possible since $\odot$ satisfies *weak proxy success*.

$$B{*}A = \begin{cases} B \odot A & A \subseteq B \odot A; \\ B *' A & \text{otherwise, where } *' \text{ is any operation that. satisfies the five axioms from Definition 3} \end{cases}$$

We need to show that:

(a) f is a (well-defined) transformation function;

(b) f satisfies the properties;

(c) $*$ is a multiple base revision, according to Definition 3;

(d) $B \odot A = B * f(A)$, for all A.

The proofs are given below:

(a) To prove that f is a (well defined) function we must show that for all $A \subseteq \mathcal{L}$ there exists $A' \subseteq \mathcal{L}$ such that $f(A) = A'$ and that, if $A_1 = A_2$, then $f(A_1) = f(A_2)$.
Let $A \subseteq \mathcal{L}$. If $A \subseteq B \odot A$, then $f(A) = A$. Otherwise, $f(A) = r(A) = A'$, for some A' such that $A' \subseteq B \odot A = B \odot A'$. Such A' exists since $\odot$ satisfies *weak proxy success*. Assume now that $A_1 = A_2$. If $A_1 \subseteq B \odot A_1$, then by $\odot$-*uniform success* it follows that $A_2 \subseteq B \odot A_2$. Thus $f(A_1) = A_1 = A_2 = f(A_2)$. If $A_1 \nsubseteq B \odot A_1$, then $A_2 \nsubseteq B \odot A_2$. Thus $f(A_1) = r(A_1)$ and $f(A_2) = r(A_2)$. r is a (well-defined) function. Thus, from $A_1 = A_2$ it follows that $r(A_1) = r(A_2)$. Therefore, $f(A_1) = f(A_2)$.

(b) To show that f satisfies conditional maximality, assume that $A \setminus B \subseteq f(A)$. From the definition of f we have that $f(A) \subseteq B \odot A$, which implies that $A \setminus B \subseteq B \odot A$. Then, by $\odot$ *conditional success*, $A \subseteq B \odot A$ and, from the definition of f, we have that $f(A) = A$.
We will now show that f satisfies consistency preservation. Assume that $A \nvdash \bot$. If $A \subseteq B \odot A$, then $f(A) = A$, from which it follows that $f(A) \nvdash \bot$. If $A \nsubseteq B \odot A$, then from the definition of f it follows that $f(A) \subseteq B \odot A$. By $\odot$ *weak consistency* it follows that $B \odot A \nvdash \bot$ and, hence, $f(A) \nvdash \bot$.
To show that f satisfies idempotence, we prove by cases:
(1) $A \subseteq B \odot A$. Thus $f(A) = A$ and $f(f(A)) = f(A)$.
(2) $A \nsubseteq B \odot A$. Thus $f(A) = r(A)$ and $r(A) \subseteq B \odot r(A)$. Hence, from the definition of f it follows that $f(r(A)) = r(A)$. From the latter and $f(A) = r(A)$ it follows that $f(f(A)) = f(A)$.
We will now show that f satisfies uniform identity. Assume that it holds for all subsets $B' \subseteq B, B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$. Let $f(A) = A$. From the definition of f it holds that $f(A) \subseteq B \odot A$. Thus $A \subseteq B \odot A$. By $\odot$ *uniform success* it follows that $C \subseteq B \odot C$, from which follows that $f(C) = C$. By symmetry of the case it follows that if $f(C) = C$, then $f(A) = A$. Hence it holds that $f(A) = A$ iff $f(C) = C$.

(c) That $*$ satisfies *success* follows trivially from definition of $*$. That $*$ satisfies *weak consistency* and *inclusion* follows from the fact that both $\odot$ and $*'$ satisfy *weak consistency* and *inclusion*.
Relevance follows from the definition of $*$, $\odot$ *weak relevance* and $*'$ *relevance*.
In order to show that $*$ satisfies uniformity, assume that it holds for all subsets $B' \subseteq B, B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$. By $\odot$ *uniform success* it follows that $A \subseteq B \odot A$ iff $C \subseteq B \odot C$. We prove by cases:
(1) $A \subseteq B \odot A$. Then $C \subseteq B \odot C$ and, hence, $B * A = B \odot A$ and $B * C = B \odot C$, from which it follows by $\odot$ *conditional uniformity* that $B \cap (B * A) = B \cap (B * C)$.
(2) $A \nsubseteq B \odot A$. Then $C \nsubseteq B \odot C$. Thus $B * A = B *' A$ and $B * C = B *' C$. Hence by $*'$ *uniformity* it follows that $B \cap (B * A) = B \cap (B * C)$.

(d) We will now prove that $B \odot A = B * f(A)$.
case 1) $A \subseteq B \odot A$. Hence $f(A) = A$ and $B * A = B \odot A$. Thus $B * f(A) = B \odot A$.
case 2) $A \nsubseteq B \odot A$. From the definition of f it holds that $B \odot A = B \odot f(A)$ and $f(A) \subseteq B \odot A$. Hence $f(A) \subseteq B \odot f(A)$. Thus, from the definition of * it follows that $B * f(A) = B \odot f(A)$, from which it follows that $B \odot A = B * f(A)$.

(2) *implies* (1): For conditional success, let $A \setminus B \subseteq B \odot A$. Since $B \odot A = B *_p f(A)$, by $*_p$-*inclusion* we have that $A \setminus B \subseteq B \cup f(A)$, which implies that $A \setminus B \subseteq f(A)$. From

f-*conditional maximality* we have that $f(A) = A$ and, by $*_p$-*success*, it follows that $A \subseteq B * A$. Thus $A \subseteq B \odot A$.

In order to prove conditional inclusion, let $A \subseteq B \odot A$. Since f satisfies *conditional maximality*, from Observation 4 we have that $f(A) = A$ and, since $B \odot A = B *_p f(A)$, by $*$-*inclusion* we have that $B \odot A \subseteq B \cup A$.

For *weak relevance*, let $A \subseteq B \odot A$ and suppose that there is some $\beta \in B$ such that $\beta \notin B \odot A$. Since $B \odot A = B *_p f(A)$, we have that $\beta \in B \setminus (B *_p f(A))$ and, by $*_p$-*relevance*, there is a set B' such that $B *_p f(A) \subseteq B' \subseteq B \cup f(A)$, $B' \nvdash \bot$ but $B' \cup \{\beta\} \vdash \bot$. So it remains to prove that $B' \subseteq B \cup A$. Since from $A \subseteq B \odot A$ and f-*conditional maximality* we have from Observation 3 that $f(A) = A$, it follows directly.

Weak consistency follows from item 2 of Observation 3.

For uniform success, consider that it holds for all subsets B' of B that $B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$. Hence, since f satisfies uniform identity, $f(A) = A$ holds iff $f(C) = C$ holds. Assume that $A \subseteq B \odot A$. From f-*conditional maximality* and Observation 4 we have that $f(A) = A$. On the other hand, by f-*uniform identity* it follows that $f(C) = C$. From the definition of $\odot$ it holds that $B \odot C = B *_p f(C)$. By $*_p$-*success* it follows that $C = f(C) \subseteq B \odot C$. Thus $C \subseteq B \odot C$. By symmetry of the case it holds that if $C \subseteq B \odot C$, then $A \subseteq B \odot A$. Hence it holds that $A \subseteq B \odot A$ iff $C \subseteq B \odot C$.

In order to show that $\odot$ satisfies conditional uniformity, consider that it holds for all subsets B' of B that $B' \cup A \vdash \bot$ iff $B' \cup C \vdash \bot$. Let $A \subseteq B \odot A$. From f-*conditional maximality* and Observation 4 we have that $f(A) = A$ and from f-*uniform identity* it follows that $f(C) = C$. By $*_p$-*uniformity* it follows that $B \cap (B *_p A) = B \cap (B *_p C)$. Thus $B \cap (B \odot A) = B \cap (B *_p f(A)) = B \cap (B *_p A) = B \cap (B *_p C) = B \cap (B *_p f(C)) = B \cap (B \odot C)$. □

To conclude, the *proxy success* and *choice success* versions:

Theorem 5. *Let B be a belief base in $\mathcal{L}$ and $\odot$ be an operator on B. Then the following conditions are equivalent:*

1. *$\odot$ satisfies conditional inclusion, weak consistency, conditional uniformity, proxy success, conditional success, uniform success and weak relevance.*
2. *There exists a multiple base revision $*_p$ for B and a transformation function f that satisfies conditional maximality, consistency preservation, idempotence, implication, uniform identity and such that $B \odot A = B *_p f(A)$, for every A.*

Proof. This proof is quite similar to that of Theorem 6. To show that (1) implies (2), we define f to be a function like that of the previous proof but with an additional restriction for r: $r(A) \subseteq Cn(A)$. The existence of such a function follows from proxy success. The proofs for f are essentially the same, and the implication property follows trivially. To show that (2) implies (1) we only have to add a proof of proxy success, which we obtain from Observation 3. □

Theorem 6. *Let B be a belief base in $\mathcal{L}$ and $\odot$ be an operator on B. Then the following conditions are equivalent:*

1. *$\odot$ satisfies inclusion, weak consistency, conditional uniformity, choice success, conditional success, uniform success and weak relevance.*
2. *There exists a multiple base revision $*_p$ for B and a transformation function f that satisfies conditional maximality, consistency preservation, idempotence, choice, uniform identity and such that $B \odot A = B *_p f(A)$, for every A.*

Proof. This proof is quite similar to that of Theorem 4. To show that (1) implies (2), we define f to be a function like that of the previous proof but with an additional restriction for r: $r(A) \subseteq A$. The existence of such a function follows from choice success. The proofs for f are essentially the same, and the choice property follows trivially. To show that (2) implies (1) we only have to add proofs for inclusion and choice success, which we obtain from item 4 of Observation 3 and from Theorem 4 and f-*choice*. □

The intuition behind these theorems is pretty much the same for theories. Theorem 4 allows very general operations which do not demand $f(A)$ to be derived from A. Theorem 5 refers to the operations in which $f(A)$ is limited to A and its logical consequences. Finally, Theorem 6 represents the most restrictive ones since $f(A)$ is limited to the subsets of A.

Example 2. *(Example 1 revisited) Consider a representation in propositional logic for C's beliefs: the bedroom is organized (p) but full of dust (q), the bathroom is flooded (r), the kitchen is full of food scraps (s), and the living room is neat (t). In addition, a clean kitchen (z) is not consistent with food scraps ($s \to \neg z$), and a broken object in the living room (x) makes it not neat ($x \to \neg t$). The other robots tell that the beds need to be made ($\neg p$), there is a silver tap in the bathroom (v) that is open (u, thus $u \wedge v$), there is a broken vase in the living room (x) because of a dinosaur that has entered there (d, thus $d \wedge x$) and the kitchen is clean (z).*

Here, we are going to adapt the filter f in order to return only subsets of input. Then a possible filter is $f(\{\neg p, u \wedge v, d \wedge x, z\}) = \{\neg p, u \wedge v\}$. After that, a prioritized revision is applied: $\{p, q, r, s, t, s \to \neg z, x \to \neg t\} *_p \{\neg p, u, x\}$. A possible final result could be $\{\neg p, q, r, u \wedge v, s, s \to \neg z, x \to \neg t, \}$

5 Related Work

There is broad literature about non-prioritized revision operations for singleton inputs. *Screened Revision* (Makinson 1997) explores the context in which an agent, in addition to its set of beliefs K, makes use of a set of core beliefs A that cannot be retracted. Then, an input sentence is accepted for revision only if it is consistent with $K \cap A$. In a slightly different approach, *Credibility-limited Revision* (Hansson et al. 2001) considers that there is a set $\mathcal{C}$ of credible sentences and an input sentence α is accepted for revision only if $\alpha \in \mathcal{C}$. Still, on the use of core beliefs but in a multiple context, *Evaluative Multiple Revision* (Yuan, Ju, and Wen

2014) works with belief states formed by a belief base B and a subset A of it that is immune to revision. An input set is submitted to a pre-processing that classifies its sentences in two disjoint sets of plausible or implausible information (consistent or not with A), and both sets are considered in the revision process since B can be initially inconsistent. All implausible information has to be given up. Besides working with multiple revision, our approach does not assume that there is a core or a set of credible sentences and permits partial acceptance (including weakening), which makes the transformation function more general.

In (Krümpelmann et al. 2011) the authors proposed a concrete implementation of a transformation function using Deductive Argumentation (Besnard and Hunter 2001) as the tool to evaluate the desirability of new information for a belief base. To allow a new belief to contain arguments, they worked with a multiple version of Selective Revision for bases. However, they explored a smaller set of properties, not working with different success/inclusion cases or relevance, for example. Also, the input sets considered in this paper do not originate from argumentation, and we also brought a generalization for theories.

6 Conclusion and Future Work

In this paper, we presented a thorough study of Multiple Selective Revision - revision operators that may reject part of the input set. Based on existing definitions and characterizations of both Multiple and Selective Revision, we studied the generalization of partial acceptance via Selective Revision to make it possible to deal with sets of sentences as input (instead of a single one).

Multiple Selective Revision can be constructed by means of applying a transformation function to the input and then performing a multiple revision by the transformed input.

We have provided lists of plausible postulates for the operations and also for the transformation functions, and relations between them were observed. Constructions were defined, and representation theorems showed the connection between the postulates and the constructions. The generalization was proposed for both beliefs sets and belief bases.

Future work includes exploring more postulates (such as the supplementary AGM ones), more constructions for multiple selective revision and the analysis of other possible scenarios (for example, when the input is inconsistent but has a consistent subset).

Acknowledgments

The first author is supported by the Brazilian funding agency CAPES.

References

Alchourrón, C.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change. *Journal of Symbolic Logic* 50(02):510–530.

Barber, K. S., and Kim, J. 2001. Belief revision process based on trust: Agents evaluating reputation of information sources. In *Trust in Cyber-societies*. Springer. 73–82.

Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artificial Intelligence* 128(1-2):203–235.

Darwiche, A., and Pearl, J. 1997. On the logic of iterated belief revision. *Artificial Intelligence* 89(1):1 – 29.

Falappa, M.; Kern-Isberner, G.; Reis, M.; and Simari, G. 2012. Prioritized and non-prioritized multiple change on belief bases. *Journal of Philosophical Logic* 41(1):77–113.

Fermé, E., and Hansson, S. O. 1999. Selective revision. *Studia Logica* 63(3):331–342.

Fermé, E., and Hansson, S. O. 2018. *Belief Change: Introduction and Overview*. Springer Briefs in Computer Science Series. Springer.

Fuhrmann, A., and Hansson, S. O. 1994. A survey of multiple contractions. *Journal of Logic, Language and Information* 3(1):39–75.

Fuhrmann, A. 1988. *Relevant Logics, Modal Logics and Theory Change*. Ph.D. Dissertation, Australian National University, Camberra.

Fuhrmann, A. 1997. *An Essay on Contraction*. FOLLI.

Hansson, S. O.; Fermé, E. L.; Cantwell, J.; and Falappa, M. A. 2001. Credibility limited revision. *Journal of Symbolic Logic* 1581–1596.

Hansson, S. O. 1991. *Belief Base Dynamics*. Ph.D. Dissertation, Uppsala University, Uppsala, Suécia.

Hansson, S. O. 1993. Reversing the levi identity. *Journal of Philosophical Logic* 22(6):637–669.

Hansson, S. O. 1999a. A survey of non-prioritized belief revision. *Erkenntnis* 50(2-3):413–427.

Hansson, S. O. 1999b. *A Textbook of Belief Dynamics*. Norwell, MA, USA: Kluwer Academic Publishers.

Krümpelmann, P.; Thimm, M.; Falappa, M. A.; García, A. J.; Kern-Isberner, G.; and Simari, G. R. 2011. Selective revision by deductive argumentation. In *International Workshop on Theorie and Applications of Formal Argumentation*, 147–162. Springer.

Makinson, D. 1997. Screened revision. *Theoria* 63(1-2):14–23.

Pantoja, C. E.; Stabile, M. F.; Lazarin, N. M.; and Sichman, J. S. 2016. Argo: An extended jason architecture that facilitates embedded robotic agents programming. In *International Workshop on Engineering Multi-Agent Systems*, 136–155. Springer.

Resina, F.; Garapa, M.; Wassermann, R.; Fermé, E.; and Reis, M. 2020. Choosing what to believe-new results in selective revision. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, 687–691.

Stabile, M. F.; Pantoja, C. E.; and Sichman, J. S. 2018. Experimental analysis of the effect of filtering perceptions in bdi agents. *International Journal of Agent-Oriented Software Engineering* 6(3-4):329–368.

Yuan, Y.; Ju, S.; and Wen, X. 2014. Evaluative multiple revision based on core beliefs. *Journal of Logic and Computation* 25(3):781–804.

Orders and Belief Revision

María Victoria León[1] , **Ramon Pino Pérez**[2]
[1] Escuela Superior Politécnica de Chimborazo - Ecuador
[2] CRIL - CNRS - Université d'Artois - France
maria.leons@espoch.edu.ec, pinoperez@cril.fr

Abstract

We present a brief survey of representation theorems in belief revision which capture different notions of ordered structures; in particular total preorders, partial orders, and semiorders. Although most of the results are known (Katsuno and Mendelzon 1991, Benferhat et al 2005, Peppas and Williams 2014), we give a new and compact presentation, their proofs and a synthetic view of the postulates' landscape which allows defining belief change operators in correspondence with every kind of structure mentioned previously.

1 Introduction

Belief revision aims at understanding how to integrate a new piece of information in a corpus of beliefs obeying certain principles. The main problem comes from the fact that the new piece of information can be in conflict with the current beliefs. The logical model of belief revision(Alchourrón, Gärdenfors, and Makinson 1985; Gärdenfors 1988) has given a satisfactory answer to this problem. It has been deeply studied in the last forty years.

Representation theorems have been an interesting and important tool in this study because they give a constructive and practical view of revision operators. They give a clear materialization of the idea that beliefs are organized in concentric spheres: as the beliefs are more entrenched, the concentric sphere which contains them is smaller. This materialization becomes clear in terms of ordered structures over the interpretations.

In this work, we present a survey of some representation theorems. One interesting feature of this survey is to have a more global vision of them. In particular, they give information about the behavior of operators via the links between certain sets of postulates and their relationships with the semantics structures representing them.

We use here the Katsuno-Mendelzon approach (Katsuno and Mendelzon 1991) in which all data is of the same type: logical formulas. In particular, the epistemic state of an agent is represented by a formula, the new piece of information and the resulting epistemic state after incorporating the new piece of information is also represented by a formula.

Let us recall the general form of a representation theorem:

> An operator satisfies a *special* set of postulates of rationality iff there is a *special* function mapping epistemic states φ in a ordered structure $\prec_\varphi$ over interpretations such that the models of the resulting state of revising φ by α are the models of the new information α which are minimal elements (the most preferred) in the ordered structure $\prec_\varphi$ associated to the old epistemic state.

Here the words *ordered structure* refer to binary relations which can have some properties of "orderings" in a very general way and not necessarily in the strict notion of order in mathematics.

Our main contribution consits in giving an complete account of representation theorems for the most natural ordered structures: total preorders, semiorders, partial orders and some special types of partial orders and semiorders.

We organize this work in four sections following this brief introduction. Section 2 is devoted to the logical notions and the ordered structures. Section 3 is devoted to postulates and the assignments. In Section 4 we state the representation theorems. Finally, in the Appendix we give the proofs. We consider this Appendix containing all the proofs as a contribution of the work because they show in extenso the techniques allowing to make the links between syntactical properties and structural properties.

2 Preliminaries

2.1 Some logical notation

We denote by $\mathcal{L}$ the set of formulas of a propositional language built over a finite set of propositional variables $\mathcal{P}$ plus the constants $\top$ and $\bot$with the usual meaning of true and false. The elements of $\mathcal{L}$ are denoted by lower case Greek letters α, β, γ, φ . . . (possibly with subscripts). The set of valuation functions (interpretations) from the set of propositional variables into the boolean set $\{0, 1\}$ (false, true) is denoted Ω. As usual, we write $\omega \models \alpha$ when a valuation $\omega \in \Omega$ satisfies a formula α, *i.e.* when ω is a model of α. The set of models of a formula α is denoted by $[\![\alpha]\!]$. If M is a set of models we denote by α_M a formula such that $[\![\alpha_M]\!] = M$. When the size of M is small we often omit the

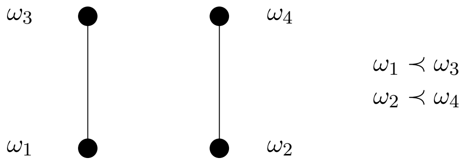


Figure 1: An example of partial order

braces, by writing, e.g., $\alpha_{\omega\omega'}$ instead of $\alpha_{\{\omega,\omega'\}}$. The set of consistent formulas will be denoted $\mathcal{L}^*$.

2.2 Structures

In this subsection we present the different types of ordered structures that will be used in this work. Actually, our ordered structures are binary relations over a finite set. We suppose that the finite set is Ω and the relations will be denoted by the symbol $\prec$ with subscript when necessary. The choice of this notation is due to the fact that our relations will be asymmetric, in particular irreflexive. We call this kind of relations *strict* ordered structures.

A binary relation $\prec$ over Ω is irreflexive when $\omega \not\prec \omega$ for every $\omega \in \Omega$. The relation $\prec$ is transitive when for every triple $\omega_1, \omega_2, \omega_3 \in \Omega$, if $\omega_1 \prec \omega_2$ and $\omega_2 \prec \omega_3$ then $\omega_1 \prec \omega_3$. The relation $\prec$ is asymmetric when for every couple $\omega_1, \omega_2 \in \Omega$, if $\omega_1 \prec \omega_2$ then $\omega_2 \not\prec \omega_1$. When $M \subseteq \Omega$, the minimal elements of M with respect to the relation $\prec$, denoted by $min(M, \prec)$ is the set defined by

$$min(M, \prec) = \{\omega \in M : \nexists \omega' \in M, \omega' \prec \omega\}$$

We denote $min(\Omega, \prec)$ by $min(\prec)$.

The most general class of structures we consider is the class of partial orders:

Definition 1. *Let $\prec$ be a binary relation over Ω. The relation $\prec$ is a* partial order *if it is irreflexive and transitive.*

It is easy to see that a partial order is asymmetric and acyclic (there are no cycles). Moreover, the set $min(\prec)$ is always nonempty.

Definition 2. *Let $\prec$ be a partial order over Ω. The relation $\prec$ is a* min-partial order *if for every $\omega \in min(\prec)$ and every $\omega' \notin min(\prec)$ we have $\omega \prec \omega'$.*

Figures 1 and 2 illustrate a partial order and a min-partial order respectively. Note that the partial order of Figure 1 is not a min-partial order.

Let $\prec$ be a partial order over Ω. We define the indifference relation $\sim$ over Ω associated to $\prec$ by putting $\omega \sim \omega'$ iff $\omega \not\prec \omega'$ and $\omega' \not\prec \omega$.

Another interesting subclass of partial orders is that of the ranking orders defined as follows:

Definition 3. *Let $\prec$ be a partial order over Ω. The relation $\prec$ is a* ranking order *if for every $\omega_1, \omega_2, \omega_3 \in \Omega$ if $\omega_1 \sim \omega_2$ and $\omega_1 \prec \omega_3$ then $\omega_2 \prec \omega_3$.*

It is well known that if $\prec$ is a ranking order then the relation $\preceq$ defined by $\omega \preceq \omega'$ iff $\omega \prec \omega'$ or $\omega \sim \omega'$ is a *total preorder*, that is, a transitive relation which is total, *i.e.*, all its elements are comparable under $\preceq$; in particular the reflexivity is satisfied for total preorders. Moreover, it is also well known that $\prec$ is a ranking order iff there exists a ranking function $r : \Omega \longrightarrow \mathbb{R}$ such that

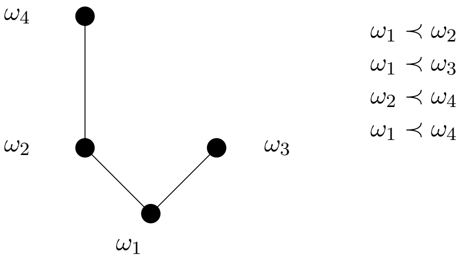

ω_4
$\omega_1 \prec \omega_2$
$\omega_1 \prec \omega_3$
$\omega_2 \prec \omega_4$
$\omega_1 \prec \omega_4$
ω_2
ω_3
ω_1

Figure 2: An example of a min-partial order

$$\omega \prec \omega' \quad \Leftrightarrow \quad r(\omega) < r(\omega')$$

Because of this property, we will call these relations ranking orders.

We are going to consider also a class of partial orders which can be defined via ranking functions in a special way as we will see later.

Definition 4. *Let $\prec$ be a binary relation over Ω. The relation $\prec$ is a* semiorder *if for every $\omega_1, \omega_2, \omega_3, \omega_4 \in \Omega$*

(SO1) $\omega_1 \not\prec \omega_1$

(SO2) *If $\omega_1 \prec \omega_2 \prec \omega_3$ and $\omega' \in Val$, then $\omega_1 \prec \omega'$ or $\omega' \prec \omega_3$*

(SO3) *If $\omega_1 \prec \omega_2$ and $\omega_3 \prec \omega_4$, then $\omega_1 \prec \omega_4$ or $\omega_3 \prec \omega_2$*

Indeed, it is not hard to see that a semiorder is a partial order, *i.e.*, the transitivity is satisfied. Moreover, it is well known (Pirlot and Vincke 1997) that the relation $\prec$ is a semiorder iff there exist a ranking function $r : \Omega \longrightarrow \mathbb{R}$ and a real $q > 0$ such that

$$\omega \prec \omega' \quad \Leftrightarrow \quad (r(\omega') - r(\omega)) > q$$

It is interesting to note that every ranking order is also a semiorder. The converse, of course, is not true. Moreover, the indifference relation $\sim$ associated to a semiorder is not, in general, transitive.

Finally, we consider the following subclass of semiorders:

Definition 5. *Let $\prec$ be a semiorder over Ω. The relation $\prec$ is a* min-semiorder *if for every $\omega \in min(\prec)$ and every $\omega' \notin min(\prec)$ we have $\omega \prec \omega'$.*

Figures 3 and 4 illustrate a semiorder and a min-semiorder respectively. Note that the semiorder in Figure 3 is not a min-semiorder. Figure 5 summarizes the inclusion relations between the ordered classes introduced so far.

3 Postulates and assignments

All operators $\circ$ considered in this work are functions of the following type

$$\circ : \mathcal{L}^* \times \mathcal{L} \longrightarrow \mathcal{L}$$

As usual $\circ(\varphi, \alpha)$ is denoted by $\varphi \circ \alpha$.

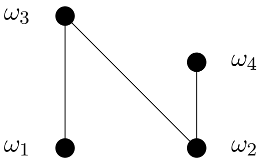


Figure 3: Example of semiorder

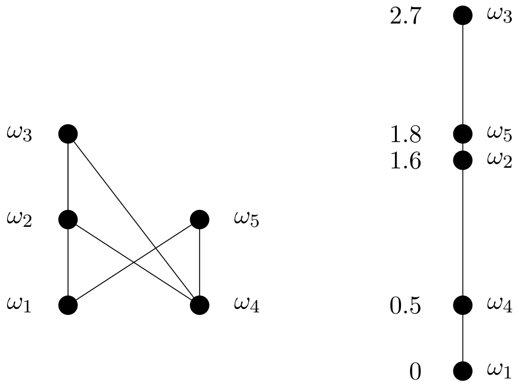


Figure 4: A min-semiorder with its ranking function and $q = 1$

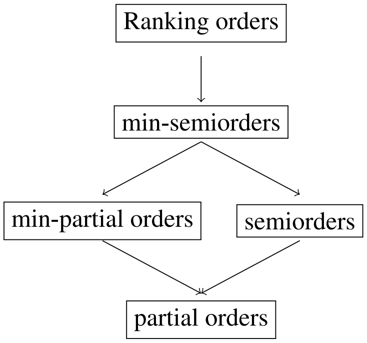


Figure 5: The landscape of structured orders

3.1 Postulates and classes of operators

(R1) $\varphi \circ \alpha \vdash \alpha$

(R2) If $\varphi \wedge \alpha \nvdash \bot$ then $\varphi \circ \alpha \equiv \varphi \wedge \alpha$

(R2') $\varphi \circ \top \equiv \varphi$

(R3) If $\alpha \nvdash \bot$ then $\varphi \circ \alpha \nvdash \bot$

(R4) If $\varphi_1 \equiv \varphi_2$ and $\alpha_1 \equiv \alpha_2$ then $\varphi_1 \circ \alpha_1 \equiv \varphi_2 \circ \alpha_2$

(R5) $(\varphi \circ \alpha) \wedge \beta \vdash \varphi \circ (\alpha \wedge \beta)$

(R6) If $(\varphi \circ \alpha) \wedge \beta \nvdash \bot$ then $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$

(R7) If $(\varphi \circ \alpha) \vdash \beta$ and $\varphi \circ \beta \vdash \alpha$ then $\varphi \circ \alpha \equiv \varphi \circ \beta$

(R8) $(\varphi \circ \alpha) \wedge (\varphi \circ \beta) \vdash \varphi \circ (\alpha \vee \beta)$

(R9) If $(\varphi \circ \alpha) \wedge \beta \nvdash \varphi \circ \beta$ then $(\varphi \circ \beta) \wedge \alpha \vdash \varphi \circ \alpha$

(R10) If $(\varphi \circ \alpha) \wedge \beta \vdash \bot$ and $(\varphi \circ \alpha) \wedge \gamma \nvdash \bot$ then $(\varphi \circ \gamma) \wedge (\alpha \wedge \beta) \vdash \varphi \circ (\alpha \wedge \beta)$

Postulates R1-R6 encode in the finite case the postulates AGM of belief revision (Alchourrón, Gärdenfors, and Makinson 1985). They were introduced in (Katsuno and Mendelzon 1991). Postulates R7 and R8 were also introduced in (Katsuno and Mendelzon 1991). Postulates R9 and R10 were introduced by (Peppas and Williams 2014). Postulate R2' was introduced by (Benferhat, Lagrue, and Papini 2005). It is a weakening of postulate R2.

Definition 6. *An operator $\circ$ is called a revision operator iff it satisfies the postulates R1-R6.*

Postulates R1-R5 plus R7 and R8 together give a special kind of operators introduced by (Katsuno and Mendelzon 1991).

Definition 7. *An operator $\circ$ is called a partial KM (p-KM for short) revision operator iff it satisfies the postulates R1-R5 plus R7 and R8.*

As we will see this class of operators is more general than the class of revision operators. Even a more general class of operators, introduced by (Benferhat, Lagrue, and Papini 2005), is obtained when we replace in the previous set of postulates, the postulate R2 by R2'.

Definition 8. *An operator $\circ$ is called a partial (p for short) revision operator iff it satisfies the postulates R1, R2', R3-R5 plus R7 and R8.*

Another combination of postulates give us an interesting class of operators introduced by (Peppas and Williams 2014).

Definition 9. *An operator $\circ$ is called a semiorder PW (so-PW for short) revision operator iff it satisfies the postulates R1-R5 plus R8-R10.*

A more general class is obtained by replacing R2 by R2' in the previous set of postulates.

Definition 10. *An operator $\circ$ is called a semiorder (so for short) revision operator iff it satisfies the postulates R1, R2', R3-R5 plus R8-R10.*

3.2 Assignments

The assignments we are considering are functions from $\mathcal{L}^*$ into a set of structured relations having always a set of minimal elements. They map a consistent formula φ into an ordered relation $\prec_\varphi$ with the following two properties:

1. If $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec_{\varphi'}$.
2. $min(\prec_\varphi) = [\![\varphi]\!]$.

Now we proceed to do a classification of different classes of assignments following the type of the ordered relation $\prec_\varphi$ associated to φ.

Definition 11. *Let $\varphi \mapsto \prec_\varphi$ be an assignment. This assignment is said to be*

1. *a* faithful assignment *iff for every φ, the relation $\prec_\varphi$ is a ranking order;*
2. *a* p-KM-faithful assignment *iff for every φ, the relation $\prec_\varphi$ is a min-partial order;*
3. *a* p-faithful assignment *iff for every φ, the relation $\prec_\varphi$ is a partial order;*
4. *a* so-PW-faithful assignment *iff for every φ, the relation $\prec_\varphi$ is a min-semiorder;*
5. *a* so-faithful assignment *iff for every φ, the relation $\prec_\varphi$ is a semiorder.*

4 Representation

In this section we give the main representation theorem. The proofs are in Appendix A.

4.1 The classical representation theorem

Theorem 1. *The operator $\circ$ is a revision operator if and only if there exists a unique faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

4.2 Representation for a kind of partial structures

Theorem 2 (Katsuno and Mendelzon, 91)**.** *The operator $\circ$ is a p-KM-revision operator if and only if there exists a unique p-KM-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

4.3 Representation for general partial structures

The following theorem is implicit in (Benferhat, Lagrue, and Papini 2005). However, in that work, the proof is not given.

Theorem 3. *The operator $\circ$ is a p-revision operator if and only if there exists a unique p-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

4.4 Representation for a kind of semiorders

The following theorem is essentially due to (Peppas and Williams 2014). However, in that work, the formulation is different.

Theorem 4. *The operator $\circ$ is a so-PW-revision operator if and only if there exists a unique so-PW-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

Operators	**Postulates**	**Structure of the assignment**
Revision	R1-R6	Ranking orders
p-KM-revision	R1-R5 R7-R8	min-partial orders
p-revision	R1,R2' R3-R5 R7-R8	partial orders
so-PW-revision	R1-R5 R8-R10	min-semiorders
so-revision	R1,R2' R3-R5 R8-R10	semiorders

Table 1: Summary of representation theorems

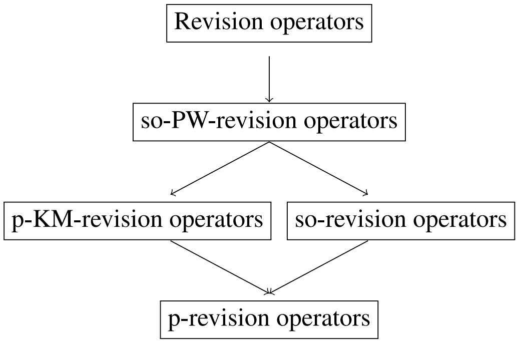


Figure 6: The hierarchy of operators

4.5 Representation for general semiorders

Finally, we obtain a slightly more general representation theorem.

Theorem 5. *The operator $\circ$ is a so-revision operator if and only if there exists a unique so-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

Table 1 summarizes the results of this section.

5 Conclusion

We conclude this work with a corollary of the representation theorems presented here. It can be summarized in the graph representing the inclusion given in Figure 6.

An important consequence of the graph in Figure 6 is that the set of postulates characterizing a vertex v in the graph entails every postulate of a class accessible from this vertex v.

Some other remarks to conclude: first, the statement of Theorem 3 is an adaptation to propositional framework of a similar Theorem in (Benferhat, Lagrue, and Papini 2005) in the framework of complex epistemic states. To our knowledge, the proof given in this work of this result is the first in the literature. Second, Theorem 4 is an adaptation to propositional framework of a similar Theorem by (Peppas and Williams 2014) in the framework of theories as epistemic states. Third, Theorem 5 is new.

Theorems 1 and 5 show different ways of organizing the information when we have a ranking function.

Acknowledgements

The second author has benefited from the support of the AI Chair BE4musIA of the French National Research Agency (ANR) and partially funded by the program PAUSE of Collège de France.

A Proofs

We state two new postulates introduced by (Katsuno and Mendelzon 1991) and a lemma which is a key tool in the proof of Theorem 2.

(R6w) If $(\varphi \circ \alpha) \vdash \beta$ then $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$

(Rt) If $\varphi\circ(\alpha\vee\beta)\equiv\alpha$ and $\varphi\circ(\beta\vee\gamma)\equiv\beta$ then $\varphi\circ(\alpha\vee\gamma)\equiv\alpha$

Lemma 1 (Katsuno and Mendelzon, 91)**.** *If $\circ$ satisfies R1, R4 and R5, then R7, R6w and Rt are equivalent.*

Proof.
R7 $\Rightarrow$ R6w. Assume $\varphi \circ \alpha \vdash \beta$. By R1 $\varphi \circ \alpha \vdash \alpha$, thus $\varphi \circ \alpha \vdash \alpha \wedge \beta$. Again, by R1, $\varphi \circ (\alpha \wedge \beta) \vdash \alpha \wedge \beta$, then $\varphi \circ (\alpha \wedge \beta) \vdash \alpha$. From $\varphi \circ \alpha \vdash \alpha \wedge \beta$, $\varphi \circ (\alpha \wedge \beta) \vdash \alpha$ and the assumption we have $\varphi \circ \alpha \equiv \varphi \circ (\alpha \wedge \beta)$, thus $\varphi \circ (\alpha \wedge \beta) \vdash \varphi \circ \alpha$ (*). Note that, $\varphi \circ (\alpha \wedge \beta) \vdash \alpha \wedge \beta$ and $\alpha \wedge \beta \vdash \beta$, therefore $\varphi \circ (\alpha \wedge \beta) \vdash \beta$ (**). Then, from (*) and (**) we have $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$.
R6w $\Rightarrow$ Rt. Assume that $\varphi\circ(\alpha\vee\beta) \equiv \alpha$ and $\varphi\circ(\beta\vee\gamma) \equiv \beta$. We want to show that $\varphi \circ (\alpha \vee \gamma) \equiv \alpha$.
Claim 1: $\varphi \circ (\alpha \vee \beta \vee \gamma) \vdash \alpha$.
By R5 $[\varphi \circ (\alpha \vee \beta \vee \gamma)] \wedge [\alpha \vee \beta] \vdash \varphi \circ (\alpha \vee \beta)$. Since, by assumption, $\varphi \circ (\alpha \vee \beta) \vdash \alpha$, we have $[\varphi \circ (\alpha \vee \beta \vee \gamma)] \wedge [\alpha \vee \beta] \vdash \alpha$. Then, by deduction rule, $\varphi \circ (\alpha \vee \beta \vee \gamma) \vdash (\alpha \vee \beta) \rightarrow \alpha$. But $(\alpha \vee \beta) \rightarrow \alpha \equiv \neg\beta \vee \alpha$, therefore $\varphi \circ (\alpha \vee \beta \vee \gamma) \vdash \neg\beta \vee \alpha$ (*). In an analogous way, we get $\varphi \circ (\alpha \vee \beta \vee \gamma) \vdash \neg\gamma \vee \beta$ (**). From R1 we have $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash \alpha\vee\beta\vee\gamma$ (***). Then, by (*), (**) and (***), we obtain $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash (\neg\beta\vee\alpha)\wedge(\neg\gamma\vee\beta)\wedge(\alpha\vee\beta\vee\gamma)$ but $(\neg\beta \vee \alpha) \wedge (\neg\gamma \vee \beta) \wedge (\alpha \vee \beta \vee \gamma) \vdash \alpha$, therefore $\varphi \circ (\alpha \vee \beta \vee \gamma) \vdash \alpha$.
Claim 2: $\varphi \circ (\alpha \vee \beta \vee \gamma) \equiv \varphi \circ (\alpha \vee \gamma)$.
By Claim 1, $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash \alpha$, then $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash \alpha\vee\gamma$ (*). From R6w and (*) we have $\varphi\circ((\alpha\vee\beta\vee\gamma)\wedge(\alpha\vee\gamma)) \vdash (\varphi\circ(\alpha\vee\beta\vee\gamma))\wedge(\alpha\vee\gamma)$ and from R4 $\varphi\circ(\alpha\vee\gamma) \vdash \varphi\circ(\alpha\vee\beta\vee\gamma)$. Note that, by R5 $(\varphi\circ(\alpha\vee\beta\vee\gamma))\wedge(\alpha\vee\gamma)) \vdash \varphi\circ(\alpha\vee\gamma)$ and, since $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash \alpha\vee\gamma$, we have $\varphi\circ(\alpha\vee\beta\vee\gamma) \vdash \varphi \circ (\alpha \vee \gamma)$. Therefore $\varphi \circ (\alpha \vee \beta \vee \gamma) \equiv \varphi \circ (\alpha \vee \gamma)$.
Claim 3: $\varphi \circ (\alpha \vee \beta \vee \gamma) \equiv \varphi \circ (\alpha \vee \beta)$.
The proof of this Claim is completely analogous to the proof of Claim 2.
Finally, by the assumption and Claims 2 and 3, we obtain $\varphi \circ (\alpha \vee \gamma) \equiv \alpha$.
Rt $\Rightarrow$ R7. Assume $\varphi \circ \alpha \vdash \beta$ and $\varphi \circ \beta \vdash \alpha$. We want to show $\varphi \circ \alpha \equiv \varphi \circ \beta$.
Claim 4: $\varphi \circ [(\varphi \circ \alpha) \vee (\varphi \circ \beta)] \equiv (\varphi \circ \alpha) \vee (\varphi \circ \beta)$
By R1 $\varphi \circ \alpha \vdash \alpha$. By assumption we have $\varphi \circ \beta \vdash \alpha$, therefore $(\varphi \circ \alpha \vee \varphi \circ \beta) \vdash \alpha$. By R5, $(\varphi \circ \alpha) \wedge [(\varphi \circ \alpha) \vee (\varphi \circ \beta)] \vdash \varphi \circ (\alpha \wedge [(\varphi \circ \alpha) \vee (\varphi \circ \beta)])$. Thus, using R4, $\varphi \circ \alpha \vdash \varphi \circ [(\varphi \circ \alpha) \vee (\varphi \circ \beta)]$ (*). In a similar way, we get $\varphi \circ \beta \vdash \varphi \circ [(\varphi \circ \alpha) \vee (\varphi \circ \beta)]$ (**). Then, from (*) and (**) we get $(\varphi \circ \alpha) \vee (\varphi \circ \beta) \vdash \varphi \circ [(\varphi \circ \alpha) \vee (\varphi \circ \beta)]$ By R1 $\varphi \circ ((\varphi \circ \alpha) \vee (\varphi \circ \beta)) \vdash (\varphi \circ \alpha) \vee (\varphi \circ \beta)$. Therefore $\varphi \circ (\varphi \circ \alpha \vee \varphi \circ \beta) \equiv (\varphi \circ \alpha) \vee (\varphi \circ \beta)$. Thus, the proof of Claim 4 is complete.

Now we put $A = (\varphi \circ \alpha) \vee (\varphi \circ \beta)$, $B = \varphi \circ \alpha$ and $C = \alpha \wedge \neg(\varphi \circ \alpha)$. Note that from R4 and Claim 4, we get $\varphi \circ (A \vee B) \equiv \varphi \circ A$ and $\varphi \circ A \equiv A$; from R4 we get $\varphi \circ (B \vee C) \equiv B$. Then, by R4 and Rt, $\varphi \circ (A \vee C) \equiv A$.

Since $\varphi\circ\beta \vdash \alpha$ and $\alpha\wedge\neg(\varphi\circ\alpha) \vdash \alpha$, we have $(\alpha\wedge\neg(\varphi\circ\alpha))\vee(\varphi\circ\beta) \vdash \alpha$. Therefore, $A \vee C \equiv \alpha$ and, by R4 and the fact that $\varphi \circ (A \vee C) \equiv A$, we get $\varphi \circ \alpha \equiv \varphi \circ \alpha \vee \varphi \circ \beta$ (•). In a similar way, putting $A = (\varphi \circ \alpha) \vee (\varphi \circ \beta)$, $B = \varphi \circ \beta$ and $C = \beta \wedge \neg(\varphi \circ \beta)$, we obtain $\varphi \circ \beta \equiv \varphi \circ \alpha \vee \varphi \circ \beta$ (••). Then, by (•) and (••), we have $\varphi \circ \alpha \equiv \varphi \circ \beta$. □

We continue by stating a proposition which summarizes the properties of operators which are representable by assignments.

Proposition 1. *Assume that $\circ$ is an operator and $\varphi \mapsto \prec_\varphi$ is an assignment such that the following representation equation holds*

$$[\![\varphi \circ \alpha]\!] = min([\![\alpha]\!], \prec_\varphi)$$

Then

1. *If the assignment is p-faithful (*i.e. *$\prec_\varphi$ is a partial order) then $\circ$ satisfies R1, R2' R3, R4 and R5, R7 and R8.*
2. *If the assignment is so-faithful (*i.e. *$\prec_\varphi$ is a semiorder) then then $\circ$ satisfies also R9 and R10.*
3. *If the assignment is p-KM-faithful (*i.e. *$\prec_\varphi$ is a min-partial order) then $\circ$ satisfies also R2.*
4. *If the assignment is faithful (*i.e. *$\prec_\varphi$ is a ranking order) then $\circ$ satisfies also R6.*

Proof. Point 1. In this case $\prec_\varphi$ is a partial order. R1 follows straightforwardly from the equation of representation. To see that R2' is true, it is enough to note that, by definition of assignment, $min(\prec_\varphi) = [\![\varphi]\!]$, *i.e.* by the representation equation $[\![\varphi \circ \top]\!] = [\![\varphi]\!]$. R3 follows from the fact that if $M \neq \emptyset$ then $min(M, \prec_\varphi) \neq \emptyset$. R4 follows straightforwardly from the equation of representation and the fact that $\varphi \mapsto \prec_\varphi$ is an assignment. R5 follows from the following fact which is very easy to check: $min(M, \prec_\varphi) \cap N \subseteq min(M \cap N, \prec_\varphi)$. In order to prove R7, assume that $(\varphi \circ \alpha) \vdash \beta$ and $\varphi \circ \beta \vdash \alpha$. We want to show that $\varphi \circ \alpha \equiv \varphi \circ \beta$. Put $A = [\![\alpha]\!]$ and $B = [\![\beta]\!]$. If $\omega \in min(A, \prec_\varphi)$, then by the assumptions, $\omega \in [\![B]\!]$. Towards a contradiction, suppose that $\omega \notin min(B, \prec_\varphi)$, then there exists $\omega' \in min(B, \prec_\varphi)$ such that $\omega' \prec_\varphi \omega$; but by assumption $\omega' \in A$, thus we have a contradiction with the fact that $\omega \in min(A, \prec_\varphi)$. Therefore $\omega \in min(B, \prec_\varphi)$ and then $[\![\varphi \circ \alpha]\!] \subseteq [\![\varphi \circ \beta]\!]$. In a similar way, we obtain $[\![\varphi \circ \beta]\!] \subseteq [\![\varphi \circ \alpha]\!]$.
Now we want to prove R8, *i.e.* $(\varphi\circ\alpha\wedge\varphi\circ\beta) \vdash \varphi\circ(\alpha\vee\beta)$. Put $A = [\![\alpha]\!]$ and $B = [\![\beta]\!]$. If $\omega \in min(A, \prec_\varphi) \cap min(B, \prec_\varphi)$, then $\omega \in A \cap B$, *i.e.* $\omega \in A$ and $\omega \in B$. If $\omega \notin$

$min(A \cup B)$, there exists $\omega' \in min(A \cup B)$ such that $\omega' \prec_\varphi \omega$. Since $\omega' \in A \cup B$, either $\omega' \in A$, contradicting the fact $\omega \in min(A, \prec_\varphi)$ or $\omega' \in B$, contradicting the fact $\omega \in min(B, \prec_\varphi)$. Therefore, $\omega \in min(A \cup B, \prec_\varphi)$.

Point 2. In this case $\prec_\varphi$ is a semiorder. In order to prove R9, we assume $(\varphi \circ \alpha) \wedge \beta \nvdash \varphi \circ \beta$ and we want to see that $(\varphi \circ \beta) \wedge \alpha \vdash \varphi \circ \alpha$ holds. Take $\omega \in \llbracket \varphi \circ \beta \rrbracket \cap \llbracket \alpha \rrbracket$. By assumption, there exists $\omega' \in \llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \beta \rrbracket$ such that $\omega' \notin \llbracket \varphi \circ \beta \rrbracket$. Then there exists $\omega'' \in \llbracket \varphi \circ \beta \rrbracket$ such that $\omega'' \prec_\varphi \omega'$ (*). Towards a contradiction, suppose $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$. Then there exists $\omega''' \in \llbracket \varphi \circ \alpha \rrbracket$ such that $\omega''' \prec_\varphi \omega''$ (**). From (*) and (**), by SO3 we have either $\omega''' \prec_\varphi \omega'$ or $\omega''' \prec_\varphi \omega'$. The first option contradicts the fact that $\omega' \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$; the second option contradicts the fact that $\omega \in min(\llbracket \beta \rrbracket, \prec_\varphi)$.
In order to prove R10, we assume $(\varphi \circ \alpha) \wedge \beta \vdash \bot$ and $(\varphi \circ \alpha) \wedge \gamma \nvdash \bot$. We want to show that $(\varphi \circ \gamma) \wedge (\alpha \wedge \beta) \vdash \varphi \circ (\alpha \wedge \beta)$ holds. Take $\omega \in \llbracket \varphi \circ \gamma \rrbracket \cap \llbracket \alpha \rrbracket \cap \llbracket \beta \rrbracket$. Towards a contradiction, suppose $\omega \notin \llbracket \varphi \circ (\alpha \wedge \beta) \rrbracket$, then there exists $\omega' \in \llbracket \varphi \circ (\alpha \wedge \beta) \rrbracket$ such that $\omega' \prec_\varphi \omega$. By assumption, $\llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \beta \rrbracket = \emptyset$, then $\omega' \notin \llbracket \varphi \circ \alpha \rrbracket$, therefore there exists $\omega'' \in \llbracket \varphi \circ \alpha \rrbracket$ such that $\omega'' \prec_\varphi \omega'$.
By the assumption $\llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \gamma \rrbracket \neq \emptyset$, there exists $\omega''' \in \llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \gamma \rrbracket$. Since $\omega'' \prec_\varphi \omega' \prec_\varphi \omega$, by S02, either $\omega'' \prec_\varphi \omega'''$ or $\omega''' \prec_\varphi \omega$. In the first case we have a contradiction with the fact that $\omega''' \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$; in the second case we have a contradiction with the fact that $\omega \in min(\llbracket \gamma \rrbracket, \prec_\varphi)$.

Point 3. In this case $\prec_\varphi$ is a min-partial order. We want to prove R2. Thus, assume $\varphi \wedge \alpha \nvdash \bot$. We want to show that $\varphi \circ \alpha \equiv \varphi \wedge \alpha$ holds. First we prove that $\llbracket \varphi \wedge \alpha \rrbracket \subseteq \llbracket \varphi \circ \alpha \rrbracket$). Take $\omega \in \llbracket \varphi \wedge \alpha \rrbracket$, then $\omega \in \llbracket \varphi \rrbracket \cap \llbracket \alpha \rrbracket$. Since $\prec_\varphi$ is a min-partial order and $\llbracket \varphi \rrbracket = min(\prec_\varphi)$, we have $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$, *i.e.* $\omega \in \llbracket \varphi \circ \alpha \rrbracket$.
Now we prove that $\llbracket \varphi \circ \alpha \rrbracket \subseteq \llbracket \varphi \wedge \alpha \rrbracket$). Take $\omega \in \llbracket \varphi \circ \alpha \rrbracket$ and towards a contradiction suppose $\omega \notin \llbracket \varphi \wedge \alpha \rrbracket$. By the representation equation, $\omega \in \llbracket \alpha \rrbracket$, thus $\omega \notin \llbracket \varphi \rrbracket$. From the assumption $\varphi \wedge \alpha \nvdash \bot$, there exists $\omega' \in \llbracket \varphi \wedge \alpha \rrbracket$, that is $\omega' \in \llbracket \varphi \rrbracket \cap \llbracket \alpha \rrbracket$. Since $\prec_\varphi$ is a min-partial order and $\llbracket \varphi \rrbracket = min(\prec_\varphi)$, we have $\omega' \prec_\varphi \omega$, therefore $\omega \notin min(\llbracket \alpha \rrbracket, \prec_\varphi)$, a contradiction.

Point 4. In this case $\prec_\varphi$ is a ranking order. In order to prove R6, assume that $(\varphi \circ \alpha) \wedge \beta \nvdash \bot$. We want to show that $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$ holds. Take $\omega \in \llbracket \varphi \circ (\alpha \wedge \beta) \rrbracket$. Towards a contradiction, suppose that $\omega \notin \llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \beta \rrbracket$. By the assumption and the representation equation $\omega \in \llbracket \beta \rrbracket$, thus, necessarily $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$. Therefore there exists $\omega' \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$ such that $\omega' \prec_\varphi \omega$. Note that by assumption there exists $\omega'' \in \llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \beta \rrbracket$. Necessarily $\omega'' \neq \omega'$; if not, $\omega'' \prec_\varphi \omega$, contradicting the fact that $\omega \in min(\llbracket \alpha \wedge \beta \rrbracket, \prec_\varphi)$. Thus, $\omega'' \sim \omega'$ because $\omega'', \omega' \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$. Therefore, since $\prec_\varphi$ is a ranking order and the fact that $\omega' \prec_\varphi \omega$, we have $\omega'' \prec_\varphi \omega$ contradicting again the fact that $\omega \in min(\llbracket \alpha \wedge \beta \rrbracket, \prec_\varphi)$. □

Proposition 2. *Assume that $\circ$ is an operator and $\varphi \mapsto \prec_\varphi$ is an assignment such that the following representation equation holds*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Then this assigment is unique with this property.

Proof. Towards a contradiction, suppose there are two assignments $\varphi \mapsto \prec_\varphi$ and $\varphi \mapsto \prec'_\varphi$ satisfying the representation equation. Then, there exists φ such that $\prec_\varphi \neq \prec'_\varphi$ and two interpretations ω, ω' such that (without loss of generality) $\omega \prec'_\varphi \omega'$ and either $\omega' \prec_\varphi \omega$ or $\omega' \sim \omega$. In the first case we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = min(\llbracket \alpha_{\omega\omega'} \rrbracket, \prec_\varphi) = \{\omega\}$ and $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = min(\llbracket \alpha_{\omega\omega'} \rrbracket, \prec'_\varphi) = \{\omega'\}$ a contradiction. In the second case we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = min(\llbracket \alpha_{\omega\omega'} \rrbracket, \prec_\varphi) = \{\omega\}$ and $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = min(\llbracket \alpha_{\omega\omega'} \rrbracket, \prec'_\varphi) = \{\omega, \omega'\}$ also a contradiction. □

Theorem 1 (Katsuno and Mendelzon, 91)**.** *The operator $\circ$ is a revision operator if and only if there exists a unique faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Proof. (*If part.*) We assume that there exists a faithful assignment $\varphi \mapsto \prec_\varphi$ such that the representation equation is satisfied. Since ranking orders are a particular case of partial orders by point 1 of Proposition 1, postulates R1, R3-R5 are satisfied. Moreover, ranking orders are special cases of min-partial orders, thus again, by Proposition 1 (point 3), postulate R2 is satisfied. Finally, by Proposition 1 (point 4), postulate R6 is satisfied.
(*Only if part.*) We assume that $\circ$ is a revision operator. Thus, postulates R1-R6 are satisfied. Define the map $\varphi \mapsto \prec_\varphi$ by putting

$$\omega \prec_\varphi \omega' \;\Leftrightarrow\; \omega \in \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \text{ and } \omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$$

We are going to prove that this map is a faithful assignment for which the representation equation is satisfied. Once these facts are established, the uniqueness of the assignment follows by Proposition 2.
By R4, it is clear that if $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec'_\varphi$.
By R2, if $\omega \in \llbracket \varphi \rrbracket$ and $\omega' \notin \llbracket \varphi \rrbracket$, we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega\}$, thus $\omega \prec_\varphi \omega'$. If $\omega, \omega' \in \llbracket \varphi \rrbracket$, by R2 we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega, \omega'\}$, thus $\omega \sim \omega'$. From these two facts we get that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$. Therefore this map is an assignment.
Now we are going to prove that $\prec_\varphi$ is a ranking order, *i.e.* a partial order such that for every $\omega_1, \omega_2, \omega_3 \in \Omega$ if $\omega_1 \sim \omega_2$ and $\omega_1 \prec \omega_3$ then $\omega_2 \prec \omega_3$. We begin by noting that the irreflexivity of $\prec_\varphi$ follows straightforwardly from its definition. Now, we turn to prove the transitivity of $\prec_\varphi$. First we prove that R6w holds. Assume that $\varphi \circ \alpha \vdash \beta$. We want to prove that $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$. If $\varphi \circ \alpha \vdash \bot$, then, by R3, $\alpha \vdash \bot$. Therefore $(\alpha \wedge \beta) \vdash \bot$. Then, by R1, $\varphi \circ (\alpha \wedge \beta) \vdash \bot$. Therefore, $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$. If $\varphi \circ \alpha \nvdash \bot$, then, by the assumption, $(\varphi \circ \alpha) \wedge \beta \nvdash \bot$. Then, by R6, we have $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$. Thus, we have proved that R6w is satisfied and by Lemma 1 we have Rt. Now suppose that $\omega_1 \prec_\varphi \omega_2 \prec_\varphi \omega_3$, that is $\llbracket \varphi \circ \alpha_{\omega_1\omega_2} \rrbracket = \{\omega_1\}$, $\llbracket \varphi \circ \alpha_{\omega_2\omega_3} \rrbracket = \{\omega_2\}$. Thus by R4 we have $\varphi \circ (\alpha_{\omega_1} \vee \alpha_{\omega_2}) = \alpha_{\omega_1}$ and $\varphi \circ (\alpha_{\omega_2} \vee \alpha_{\omega_3}) = \alpha_{\omega_2}$.

Then, by Rt, we have $\varphi \circ (\alpha_{\omega_1} \vee \alpha_{\omega_3}) = \alpha_{\omega_1}$. From this we obtain easily, $\omega_1 \prec_\varphi \omega_3$.
We prove now that $\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$. First we prove the inclusion
$(\llbracket \varphi \circ \alpha \rrbracket \subseteq min(\llbracket \alpha \rrbracket, \prec_\varphi))$. Take $\omega \in \llbracket \varphi \circ \alpha \rrbracket$ and, towards a contradiction, suppose $\omega \notin min(\llbracket \alpha \rrbracket, \prec_\varphi)$. By R1, $\omega \in \llbracket \alpha \rrbracket$, thus, there exists $\omega' \in \llbracket \alpha \rrbracket$ such that $\omega' \prec_\varphi \omega$. Thus, $\{\omega'\} = \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$. But $\alpha \wedge \alpha_{\omega,\omega'} \equiv \alpha_{\omega,\omega'}$, then, by R5, we have $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'} \vdash \varphi \circ (\alpha \wedge \alpha_{\omega,\omega'})$. By R4, we get $\llbracket \varphi \circ \alpha \rrbracket \cap \{\omega, \omega'\} \subseteq \{\omega'\}$ therefore $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$, a contradiction.
$(min(\llbracket \alpha \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \alpha \rrbracket)$. Take $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$ and, towards a contradiction, suppose $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$. By R3, there exists $\omega' \in \llbracket \varphi \circ \alpha \rrbracket$. Note that $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'}$ is consistent, thus, by R5, R6 and R4, we have $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'} \equiv \varphi \circ \alpha_{\omega,\omega'}$. Then, $\llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket = \{\omega'\}$. Therefore $\omega' \prec_\varphi \omega$, a contradiction.
Now we are going to prove that the property characterizing ranking orders holds. Thus, assume $\omega_1 \sim \omega_2$ and $\omega_1 \prec_\varphi \omega_3$. We want to show that $\omega_2 \prec_\varphi \omega_3$. Towards a contradiction, suppose that it is not the case that $\omega_2 \prec_\varphi \omega_3$. Then, by R1 and R3, we have two possibilities: either $\llbracket \varphi \circ \alpha_{\omega_2\omega_3} \rrbracket = \{\omega_3\}$ or $\llbracket \varphi \circ \alpha_{\omega_2\omega_3} \rrbracket = \{\omega_2, \omega_3\}$. In the first case, we obtain $\omega_3 \prec_\varphi \omega_2$. From this, by the assumption and transitivity, we get $\omega_1 \prec_\varphi \omega_2$, a contradiction. In the second case we obtain $\omega_2 \sim \omega_3$. Then, by the assumptions $min(\{\omega_1, \omega_2, \omega_3\}, \prec_\varphi) = \{\omega_1, \omega_2\}$. By the representation equation and R4, we have $\llbracket \varphi \circ \alpha_{\omega_1\omega_2\omega_3} \rrbracket = \{\omega_1, \omega_2\}$. Thus, $(\varphi \circ \alpha_{\omega_1\omega_2\omega_3}) \wedge \alpha_{\omega_2\omega_3}$ is consistent. By R5, R6 and R4 we have $(\varphi \circ \alpha_{\omega_1\omega_2\omega_3}) \wedge \alpha_{\omega_2\omega_3} \equiv \varphi \circ \alpha_{\omega_2\omega_3}$. Thus, $\llbracket \varphi \circ \alpha_{\omega_2\omega_3} \rrbracket = \{\omega_2\}$, contradicting our supposition that $\llbracket \varphi \circ \alpha_{\omega_2\omega_3} \rrbracket = \{\omega_2, \omega_3\}$. □

Theorem 2 (Katsuno and Mendelzon, 91)**.** *The operator $\circ$ is a p-KM-revision operator if and only if there exists a unique p-KM-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Proof. (*If part*). Note that a p-KM-faithful assignment is in particular a p-faithful assignment. Then by Proposition 1, Point 1, R1, R3-R5 and R7 and R8 hold. By Proposition 1, Point 3, the postulate R2 holds. Then, $\circ$ is a p-KM-revision operator.
(*Only if part*). We assume that $\circ$ is a p-KM-revision operator, *i.e.* R1-R5, R7 and R8 hold. We define the map $\varphi \mapsto \prec_\varphi$ by putting

$$\omega \prec_\varphi \omega' \;\Leftrightarrow\; \begin{cases} \omega \in \llbracket \varphi \rrbracket \text{ and } \omega' \notin \llbracket \varphi \rrbracket \quad \text{or} \\ \\ \omega \in \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \text{ and } \omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \end{cases}$$

We are going to prove that this map is a p-KM-faithful assignment for which the representation equation is satisfied. Once these facts are established, the uniqueness of the assignment follows from Proposition 2.
By R4, it is clear that if $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec'_\varphi$.
By R2, if $\omega \in \llbracket \varphi \rrbracket$ and $\omega' \notin \llbracket \varphi \rrbracket$, we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega\}$, thus $\omega \prec_\varphi \omega'$. If $\omega, \omega' \in \llbracket \varphi \rrbracket$, by R2 we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega, \omega'\}$, thus $\omega \sim \omega'$. From these two facts we get that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$.
Therefore this map is an assignment.
We have to show that $\prec_\varphi$ is a min-partial order and the representation equation. First we see that $\prec_\varphi$ is a partial order.
Irreflexivity: Straightforward by definition of $\prec_\varphi$.
Transitivity: By Lemma 1, Rt holds. Then we apply exactly the same argument to prove the transitivity in Theorem 1.
Now, we know that $\prec_\varphi$ is a partial order. By definition and the fact that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$, it is clear that $\prec_\varphi$ is a min-partial order.
We are going to show the representation equation $\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$. We begin by proving the inclusion $(\llbracket \varphi \circ \alpha \rrbracket \subseteq min(\llbracket \alpha \rrbracket, \prec_\varphi))$.
Take $\omega \in \llbracket \varphi \circ \alpha \rrbracket$ and, towards a contradiction suppose $\omega \notin min(\llbracket \alpha \rrbracket, \prec_\varphi)$. By R1, $\omega \in \llbracket \alpha \rrbracket$, thus, there exists $\omega' \in \llbracket \alpha \rrbracket$ such that $\omega' \prec_\varphi \omega$. Thus, $\{\omega'\} = \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$. But $\alpha \wedge \alpha_{\omega,\omega'} \equiv \alpha_{\omega,\omega'}$, then, by R5, we have $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'} \vdash \varphi \circ (\alpha \wedge \alpha_{\omega,\omega'})$. By R4, we get $\llbracket \varphi \circ \alpha \rrbracket \cap \{\omega, \omega'\} \subseteq \{\omega'\}$ therefore $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$, a contradiction.
$(min(\llbracket \alpha \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \alpha \rrbracket)$.
Take $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$. Let $\llbracket \alpha \rrbracket = \{\omega_1, \cdots \omega_k\}$.
Note that $\alpha \equiv (\alpha_{\omega,\omega_1} \vee \alpha_{\omega,\omega_2} \vee \cdots \vee \alpha_{\omega,\omega_k})$ (*). Take $\omega_j \in \llbracket \alpha \rrbracket$, by R1 $\llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket \subseteq \{\omega, \omega_j\}$. Since ω is minimal in $\llbracket \alpha \rrbracket$ we have $\omega_j \not\prec_\varphi \omega$. Thus, $\llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket \neq \{\omega_j\}$. Therefore $\omega \in \llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket$ for every $\omega_j \in \llbracket \alpha \rrbracket$. Then, $\omega \in \llbracket (\varphi \circ \alpha_{\omega,\omega_1}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k}) \rrbracket$.
Thus, by an application of R8 $k-1$ times, we get $\omega \in \llbracket \varphi \circ ((\alpha_{\omega,\omega_1}) \vee \cdots \vee (\alpha_{\omega,\omega_k})) \rrbracket$ (**). Then, from (*), (**) and R4, we obtain $\omega \in \llbracket \varphi \circ \alpha \rrbracket$. □

Theorem 3. *The operator $\circ$ is a p-revision operator if and only if there exists a unique p-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Proof. (*If part*). By Proposition 1 (Point 1), R1,R2', R3-R5, R7 and R8 hold. Then, $\circ$ is a p-revision operator.

(*Only if part*). Assume that $\circ$ is a p-revision operator, *i.e.* R1, R2', R3-R5, R7 and R8 hold . We define the map $\varphi \mapsto \prec_\varphi$ by putting

$$\omega \prec_\varphi \omega' \;\Leftrightarrow\; \omega \in \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \text{ and } \omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$$

We are going to prove that this map is a p-faithful assignment for which the representation equation is satisfied. Once these facts established, the uniqueness of the assignment follows from Proposition 2.
By R4, it is clear that if $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec'_\varphi$.
Let us see that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$.
$(\llbracket \varphi \rrbracket \subseteq min(\prec_\varphi))$. Take $\omega \in \llbracket \varphi \rrbracket$ and towards a contradiction, suppose $\omega \notin min(\prec_\varphi)$. Therefore, there exists $\omega' \in min(\prec_\varphi)$ such that $\omega' \prec_\varphi \omega$. Thus $min(\llbracket \alpha_{\omega,\omega'} \rrbracket, \prec_\varphi) = \llbracket \alpha_{\omega'} \rrbracket$. By R5, $(\varphi \circ \top) \wedge (\alpha_{\omega,\omega'}) \vdash \varphi \circ (\top \wedge \alpha_{\omega,\omega'})$. From R1 and R4, we get $\varphi \circ (\top \wedge \alpha_{\omega,\omega'}) \vdash \alpha_{\omega'}$, thus $(\varphi \circ \top) \wedge (\alpha_{\omega,\omega'}) \vdash \alpha_{\omega'}$ (*).
By R2' $(\varphi \circ \top) \wedge \alpha_{\omega,\omega'} \equiv \varphi \wedge \alpha_{\omega,\omega'}$. Since $\omega \in \llbracket \varphi \wedge \alpha_{\omega,\omega'} \rrbracket$, from (*), we obtain $\omega \in \llbracket \alpha_{\omega'} \rrbracket$, a contradiction. Thus, $\omega \in min(\prec_\varphi)$.
$(min(\prec_\varphi) \subseteq \llbracket \varphi \rrbracket)$. Suppose $\llbracket \top \rrbracket = \{\omega_1, \cdots, \omega_k\}$ and take $\omega \in min(\prec_\varphi)$. Then, $\omega \in min(\llbracket \top \rrbracket, \prec_\varphi)$. Note that

$$\top \equiv (\alpha_{\omega,\omega_1}) \vee (\alpha_{\omega,\omega_2}) \vee \cdots \vee (\alpha_{\omega,\omega_k})$$

Note that $\omega \in \llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket$ for every $\omega_j \in \llbracket \top \rrbracket$ such that $\omega \neq \omega_j$, because ω is a minimal of $\llbracket \top \rrbracket$. Thus,

$$\omega \in \llbracket (\varphi \circ \alpha_{\omega,\omega_1}) \wedge (\varphi \circ \alpha_{\omega,\omega_2}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k}) \rrbracket$$

Applying k-times R8, we have $\omega \in \llbracket \varphi \circ (\alpha_{\omega,\omega_1} \vee \alpha_{\omega,\omega_2} \vee \cdots \vee \alpha_{\omega,\omega_k}) \rrbracket$. That is, by R4, $\omega \in \llbracket \varphi \circ \top \rrbracket$. But $\llbracket \varphi \circ \top \rrbracket = \llbracket \varphi \rrbracket$, thus, $\omega \in \varphi$.
We have proved that the map is indeed an assignment.

Now we are going to verify that the assignment is a p-faithful assignment. Actually we have to check that $\prec_\varphi$ is a partial order.
Irreflexivity: Straightforward by definition of $\prec_\varphi$.
Transitivity: By Lemma 1, Rt holds. Then we apply exactly the same argument used to prove the transitivity in Theorem 1.
Let us prove the representation equation $\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$. We begin by proving the inclusion $(\llbracket \varphi \circ \alpha \rrbracket \subseteq min(\llbracket \alpha \rrbracket, \prec_\varphi))$.
Take $\omega \in \llbracket \varphi \circ \alpha \rrbracket$ and, towards a contradiction suppose $\omega \notin min(\llbracket \alpha \rrbracket, \prec_\varphi)$. By R1, $\omega \in \llbracket \alpha \rrbracket$, thus, there exists $\omega' \in \llbracket \alpha \rrbracket$ such that $\omega' \prec_\varphi \omega$. Thus, $\{\omega'\} = \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$. But $\alpha \wedge \alpha_{\omega,\omega'} \equiv \alpha_{\omega,\omega'}$, then, by R5, we have $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'} \vdash \varphi \circ (\alpha \wedge \alpha_{\omega,\omega'})$. By R4, we get $\llbracket \varphi \circ \alpha \rrbracket \cap \{\omega, \omega'\} \subseteq \{\omega'\}$ therefore $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$, a contradiction.
$(min(\llbracket \alpha \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \alpha \rrbracket)$.
Take $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$. Let $\llbracket \alpha \rrbracket = \{\omega_1, \cdots \omega_k\}$.
Note that $\alpha \equiv (\alpha_{\omega,\omega_1} \vee \alpha_{\omega,\omega_2} \vee \cdots \vee \alpha_{\omega,\omega_k})$ (*). Take $\omega_j \in \llbracket \alpha \rrbracket$, by R1 $\llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket \subseteq \{\omega, \omega_j\}$. Since ω is minimal in $\llbracket \alpha \rrbracket$ we have $\omega_j \not\prec_\varphi \omega$. Thus, $\llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket \neq \{\omega_j\}$. Therefore $\omega \in \llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket$ for every $\omega_j \in \llbracket \alpha \rrbracket$. Then, $\omega \in \llbracket (\varphi \circ \alpha_{\omega,\omega_1}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k}) \rrbracket$.
Thus, by an application of R8 $k-1$ times, we get $\omega \in \llbracket \varphi \circ ((\alpha_{\omega,\omega_1}) \vee \cdots \vee (\alpha_{\omega,\omega_k})) \rrbracket$ (**). Then, from (*), (**) and R4, we obtain $\omega \in \llbracket \varphi \circ \alpha \rrbracket$.

□

Theorem 4. *The operator $\circ$ is a so-PW-revision operator if and only if there exists a unique so-PW-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Proof. (*If part*). Note that a so-PW-faithful assignment is in particular a p-KM-faithful assignment. Thus, by Proposition 1, Points 1 and 3, R1-R5, and R8 hold. Since a so-PW-faithful assignment is in particular a so-faithful assignment, by Proposition 1, Point 2, postulate R9 and R10 hold Then, $\circ$ is a so-PW-revision operator.

(*Only if part*). As the operator $\circ$ is a so-PW-revision operator, the postulates R1-R5 plus R8-R10 hold. We define the map $\varphi \mapsto \prec_\varphi$ by putting

$$\omega \prec_\varphi \omega' \;\Leftrightarrow\; \begin{cases} \omega \in \llbracket \varphi \rrbracket \text{ and } \omega' \notin \llbracket \varphi \rrbracket \quad \text{or} \\ \\ \omega \in \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \text{ and } \omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket \end{cases}$$

We are going to prove that this map is a so-PW-faithful assignment for which the representation equation is satisfied. Once these facts established, the uniqueness of the assignment follows from Proposition 2.

By R4, it is clear that if $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec'_\varphi$.
Let us see that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$.
By R2, if $\omega \in \llbracket \varphi \rrbracket$ and $\omega' \notin \llbracket \varphi \rrbracket$, we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega\}$, thus $\omega \prec_\varphi \omega'$. If $\omega, \omega' \in \llbracket \varphi \rrbracket$, by R2 we have $\llbracket \varphi \circ \alpha_{\omega\omega'} \rrbracket = \llbracket \varphi \rrbracket \cap \llbracket \alpha_{\omega\omega'} \rrbracket = \{\omega, \omega'\}$, thus $\omega \sim \omega'$. From these two facts we get that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$.
Therefore this map is an assignment.
It remains to prove that $\prec_\varphi$ is a min-semiorder. Clearly, by definition, $\prec_\varphi$ is irreflexive. Thus, SO1 is satisfied.
Let us prove the representation equation $\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$.
We begin by proving the inclusion $(\llbracket \varphi \circ \alpha \rrbracket \subseteq min(\llbracket \alpha \rrbracket, \prec_\varphi))$.
Take $\omega \in \llbracket \varphi \circ \alpha \rrbracket$ and, towards a contradiction suppose $\omega \notin min(\llbracket \alpha \rrbracket, \prec_\varphi)$. By R1, $\omega \in \llbracket \alpha \rrbracket$, thus, there exists $\omega' \in \llbracket \alpha \rrbracket$ such that $\omega' \prec_\varphi \omega$. Thus, $\{\omega'\} = \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$. But $\alpha \wedge \alpha_{\omega,\omega'} \equiv \alpha_{\omega,\omega'}$, then, by R5, we have $(\varphi \circ \alpha) \wedge \alpha_{\omega,\omega'} \vdash \varphi \circ (\alpha \wedge \alpha_{\omega,\omega'})$. By R4, we get $\llbracket \varphi \circ \alpha \rrbracket \cap \{\omega, \omega'\} \subseteq \{\omega'\}$ therefore $\omega \notin \llbracket \varphi \circ \alpha \rrbracket$, a contradiction.

Now we want to prove $(min(\llbracket \alpha \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \alpha \rrbracket)$.
We proceed by induction over the size of $\llbracket \alpha \rrbracket$. If $|\llbracket \alpha \rrbracket| = 1$, then $\llbracket \alpha \rrbracket = \{\omega\}$.

By R3, $\omega \in \llbracket \varphi \circ \alpha \rrbracket$. Since ω is the unique model of α, it is the unique minimal. Therefore $min(\llbracket \alpha \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \alpha \rrbracket$.
Our induction hypothesis is that for every $\gamma \in \mathcal{L}$ such that $|\llbracket \gamma \rrbracket| \leq k$ we have $min(\llbracket \gamma \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \gamma \rrbracket$. Let α be a formula such that $|\llbracket \alpha \rrbracket| = k+1$. Take $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$; we want to show that $\omega \in \llbracket \varphi \circ \alpha \rrbracket$.
Let ω' be in $\llbracket \alpha \rrbracket$ such that $\omega \neq \omega'$. Define $\beta = \alpha \wedge \neg \alpha_{\omega'}$. Note that $\llbracket \beta \rrbracket = \llbracket \alpha \rrbracket \setminus \{\omega'\}$. Therefore $|\llbracket \beta \rrbracket| = k$. Then by the induction hypothesis

$$min(\llbracket \beta \rrbracket, \prec_\varphi) \subseteq \llbracket \varphi \circ \beta \rrbracket$$

Since $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$ necessarily $\omega \in min(\llbracket \beta \rrbracket, \prec_\varphi$). Therefore, $\omega \in \llbracket \varphi \circ \beta \rrbracket$.
Note that $\omega \in min(\llbracket \alpha \rrbracket, \prec_\varphi)$ and $\omega' \in \llbracket \alpha \rrbracket$, then $\omega' \not\prec_\varphi \omega$. Therefore $\omega \in \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$.
By R8 we have

$$(\varphi \circ \beta) \wedge (\varphi \circ \alpha_{\omega,\omega'}) \vdash \varphi \circ (\beta \vee \alpha_{\omega,\omega'})$$

Note that $\omega \in \llbracket \varphi \circ \beta \rrbracket \cap \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$ and $\beta \vee \alpha_{\omega,\omega'} \equiv \alpha$. Therefore $\omega \in \llbracket \varphi \circ \alpha \rrbracket$.

We show that $\prec_\varphi$ is transitive.
Suppose $\omega_1 \prec_\varphi \omega_2$ and $\omega_2 \prec_\varphi \omega_3$. We want to see that $\omega_1 \prec_\varphi \omega_3$.
Put $\alpha = \alpha_{\omega_1,\omega_2,\omega_3}$. From the assumption

$$min(\llbracket \alpha_{\omega_1,\omega_2,\omega_3} \rrbracket, \prec_\varphi) = \{\omega_1\}$$

From the representation equation and R4, we have $\llbracket \varphi \circ \alpha \rrbracket = \{\omega_1\}$. Thus,

$$\llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \alpha_{\omega_2,\omega_3} \rrbracket = \emptyset$$

$$\llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \alpha_{\omega_1,\omega_3} \rrbracket \neq \emptyset$$

Put $\beta = \alpha_{\omega_2\omega_3}$ and $\gamma = \alpha_{\omega_1\omega_3}$. Then $\varphi \circ \alpha \vdash \neg\beta$ and $(\varphi \circ \alpha) \wedge \gamma \nvdash \bot$. Then, by R10,

$$(\varphi \circ \gamma) \wedge (\alpha \wedge \beta) \vdash \varphi \circ (\alpha \wedge \beta)$$

Therefore $\llbracket \varphi \circ \alpha_{\omega_1,\omega_3} \rrbracket \cap \llbracket \alpha_{\omega_2,\omega_3} \rrbracket \subseteq \llbracket \varphi \circ \alpha_{\omega_2,\omega_3} \rrbracket$. Since $\omega_2 \prec_\varphi \omega_3$ by the representation equation $\prec_\varphi$ we have $\omega_3 \notin \llbracket \varphi \circ \alpha_{\omega_2,\omega_3} \rrbracket$. Thus, necessarily, $\omega_3 \notin \llbracket \varphi \circ \alpha_{\omega_1,\omega_3} \rrbracket$. Therefore, $\omega_1 \prec_\varphi \omega_3$.

We prove SO2. Suppose $\omega_1 \prec_\varphi \omega_2 \prec_\varphi \omega_3$ and $\omega' \in \Omega$. We want to show that $\omega_1 \prec_\varphi \omega'$ or $\omega' \prec_\varphi \omega_3$ Take $\omega' \in \Omega$ and towards a contradiction suppose $\omega_1 \not\prec_\varphi \omega'$ and $\omega' \not\prec_\varphi \omega_3$.
- If $\omega_2 \prec_\varphi \omega'$, since $\omega_1 \prec_\varphi \omega_2$, by transitivity $\omega_1 \prec_\varphi \omega'$, a contradiction. Then, $\omega_2 \not\prec_\varphi \omega'$.
- If $\omega_3 \prec_\varphi \omega'$, since $\omega_2 \prec_\varphi \omega_3$, by transitivity $\omega_2 \prec_\varphi \omega'$, a contradiction. Then, $\omega_3 \not\prec_\varphi \omega'$.
Thus, $\omega_1 \not\prec_\varphi \omega'$, $\omega_2 \not\prec_\varphi \omega'$, $\omega_3 \not\prec_\varphi \omega'$. Therefore, $\omega' \in min(\llbracket \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket, \prec_\varphi)$ and by the representation equation, $\omega' \in \llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket$.

Since $\omega_1 \prec_\varphi \omega_2$, we have $\omega_2 \notin \llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket$. Since $\omega_2 \prec_\varphi \omega_3$, we have $\omega_3 \notin \llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket$. Thus, $\llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket \cap \llbracket \alpha_{\omega_2,\omega_3} \rrbracket = \emptyset$ and $\llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_3,\omega'} \rrbracket \cap \llbracket \alpha_{\omega_3,\omega'} \rrbracket \neq \emptyset$.
Define $\alpha \equiv \alpha_{\omega_1,\omega_2,\omega_3,\omega'}$, $\beta \equiv \alpha_{\omega_2,\omega_3}$ and $\gamma \equiv \alpha_{\omega_3,\omega'}$. Since $(\varphi \circ \alpha) \vdash \neg\beta$ and $(\varphi \circ \alpha) \wedge \gamma \nvdash \bot$, by R10, we get

$$(\varphi \circ \gamma) \wedge (\alpha \wedge \beta) \vdash \varphi \circ (\alpha \wedge \beta)$$

By R4 $(\varphi \circ \alpha_{\omega_3,\omega'}) \wedge (\alpha_{\omega_2\omega_3}) \vdash \varphi \circ \alpha_{\omega_2\omega_3}$. Since $\omega_2 \prec_\varphi \omega_3$, we have $\omega_3 \notin \llbracket \varphi \circ \alpha_{\omega_2,\omega_3} \rrbracket$. Therefore $\omega_3 \notin \llbracket \varphi \circ \alpha_{\omega_3,\omega'} \rrbracket$, *i.e.* $\omega' \prec_\varphi \omega_3$, contradicting our assumptions. Therefore $\omega_1 \prec_\varphi \omega'$ or $\omega' \prec_\varphi \omega_3$.
We show S03. Suppose we have $\omega_1 \prec_\varphi \omega_2$ and $\omega_3 \prec_\varphi \omega_4$. We want to prove $\omega_1 \prec_\varphi \omega_4$ o $\omega_3 \prec_\varphi \omega_2$

Towards a contradiction, suppose $\omega_1 \not\prec_\varphi \omega_4$ and $\omega_3 \not\prec_\varphi \omega_2$.
- If $\omega_2 \prec_\varphi \omega_4$, since $\omega_1 \prec_\varphi \omega_2$, by transitivity $\omega_1 \prec_\varphi \omega_4$, a contradiction. Therefore $\omega_2 \not\prec_\varphi \omega_4$.
- If $\omega_4 \prec_\varphi \omega_2$, since $\omega_3 \prec_\varphi \omega_4$, by transitivity $\omega_3 \prec_\varphi \omega_2$, a contradiction. Therefore $\omega_4 \not\prec_\varphi \omega_2$.
Since, $\omega_1 \not\prec_\varphi \omega_4$, $\omega_2 \not\prec_\varphi \omega_4$ and $\omega_4 \not\prec_\varphi \omega_4$. By the representation equation $\omega_4 \in \llbracket \varphi \circ \alpha_{\omega_1\omega_2\omega_4} \rrbracket$, since $\omega_3 \prec_\varphi \omega_4$, we have $\omega_4 \notin \llbracket \varphi \circ \alpha_{\omega_2\omega_3\omega_4} \rrbracket$.
Define $\alpha \equiv \alpha_{\omega_1,\omega_2,\omega_4}$ and $\beta \equiv \alpha_{\omega_2,\omega_3,\omega_4}$. Note that $(\varphi \circ \alpha) \wedge \beta \nvdash \varphi \circ \beta$ because $\omega_4 \in \llbracket \varphi \circ \alpha \rrbracket \cap \llbracket \beta \rrbracket$, but $\omega_4 \notin \llbracket \varphi \circ \beta \rrbracket$. By R9, we get $(\varphi \circ \beta) \wedge \alpha \vdash \varphi \circ \alpha$. Then

$$(\varphi \circ \alpha_{\omega_2,\omega_3,\omega_4}) \wedge (\alpha_{\omega_1,\omega_2,\omega_4}) \vdash (\varphi \circ \alpha_{\omega_1,\omega_2,\omega_4})$$

Since $\omega_3 \not\prec_\varphi \omega_2$, $\omega_4 \not\prec_\varphi \omega_2$ we have $\omega_2 \in \llbracket \varphi \circ \alpha_{\omega_2,\omega_3,\omega_4} \rrbracket$. Therefore $\omega_2 \in \llbracket \varphi \circ \alpha_{\omega_1,\omega_2,\omega_4} \rrbracket$ but this, via the representation equation, contradicts the fact that $\omega_1 \not\prec_\varphi \omega_2$.
Thus, we have proved that $\prec_\varphi$ is a semiorder. It remains to show that it is min-semiorder. Since $\llbracket \varphi \rrbracket = min(\prec_\varphi)$, it is enough to prove that if $\omega \in \llbracket \varphi \rrbracket$ and $\omega' \notin \llbracket \varphi \rrbracket$ then $\omega \prec_\varphi \omega'$.
Put $\omega \in \llbracket \varphi \rrbracket$, $\omega' \notin \llbracket \varphi \rrbracket$ and $\alpha_{\omega,\omega'} \in \mathcal{L}$. Since $\varphi \wedge \alpha_{\omega,\omega'} \nvdash \bot$, by R2, we have $\varphi \circ \alpha_{\omega,\omega'} \equiv \varphi \wedge \alpha_{\omega,\omega'}$. Thus, $\omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$, therefore $\omega \prec_\varphi \omega'$.

□

Theorem 5. *The operator $\circ$ is a so-revision operator if and only if there exists a unique so-faithful assignment $\varphi \mapsto \prec_\varphi$ such that*

$$\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$$

Proof. (*If part*). Note that a so-faithful assignment is in particular a p-faithful assignment. Thus, by Proposition 1 (Point 1), R1, R2', R3-R5, and R8 hold. By Proposition 1 (Point 2), postulate R9 and R10 hold. Then, $\circ$ is a so-revision operator.

(*Only if part*). As the operator $\circ$ is a so-revision operator, the postulates R1, R2', R3-R5 plus R8-R10 hold. We define the map $\varphi \mapsto \prec_\varphi$ by putting

$$\omega \prec_\varphi \omega' \Leftrightarrow \omega' \notin \llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket$$

We are going to prove that this map is a so-faithful assignment for which the representation equation is satisfied. Once these facts established, the uniqueness of the assignment follows by Proposition 2.
By R4, it is clear that if $\varphi \equiv \varphi'$ then $\prec_\varphi = \prec'_\varphi$.
The representation equation, $\llbracket \varphi \circ \alpha \rrbracket = min(\llbracket \alpha \rrbracket, \prec_\varphi)$, has the same proof as in the previous theorem.
Also the transitivity of $\prec_\varphi$ is exactly as in the previous theorem.
Let us see that $\llbracket \varphi \rrbracket = min(\prec_\varphi)$.
Take $\omega \in \llbracket \varphi \rrbracket$. If $\omega \notin min(\prec_\varphi)$ there exists $\omega' \in min(\prec_\varphi)$ such that $\omega' \prec_\varphi \omega$.
Then,

$$\llbracket \varphi \circ \alpha_{\omega,\omega'} \rrbracket = min(\llbracket \alpha_{\omega,\omega'} \rrbracket, \prec_\varphi) = \{\omega'\}$$

By R5:

$$(\varphi \circ \top) \wedge (\alpha_{\omega,\omega'}) \vdash \varphi \circ (\top \wedge \alpha_{\omega,\omega'})$$

By R2' $(\varphi \circ \top) \equiv \varphi$ and since $(\top \wedge \alpha_{\omega,\omega'}) \equiv \alpha_{\omega,\omega'}$ and $\varphi \circ \alpha_{\omega,\omega'} \equiv \alpha_{\omega'}$, by R4 we get

$$\varphi \wedge \alpha_{\omega,\omega'} \vdash \alpha_{\omega'}$$

Since $\omega \in \llbracket \varphi \rrbracket$ and $\omega \in \llbracket \alpha_{\omega,\omega'} \rrbracket$, necessarily $\omega \in \llbracket \alpha_{\omega'} \rrbracket$, a contradiction. Therefore, $\omega \in min(\prec_\varphi$.

Now we proceed to see that $min(\prec_\varphi) \subseteq \llbracket \varphi \rrbracket$.
Put $\llbracket \top \rrbracket = \{\omega_1, \cdots \omega_k\}$ and take $\omega \in min(\llbracket \top \rrbracket, \prec_\varphi)$. By R2':

$$min(\llbracket \top \rrbracket, \prec_\varphi) = min(\llbracket \varphi \rrbracket, \prec_\varphi)$$

Note that $\top \equiv (\alpha_{\omega,\omega_1}) \vee (\alpha_{\omega,\omega_2}) \vee \cdots \vee (\alpha_{\omega,\omega_k})$. Moreover, $\omega \in \llbracket \varphi \circ \alpha_{\omega,\omega_j} \rrbracket$ for every $\omega_j \in \llbracket \top \rrbracket$. Thus an iterated use of R8, leads us to $[(\varphi \circ \alpha_{\omega,\omega_1}) \wedge (\varphi \circ \alpha_{\omega,\omega_2}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k})] \vdash \varphi \circ (\alpha_{\omega,\omega_1} \vee \alpha_{\omega,\omega_2} \vee \cdots \vee \alpha_{\omega,\omega_k})$ By R4,

$$[(\varphi \circ \alpha_{\omega,\omega_1}) \wedge (\varphi \circ \alpha_{\omega,\omega_2}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k})] \vdash \varphi \circ \top.$$

By R2' $\varphi \circ \top \equiv \varphi$ and by R4 again $[(\varphi \circ \alpha_{\omega,\omega_1}) \wedge (\varphi \circ \alpha_{\omega,\omega_2}) \wedge \cdots \wedge (\varphi \circ \alpha_{\omega,\omega_k})] \vdash \varphi$ Therefore, $\omega \in \llbracket \varphi \rrbracket$.

Finally the proof that the axioms of semiorder are satisfied by $\prec_\varphi$ is exactly the same as in the previous theorem.

□

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50:510–530.

Benferhat, S.; Lagrue, S.; and Papini, O. 2005. Revision of partially ordered information: Axiomatization, semantics and iteration. In Kaelbling, L. P., and Saffiotti, A., eds., *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, 376–381. Professional Book Center.

Gärdenfors, P. 1988. *Knowledge in flux*. MIT Press.

Katsuno, H., and Mendelzon, A. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52:263–294.

Peppas, P., and Williams, M. 2014. Belief change and semiorders. In Baral, C.; Giacomo, G. D.; and Eiter, T., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press.

Pirlot, M., and Vincke, P. 1997. *Semiorders*. Springer.

Surprise Minimization Revision Operators

Adrian Haret
Institute for Logic, Language and Computation
The University of Amsterdam
a.haret@uva.nl

Abstract

Prominent approaches to belief revision prescribe the adoption of a new belief that is as close as possible to the prior belief, in a process that, even in the standard case, can be described as attempting to minimize surprise. Here we extend the existing model by proposing a measure of surprise, dubbed *relative surprise*, in which surprise is computed with respect not just to the prior belief, but also to the broader context provided by the new information, using a measure derived from familiar distance notions between truth-value assignments. We characterize the surprise minimization revision operator thus defined using a set of intuitive rationality postulates in the AGM mould, along the way obtaining representation results for other existing revision operators in the literature, such as the Dalal operator and a recently introduced distance-based min-max operator.

1 Introduction

Belief change models rational adjustments made to an agent's epistemic state upon acquiring new information (Peppas 2008; Hansson 2017; Fermé and Hansson 2018). When the new information is assumed to be reliable, the logic of changing one's prior beliefs to accommodate such new-found knowledge falls under the heading of *revision*. Belief revision is typically thought of by appeal to a set of intuitive normative principles, usually along the lines of the AGM framework (Alchourrón, Gärdenfors, and Makinson 1985), alongside more concrete revision representations and mechanisms (Grove 1988; Dalal 1988; Gärdenfors and Makinson 1988; Katsuno and Mendelzon 1992; Rott 1992).

A perspective underlying many of these representations, which we share here, is that belief revision is akin to a choice procedure guided by a plausibility relation over possible states of affairs: revising a belief, in this sense, amounts to choosing the most plausible states of affairs consistent with the new information. Plausibility over states of affairs, in turn, is judged according to some notion of dissimilarity, or distance between states of affairs: I judge a situation to be less likely the further away from my own belief it is. Among the various distance notions that can be used to make this intuition precise, the approach using Hamming distance to rank truth-value assignments is among the most prominent, used for the well-known Dalal revision operator (Dalal 1988), and the more recently introduced Hamming distance min-max operator (Haret and Woltran 2019).

Both the Dalal and the Hamming distance min-max operator are designed to respond to new information by minimizing departures from the prior belief, in what can be described, just as well, as an attempt to prevent major surprise: if I have a prior belief that all major carbon emitting countries will have halved their emissions by the end of 2049, and it turns out that neither of them has, then I am likely to be surprised—certainly more suprised than seeing my belief confirmed. Consequently, if I acquire information to the effect that these are the only two possible outcomes (i.e., either all countries cut emissions, or none of them does), then, on the assumption that this information stems from some noisy observation of the true state, I will use my prior belief and gravitate towards the outcome that occasions less surprise.

In this revision procedure, consistent with both the Dalal and the min-max operators, the measure of surprise is taken to depend only on the absolute difference between my prior belief and the states of affairs learned to be viable. However, we can readily imagine that the amount of anticipated surprise depends in equal measure on other factors, e.g., the context provided by the newly acquired information: if in 2049 it turns out that none of the countries has reduced emissions, then I am likely to be less surprised if I had been told in advance that at most one of them would than if I had been told that, possibly, any number of them could achieve the target. In other words, it is desirable to have a broader notion of surprise complementing the absolute one, to account for situations in which change in the epistemic state depends not just on the prior belief but also on the range of options provided by the new information. However, despite the fact that surprise minimization is a natural idea that has been gaining traction in Cognitive Science (Friston 2010; Hohwy 2016), there are not many belief revision policies that explicitly take it into account.

In this paper we put forward a notion of relative surprise that is richer in precisely this sense, and leverage it to define a new type of revision operator, called the *Hamming surprise min-max operator*, and which is calibrated to take into account contextual effects as described above. Though it deviates from some of the postulates in the AGM framework (notably, *Vacuity*, *Superexpansion* and *Subexpansion* (Fermé and Hansson 2018)), we show that the Hamming surprise min-max operator shares other desirable, though less obvious, features with the Dalal and the Hamming distance min-max operator. Significantly, we use these features to fully characterize the newly introduced surprise operator, in the process obtaining full chacterizations for the Dalal and Hamming distance min-max operators.

Contributions. On a conceptual level, we argue that the notion of distance standardly used to define revision operators can be seen as quantifying a measure of surprise, with different distance-based operators providing different ways to minimize it. We then enrich this landscape by introducing a notion of *relative* surprise, which is then put to use in defining the Hamming surprise min-max operator. We compare this operator against the standard KM postulates for revision (Katsuno and Mendelzon 1992) and present new postulates that complement the KM ones, for a full characterization. The versatility of the ideas underlying these postulates is showcased by adapting them to the Dalal and Hamming distance min-max operators: in the case of the min-max operator our postulates complement the subset of KM postulates the operator is known to satisfy; in the case of the Dalal operator our postulates strengthen the KM postulates. In both cases, we obtain full characterizations.

Related work. Among belief revision operators that are insensitive to syntax, the Dalal operator has received a significant amount of attention, either from attempts to express it by encoding the Hamming distance between truth-value assignments at the syntactic level (del Val 1993; Pozos-Parra, Liu, and Perrussel 2013); as an instance of the more general class of parameterized difference operators (Peppas and Williams 2018; Aravanis, Peppas, and Williams 2021); or in relation to Parikh's relevance-sensitivity axiom (Peppas et al. 2015). However, to the best of our knowledge, the characterization we offer here is the first of its kind.

Strengthening the AGM framework to induce additional desired behavior from revision operators has been considered in relation to issues of iterated revision (Darwiche and Pearl 1997), or relevance sensitivity (Parikh 1999; Peppas and Williams 2016). In terms of choice rules, the closest analogue to the surprise minimization operator is the decision rule that minimizes maximum regret in decisions with ignorance (Milnor 1954; Lave and March 1993; Peterson 2017), with Hamming distances playing the role of utilities in our present setting. However, the logical setting and the fact that the distances depend on the states themselves means that decision theoretic results do not translate easily to our current framework.

Outline. Section 2 introduces the main notions related to propositional logic and belief revision that will be used in the rest of the paper, and argues for the surprise-based interpretations of distances, Section 3 defines the relative Hamming surprise measure and the Hamming surprise min-max operator. Sections 4 and 5 consist of a slight detour in which the Dalal and Hamming distance min-max operators are characterized, setting up the stage for the characterization of the surprise operator in Section 6. Section 7 offers conclusions.

2 Preliminaries

Propositional Logic. We assume a finite set A of *propositional atoms*, large enough that we can always reach into it and find additional, unused atoms, if any are needed. The *set $\mathcal{L}$ of propositional formulas* is generated from the atoms in A using the usual propositional connectives ($\land$, $\lor$, $\neg$, $\rightarrow$ and $\leftrightarrow$), as well as the constants $\bot$ and $\top$.

An *interpretation* w is a function mapping every atom in A to either *true* or *false*. Since an interpretation w is completely determined by the set of atoms in A it makes true, we will identify w with this set of atoms and, if there is no danger of ambiguity, display w as a word where the letters are the atoms assigned to true. The *universe $\mathcal{U}$* is the set of all interpretations for formulas in $\mathcal{L}$. If w_1 and w_2 are interpretations, the *symmetric difference $w_1 \triangle w_2$ of w_1 and w_2* is defined as $w_1 \triangle w_2 = (w_1 \setminus w_2) \cup (w_2 \setminus w_1)$, i.e., as the set of atoms on which w_1 and w_2 differ. The *Hamming distance* $d_{\mathrm{H}} \colon \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{N}$ is defined, for any interpretations w_1 and w_2, as $d_{\mathrm{H}}(w_1, w_2) = |w_1 \triangle w_2|$. Intuitively, the Hamming distance $d_{\mathrm{H}}(w_1, w_2)$ between w_1 and w_2 counts the number of atoms that w_1 and w_2 differ on, and is used to quantify the disagreement between two interpretations.

The models of a propositional formula φ are the interpretations that satisfy it, and we write $[\varphi]$ for the set of models of φ. If φ_1 and φ_2 are propositional formulas, we say that φ_1 *entails* φ_2, written $\varphi_1 \models \varphi_2$, if $[\varphi_1] \subseteq [\varphi_2]$, and that they are *equivalent*, written $\varphi_1 \equiv \varphi_2$, if $[\varphi_1] = [\varphi_2]$. A propositional formula φ is *consistent* if $[\varphi] \neq \emptyset$. The models of $\bot$ and $\top$ are $[\bot] = \emptyset$ and $[\top] = \mathcal{U}$. We will occasionally find it useful to explicitly represent the models of a formula, in which case we write $\varphi_{v_1,\dots,v_n}$ for a propositional formula such that $[\varphi_{v_1,\dots,v_n}] = \{v_1, \dots, v_n\}$. A propositional formula φ is *complete* if it has exactly one model, and we will typically denote a complete formula as φ_v to draw attention to its unique model v. The *null formula ε* and the *full formula α* are defined as $\varepsilon = \bigwedge_{p \in A} \neg p$ and $\alpha = \bigwedge_{p \in A} p$, i.e., as the conjunction of the negated and non-negated atoms in A, respectively. Note that $[\varepsilon] = \{\emptyset\}$ and $[\alpha] = A$.

Distance-based belief revision. A *revision operator* $\circ$ is a function $\circ \colon \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$, taking as input two propositional formulas, denoted φ and μ, and standing for the agent's prior and newly acquired information, respectively, and returning a propositional formula, denoted $\varphi \circ \mu$. Two revision operators $\circ_1$ and $\circ_2$ are *equivalent*, written $\circ_1 \equiv \circ_2$, if $\varphi \circ_1 \mu \equiv \varphi \circ_2 \mu$, for any formulas φ and μ.

The primary device for generating concrete revision operators we make recourse to here is the Hamming distance. Thus, the *Hamming distance min-min operator* $\circ^{d_{\mathrm{H}}, \min}$, or, as it is more commonly known, *the Dalal operator* (Dalal 1988), is defined, for any propositional formulas φ and μ, as a formula $\varphi \circ^{d_{\mathrm{H}}, \min} \mu$ such that:

$$[\varphi \circ^{d_{\mathrm{H}}, \min} \mu] = \operatorname{argmin}_{w \in [\mu]} \min_{v \in [\varphi]} d_{\mathrm{H}}(v, w).$$

Intuitively, the shortest distance from w to any model of φ, i.e., $\min_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$, can be interpreted as a measure of distance between w and φ, and we will refer to it as the *Hamming min-distance between φ and μ*. The result $\varphi \circ^{d_{\mathrm{H}}, \min} \mu$

d_{H}	$\emptyset$	$abcd$	min	max
$\emptyset$	0	4	**0**	4
$abcd$	4	0	**0**	4
abe	3	3	3	**3**

Table 1: Hamming distances $d_{\mathrm{H}}(v, w)$ for $v \in [\varphi]$, $w \in [\mu]$, with $[\varphi] = \{\emptyset, abcd\}$ and $[\mu] = \{\emptyset, abcd, abe\}$. The lower $d_{\mathrm{H}}(v, w)$ is, the more plausible w is considered to be, from the standpoint of v. The minimal and maximal values per model of μ are tallied on the right, with the values preferred by operators $\circ^{d_{\mathrm{H}},\min}$ and $\circ^{d_{\mathrm{H}},\max}$, i.e., the minimal among the minimal and maximal values, respectively, in bold font.

of revision, then, selects those models of μ that are closest to φ according to this measure.

Recently, an alternative revision operator has been analyzed (Haret and Woltran 2019): what we will call here the *Hamming distance min-max operator* $\circ^{d_{\mathrm{H}},\max}$, defined, for any φ and μ, as a formula $\varphi \circ^{d_{\mathrm{H}},\max} \mu$ such that:

$$[\varphi \circ^{d_{\mathrm{H}},\max} \mu] = \mathrm{argmin}_{w \in [\mu]} \max_{v \in [\varphi]} d_{\mathrm{H}}(v, w),$$

i.e., a formula whose models are exactly those models of $[\mu]$ that minimize the Hamming distance to $\max_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$, the *Hamming max-distance between φ and μ*.

Distance as surprise. Consistent with the idea that revision models the agent learning about the world around it, we can see the new information μ as a noisy observation of some underlying ground truth state w^*: by acquiring μ, the agent learns of a set of outcomes (the models of μ), all of which stand a chance of being the true state w^*. In that sense, the distance $d(v, w)$ between any $v \in [\varphi]$ and $w \in [\mu]$ stands for a quantity that can be aptly described as *surprise*: it is the difference between what the agent expects is the case (v) and what might turn out to actually be the case (w). Naturally, the agent will want to minimize the divergence between its predictions and reality, with existing revision operators providing different means to do so.

Example 1. *Consider a set $A = \{a, b, c, d, e\}$ of atoms, standing for countries that might meet their emission targets before 2049, and formulas $\varphi = (\neg a \wedge \neg b \wedge \neg c \wedge \neg d \wedge \neg e) \vee (a \wedge b \wedge c \wedge d \wedge \neg e)$ and $\mu = \varphi \vee (a \wedge b \wedge \neg c \wedge \neg d \wedge e)$, with $[\varphi] = \{\emptyset, abcd\}$ and $[\mu] = \{\emptyset, abcd, abe\}$. Using the Hamming distances depicted in Table 1, we obtain that $[\varphi \circ^{d_{\mathrm{H}},\min} \mu] = \{\emptyset, abcd\}$ and $[\varphi \circ^{d_{\mathrm{H}},\max} \mu] = \{abe\}$.*

Intuitively, we read this as saying that if an agent believes the true state to be either of the worlds in $[\varphi] = \{\emptyset, abcd\}$, but finds out it is one among $[\mu] = \{\emptyset, abcd, abe\}$, then $\circ^{d_{\mathrm{H}},\min}$ selects the new belief to be $\{\emptyset, abcd\}$, as this supplies the least amount of surprise in an optimistic, best best-case scenario: if the true state turns out to be either of $\emptyset$ or abcd, then the agent, believing this, will be able to say "I told you so!"; the abe case, which is surprising in both cases, is ignored. In a complementary approach, the $\circ^{d_{\mathrm{H}},\max}$ operator shifts the agent's belief to $\{abe\}$, as this provides, more cautiously, the best worst-case scenario: from the standpoint of both $\emptyset$ or abcd, abe seems the least risky of the other options.

Example 1 serves as a springboard for some important observations. Firstly, it illustrates that $\circ^{d_{\mathrm{H}},\min}$ and $\circ^{d_{\mathrm{H}},\max}$ are distinct operators. Secondly, it is apparent from Example 1 that, given prior beliefs φ, interpretations can be ranked according to their Hamming min- or max-distance to φ. It is straightforward to see that (i) in both cases the resulting rankings depend only on the models of φ, are total and admit ties; (ii) the min-distance places models of φ at the bottom of this ranking, i.e., as the most plausible interpretations according to φ, in a pattern that goes under the name of a *faithful ranking* (Katsuno and Mendelzon 1992); and, perhaps, less conspicuously, that (iii) the max-distance places models of the so-called *dual of* φ (i.e., the formula obtained from φ by replacing all its atoms with their negations), at the very top, i.e., as the least plausible interpretations according to φ (Haret and Woltran 2019). The different flavors of rankings, faithful or otherwise, generated in this distance-based approach usually play a prominent role in representation results for revision, as they open up a level of abstraction between that of concrete numbers and general principles. In this work, however, we will bypass talk of rankings and work directly at the interface between distance-based measures and normative principles.

Finally, an observation that will prove useful is that we can (and will) think of the individual models v of φ as generating their own plausibility rankings over interpretations: these rankings correspond to the columns in Table 1 and are the rankings that would be generated if the prior belief were the complete formula $[\varphi_v] = \{v\}$, i.e., what the landscape of plausibility looks like if the agent puts the entire weight of its belief on φ_v. Revision can then be seen as employing a function ($\min$ or $\max$) to aggregate the individual rankings, and then choosing something out of the aggregated result: the Dalal operator $\circ^{d_{\mathrm{H}},\min}$ chooses, optimistically, the models that are the best of the best, while $\circ^{d_{\mathrm{H}},\max}$ chooses, pessimistically, the best of the worst models across the individual rankings. In keeping with this way of looking at things, we will often speak, loosely, of formulas and interpretations 'judging' and 'choosing' among possible outcomes.

What recommends the choice behavior of operators (such as Dalal's operator) as reasonable is adherence to a set of intuitive normative principles, or *rationality postulates*. The most common set of such principles consists of the AGM postulates for revision (Alchourrón, Gärdenfors, and Makinson 1985), which we present here in the Katsuno-Mendelzon formulation (Katsuno and Mendelzon 1992). The postulates apply for any propositional formulas φ, μ, μ_1 and μ_2:

(R_1) $\varphi \circ \mu \models \mu$.

(R_2) If $\varphi \wedge \mu$ is consistent, then $\varphi \circ \mu \equiv \varphi \wedge \mu$.

(R_3) If μ is consistent, then $\varphi \circ \mu$ is consistent.

(R_4) If $\varphi_1 \equiv \varphi_2$ and $\mu_1 \equiv \mu_2$, then $\varphi_1 \circ \mu_1 \equiv \varphi_2 \circ \mu_2$.

(R_5) $(\varphi \circ \mu_1) \wedge \mu_2 \models \varphi \circ (\mu_1 \wedge \mu_2)$.

(R_6) If $(\varphi \circ \mu_1) \wedge \mu_2$ is consistent, then $\varphi \circ (\mu_1 \wedge \mu_2) \models (\varphi \circ \mu_1) \wedge \mu_2$.

The primary assumption of revision (postulate R_1) is that new information originates with a trustworthy source; thus, revising φ by μ involves a commitment to accept the newly acquired information. Postulate R_2, known as the *Vacuity* postulate, says that if the newly acquired information μ does not contradict the prior information φ, the result is just the conjunction of μ and φ. Postulate R_3 says that if the newly acquired information μ is consistent, then the revision result should also be consistent. Postulate R_4 says that the result depends only on the semantic content of the information involved. Postulates R_5 and R_6, known as *Subexpansion* and *Superexpansion*, respectively, enforce a certain kind of coherence when the new information is presented sequentially, which is for present purposes best understood as akin to a form of *independence of irrelevant alternatives* familiar from rational choice (Sen 2017): the choice over two alternatives (here, interpretations w_1 and w_2 in $[\mu]$) should *not* depend on the presence of other alternatives in the menu (here represented by new information μ).

The Dalal operator $\circ^{d_{\mathrm{H}},\min}$ satisfies postulates R_1-R_6 (Katsuno and Mendelzon 1992), though these postulates do not uniquely characterize it. The Hamming distance max-operator $\circ^{d_{\mathrm{H}},\max}$ satisfies postulates R_1 and R_3-R_6 but not R_2, though it does satisfy the following two postulates (Haret and Woltran 2019), where $\overline{\varphi}$ stands for the *dual of* φ, as defined above:

(R_7) If $\varphi \circ \mu \models \overline{\varphi}$, then $\varphi \circ \mu \equiv \mu$.

(R_8) If $\mu \not\models \overline{\varphi}$, then $(\varphi \circ \mu) \land \overline{\varphi}$ is inconsistent.

In certain circumstances, $\overline{\varphi}$ can be thought of as the point of view opposite to that of φ, such that, taken together, postulates R_7 and R_8 inform the agent to believe states of affairs compatible with $\overline{\varphi}$ only if it has no other choice in the matter: the models of $\overline{\varphi}$ should be part of a viewpoint one is willing to accept only as a last resort.

3 Relative Hamming Surprise Minimization

In this section we introduce our novel surprise-based operator. We start by defining, for any interpretations v and w, the (relative) *Hamming surprise* $s^{\mu}_{\mathrm{H}}(v, w)$ *of* v *with respect to* w *relative to* μ, as:

$$s^{\mu}_{\mathrm{H}}(v, w) = d_{\mathrm{H}}(v, w) - d_{\mathrm{H}}(v, \mu),$$

i.e., the distance between v and w normalized by the distance between v and μ. The new information μ, here, serves as the reference point, or context, relative to which surprise is calculated. The *Hamming surprise min-max operator* $\circ^{s,\max}$ is defined as a formula $\varphi \circ^{s_{\mathrm{H}},\max} \mu$ such that:

$$[\varphi \circ^{s_{\mathrm{H}},\max} \mu] = \operatorname{argmin}_{w\in[\mu]} \max_{v\in[\varphi]} s^{\mu}_{\mathrm{H}}(v, w),$$

i.e., as a formula whose models are exactly those models of μ that minimize maximum Hamming surprise with respect to φ, and relative to μ. We refer to $\max_{v\in[\varphi]} s^{\mu}_{\mathrm{H}}(v, w)$, as the *max-surprise of* φ *with* w *relative* μ.

Example 2. *Consider formulas* φ *and* μ *as in Example 1, with* $[\varphi] = \{\emptyset, abcd\}$ *and* $[\mu] = \{\emptyset, abcd, abe\}$. *We have that* $d_{\mathrm{H}}(\emptyset, \mu) = \min_{w\in[\mu]} d_{\mathrm{H}}(\emptyset, w) = 0$, *and thus* $s^{\mu}_{\mathrm{H}}(\emptyset, abcd) = d_{\mathrm{H}}(\emptyset, abcd) - d_{\mathrm{H}}(\emptyset, \mu) = 4 - 0 = 4$. *The surprise terms are depicted in Table 2. We obtain, thus, that* $[\varphi \circ^{s_{\mathrm{H}},\max} \mu] = [\varphi \circ^{d_{\mathrm{H}},\max} \mu] = \{abe\}$. *Consider, now, a formula* ν *with* $[\nu] = \{abcd, abe\}$, *with the surprise scores depicted in Table 3. Note that in this case we obtain that* $[\varphi \circ^{s_{\mathrm{H}},\max} \nu] = \{abcd\}$. *Thus, in revision by* μ, *abe is chosen over abcd, whereas in revision by* ν *the choice is reversed. Intuitively, when* $\emptyset$ *stops being a viable option, abcd becomes more attractive than abe, as the amount of surprise it would inflict, from the standpoint of* $\emptyset$, *relative to abe, becomes smaller: considering the options, abcd is not as extreme as abe. In other words, for* $\emptyset$ *the two interpretations abcd and abe are sufficiently alike to be considered almost equally risky: the marginal surprise that abcd carries over abe is not big enough to be considered significant, so that the final decision ends up choosing abcd as carrying the least amount of risk. By contrast, when* $\emptyset$ *is present as an option (see Table 2) the situation is markedly different, as the relative surprise of actually ending up with abcd or abe becomes much more significant.*

s^{μ}_{H}	$\emptyset$	*abcd*	max
$\emptyset$	$0-0$	$4-0$	4
abcd	$4-0$	$0-0$	4
abe	$3-0$	$3-0$	**3**

Table 2: Relative Hamming surprise $s^{\mu}_{\mathrm{H}}(v, w)$ for $v \in [\varphi]$, $w \in [\mu]$, for $[\varphi] = \{\emptyset, abcd\}$, $[\mu] = \{\emptyset, abcd, abe\}$, and relative to μ: $d_{\mathrm{H}}(v, w)$ is normalized by the distance $d_{\mathrm{H}}(v, \mu)$ from v to μ. The lower surprise is, the more plausible w is considered to be, from the standpoint of v. The model minimizing overall surprise is emphasized in bold font.

s^{ν}_{H}	$\emptyset$	*abcd*	max
abcd	$4-3$	$0-0$	**1**
abe	$3-3$	$3-0$	3

Table 3: Relative Hamming surprise $s^{\nu}_{\mathrm{H}}(v, w)$, $[\varphi] = \{\emptyset, abcd\}$, $[\mu] = \{\emptyset, abcd, abe\}$. The best interpretation is now $abcd$: the ranking induced by relative surprise depends on μ, as well as φ.

The type of scenario depicted in Example 2 is reminiscent of deviations from the principle of independence from irrelevant alternatives signaled in the rational choice literature (Sen 1993), and immediately points toward a salient feature of the relative surprise operator we have introduced: it is not guaranteed to satisfy postulates R_2, R_5 and R_6. Indeed, for φ and μ from Example 2 we have that $[\varphi \circ^{s_{\mathrm{H}},\max} \mu] = \{abe\}$, despite the fact that $[\varphi \land \mu] = \{\emptyset, abcd\}$, which speaks to postulate R_2. Since $\varphi \circ^{s_{\mathrm{H}},\max} \mu$ coincides, in this case, with $\varphi \circ^{d_{\mathrm{H}},\max} \mu$, and $\circ^{d_{\mathrm{H}},\max}$ is already known not to satisfy postulate R_2, this is perhaps not surprising, but similar reasoning shows that $\varphi \circ^{s_{\mathrm{H}},\max} \mu$ does not satisfy postulates R_7 and R_8 either. And $[\varphi \circ^{s_{\mathrm{H}},\max} (\mu \land \nu)] = \{abcd\}$, despite the fact that $[(\varphi \circ^{s_{\mathrm{H}},\max} \mu) \land \nu] = \{abe\}$, which speaks to postulates R_5 and R_6. More to the point, the ranking on interpretations that is generated by the surprise measure $s + \mathrm{H}$ varies with μ, to the extent that narrowing down the new information, as in Example 2, can lead to inversions between the relative ranking of two interpretations. At the same time, the ranking plainly depends on nothing more than φ and μ, such that the result of revision is invariant to the syntax of the prior and new information. Additionally, $\circ^{s_{\mathrm{H}},\max}$ selects the result from the models of μ, and is guaranteed to output *something* as long as μ is consistent. We summarize these observations in the following proposition.

Proposition 1. *The operator* $\circ^{s_{\mathrm{H}},\max}$ *satisfies postulates* R_1, R_3 *and* R_4, *but not* R_2, R_5, R_6, R_7 *and* R_8.

One detail worth mentioning is that when φ is complete all operators presented so far coincide.

Proposition 2. *For any complete formula* φ_v, $\varphi \circ^{d_{\mathrm{H}},\min} \mu \equiv \varphi \circ^{d_{\mathrm{H}},\max} \equiv \varphi \circ^{s_{\mathrm{H}},\max} \mu$, *for any formula* μ.

Proof. For complete φ_v it is only the relative ranking of interpretations with respect to v that matters, and this is the same for all three operators. □

Proposition 1 shows that the $\circ^{s_{\mathrm{H}},\max}$ operator does not fit neatly into the standard revision framework. However, since, we have argued, $\circ^{s_{\mathrm{H}},\max}$ formalizes an appealing intuition, it will be useful to unearth the general rules underpinning it: our goal, now, is to find a set of normative principles strong enough to characterize $\circ^{s_{\mathrm{H}},\max}$. A set of such principles is offered in Section 6, but, since $\circ^{s_{\mathrm{H}},\max}$ can be seen as a more involved min-max operator, we set the scene by first characterizing $\circ^{d_{\mathrm{H}},\max}$. And to set the scene for $\circ^{d_{\mathrm{H}},\max}$, we first characterize the Dalal operator.

4 Characterizing the Dalal Operator

In this section we present a set of postulates that characterize the Dalal operator $\circ^{d_{\mathrm{H}},\min}$. Apart from being of independent interest, this section presents, in the familiar setting of a known operator, the main intuitions and techniques used in subsequent sections. We start by introducing some additional new notions.

A *renaming* r *of* A is a bijective function $r\colon A \to A$. If φ is a propositional formula, the *renaming* $r(\varphi)$ *of* φ is a formula $r(\varphi)$ whose atoms are replaced according to r. On the semantic side, if w is an interpretation and r is a renaming of A, the *renaming* $r(w)$ *of* w is an interpretation obtained by replacing every atom p in w with $r(p)$. If $\mathcal{W}$ is a set of interpretations, the *renaming* $r(\mathcal{W})$ *of* $\mathcal{W}$ is defined as $r(\mathcal{W}) = \{r(w) \mid w \in \mathcal{W}\}$, i.e., the set of interpretations whose elements are the renamed interpretations in $\mathcal{W}$.

A *flip function* $f\colon 2^A \times \mathcal{L} \to \mathcal{L}$ is a function that takes as input a set $v \subseteq A$ of atoms (equivalently, v can be thought of as an interpretation) and a propositional formula φ, and returns a propositional formula $f_v(\varphi)$ that is just like φ except that all the atoms from v that appear in φ are flipped, i.e., replaced with their negations. Overloading notation, a flip function applied to interpretations v and w returns an interpretation $f_v(w)$ in which all the atoms from v that appear in w are flipped, i.e., $f_v(w) = \{p \in A \mid p \in w \text{ and } p \notin v, \text{ or } p \in v \text{ and } p \notin w\}$. It is straightforward to see that $f_v(w) = w \triangle v$. If $\mathcal{W}$ is a set of interpretations, then $f_v(\mathcal{W}) = \{f_v(w) \mid w \in \mathcal{W}\}$, i.e., the set of interpretations obtained by flipping every atom in v.

Example 3. *For the set* $A = \{a, b, c\}$ *of atoms, consider a formula* $\varphi = a \wedge \neg c$, *with* $[\varphi] = \{a, ab\}$, *and a renaming* r *such that* $r(a) = b$, $r(b) = c$ *and* $r(c) = a$. *We obtain that* $r(\varphi) = r(a) \wedge \neg r(c) = b \wedge \neg a$, *with* $[r(\varphi)] = \{b, bc\} = \{r(a), r(ab)\}$. *Flipping atoms* b *and* c, *we have that* $f_{bc}(\varphi) = a \wedge \neg(\neg c)$, *with* $[f_{bc}(\varphi)] = \{abc, ac\}$. *Note that* $[f_{bc}(\varphi)] = \{f_{bc}(a), f_{bc}(ab)\} = \{a \triangle bc, ab \triangle bc\}$.

In Example 3 it holds that: (*i*) $[r(\varphi)] = r([\varphi])$, (*ii*) $[f_w(\varphi)] = f_w([\varphi])$ and (*iii*) $[f_w(\varphi)] = \{v \triangle w \mid v \in [\varphi]\}$, and we note here that all these equalities hold generally (for (*ii*) see, for instance, Exercise 2.28 in (Goldrei 2005)). Their relevance will become apparent shortly.

To characterize the Dalal operator $\circ^{d_{\mathrm{H}},\min}$ we introduce a set of new postulates, starting with *Neutrality* $\mathrm{R}_{\mathtt{N}}$:

($\mathrm{R}_{\mathtt{N}}$) If φ is complete, then $r(\varphi \circ \mu) \equiv r(\varphi) \circ r(\mu)$.

Postulate $\mathrm{R}_{\mathtt{N}}$ states that revision is invariant under renaming atoms and hence neutral in that the specific labels for the atoms do not matter towards the final result. This postulate is inspired by similar ideas in social choice and has appeared before in belief change contexts (Herzig and Rifi 1999; Marquis and Schwind 2014; Haret and Woltran 2019).

The next postulate concerns the effect of flipping the same atoms in both φ and μ, and is called, appropriately, the *Flipping* postulate $\mathrm{R}_{\mathtt{F}}$:

($\mathrm{R}_{\mathtt{F}}$) If φ is complete, then $f_v(\varphi \circ \mu) = f_v(\varphi) \circ f_v(\mu)$.

An additional constraint, the *Addition* postulate $\mathrm{R}_{\mathtt{A}}$, is obtained by considering the effect of adding new atoms that affect the standing of one interpretation, and is meant to apply to any formulas φ and μ and set x of new atoms, i.e., such that none of the atoms in x appears in either φ or μ:

($\mathrm{R}_{\mathtt{A}}$) If φ is complete and $(\varphi \circ \mu_{w_1,w_2}) \wedge \mu_{w_1}$ is consistent, then $\varphi \circ \mu_{w_1,w_2 \cup x} \equiv \mu_{w_1}$.

Postulate $\mathrm{R}_{\mathtt{A}}$ is best understood through a choice perspective: if w_1 is chosen by φ over w_2 when the choice is $[\mu_{w_1,w_2}] = \{w_1, w_2\}$, then adding extra new atoms x to w_2, (and, thereby, increasing the distance to φ) ensures that $w_2 \cup x$ is not chosen when the choice is $[\mu_{w_1,w_2 \cup x}] = \{w_1, w_2 \cup x\}$. In all of these postulates the prior belief φ is assumed to be complete: this is not essential for the characterization of the Dalal operator, but makes life easier in the characterization of the surprise minimization operator, in Section 6.

The next postulate involves a mix of flips and we ease into it by introducing an intermediary notion. The *best-of-best formula* $\beta_{\varphi,\mu}$ *with respect to* φ *and* μ is defined as:

$$\beta_{\varphi,\mu} = \varepsilon \circ \Big(\bigvee_{v \in [\varphi]} f_v(\mu) \Big),$$

i.e., as the result of revising the null formula ε (recall that $[\varepsilon] = \{\emptyset\}$) by a disjunction made up of multiple versions of μ, where each such version is obtained by flipping the atoms in a model v of φ. Intuitively, the intention is to recreate the table of Hamming distances (e.g., Table 1) without using numbers: recall that $[f_v(\mu)] = f_v([\mu])$ and $f_v(w) = w \triangle v$ and thus, semantically, we have that $[\bigvee_{v \in [\varphi]} f_v(\mu)] =$

$\{w_i \triangle v_j \mid w_i \in [\mu], v_j \in [\varphi]\}$. In other words, we are creating a scenario in which ε has to choose between interpretations obtained as the symmetric difference of the elements of $[\varphi]$ and $[\mu]$. The result we are working towards, yet to be proven, is that an element of $[\bigvee_{v \in [\varphi]} f_v(\mu)]$ chosen by ε, i.e., an interpretation $w_i \triangle v_j \in [\beta_{\varphi,\mu}]$, corresponds to an interpretation $w_i \in [\mu]$ that minimizes the overall Hamming distance to φ, and is thus among the best of the best interpretations in this revision scenario. The role of the *Best-of-Best* postulate $\mathrm{R}_{\mathtt{BOB}}$, then, is to recover the models of μ from the models of $\beta_{\varphi,\mu}$:

$$(\mathrm{R}_{\mathtt{BOB}}) \;\; \varphi \circ \mu \equiv \left(\bigvee_{v \in [\varphi]} f_v(\beta_{\varphi,\mu}) \right) \wedge \mu.$$

Postulate $\mathrm{R}_{\mathtt{BOB}}$ stipulates that the result of revising φ by μ consists of those interpretations of μ that come out of flipping $\beta_{\varphi,\mu}$ by each model of φ, in this way reversing the initial flips that delivered the revision formula posed to ε.

What is the significance of the null formula ε in $\beta_{\varphi,\mu}$? We want to reduce arbitrary revision tasks to a common denominator, a base case in which the result of revision can be decided without explicit appeal to distances (i.e., numbers), and only by appeal to desirable normative principles, such as the postulates laid out above. The case when the prior belief is ε turns out to be well suited for this task, since, as we show next, postulates R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$ and $\mathrm{R}_{\mathtt{A}}$ guarantee that ε always selects the interpretations with minimal cardinality.

Lemma 1. *If a revision operator $\circ$ satisfies postulates* R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$ *and* $\mathrm{R}_{\mathtt{A}}$, *then, for any formula μ, it holds that* $[\varepsilon \circ \mu] = \mathrm{argmin}_{w \in [\mu]} |w|$.

Proof. ("$\subseteq$") Suppose, first, that $w_1 \in [\varepsilon \circ \mu]$ and there is $w_2 \in [\mu]$ such that $|w_1| > |w_2|$. Using postulate R_5 we obtain that $w_1 \in [\varphi \circ \mu_{w_1,w_2}]$. We now show that this leads to a contradiction, and we do this using the Neutrality postulate $\mathrm{R}_{\mathtt{N}}$: however, we would like to apply $\mathrm{R}_{\mathtt{N}}$ to interpretations of equal size. Towards this, take a set x of new atoms (i.e., that do not occur in either φ or μ), with $|x| = |w_1| - |w_2|$, and add x to w_2 to form $w'_2 = w_2 \cup x$. We have that $|w'_2| = |w_2| + (|w_1| - |w_2|) = |w_1|$, i.e., w_1 and w'_2 are of the same size, which implies that $|w_1 \setminus w'_2| = |w'_2 \setminus w_1|$. Applying the addition postulate $\mathrm{R}_{\mathtt{A}}$, we obtain that $w'_2 \notin [\varepsilon \circ \mu_{w_1,w'2}]$.

Consider, now, a renaming r that swaps atoms in $w_1 \setminus w'_2$ with atoms in $w'_2 \setminus w_1$, made possible by the fact that $w_1 \setminus w'_2$ and $w'_2 \setminus w_1$ are of the same size. This implies that $r(w_1) = w'_2$ and $r(w'_2) = w_1$ and thus $r([\mu_{w_1,w'_2}]) = r(\{w_1, w'_2\}) = \{r(w_1), r(w'_2)\} = \{w'_2, w_1\} = [\mu_{w_1,w'_2}]$. Applying the Neutrality postulate $\mathrm{R}_{\mathtt{N}}$ to $\varepsilon \circ \mu_{w_1,w'_2}$ with the renaming r thus defined, and, keeping in mind that $[r(\varepsilon)] = [\varepsilon]$, and thus that $r(\varepsilon) \equiv \varepsilon$, we obtain that:

$$\begin{aligned}
\{w_1\} &= [\varepsilon \circ \mu_{w_1,w'_2}] && \text{by assumption and } \mathtt{A} \\
&= [r(\varepsilon) \circ r(\mu_{w_1,w'_2})] && \text{by def. of } r \text{ and } \mathrm{R}_4 \\
&= [r(\varepsilon \circ \mu_{w_1,w'_2}))] && \text{by } \mathtt{N} \\
&= r([\varepsilon \circ \mu_{w_1,w'_2}]) && \text{property of } r \\
&= r(\{w_1\}) && \text{by assumption} \\
&= \{w'_2\}.
\end{aligned}$$

This implies that $w_1 = w'_2$ but, since w'_2 contains a non-negative number of atoms that do not appear in w_1, this is a contradiction.

("$\supseteq$") For the opposite direction, suppose that $w_1 \in \mathrm{argmin}_{w \in [\mu]} |w|$ but $w_1 \notin [\varepsilon \circ \mu]$. Using postulates R_1 and R_3 we have that there is $w_2 \in [\varphi \circ \mu]$ and, with postulate R_6 we obtain that $[\varepsilon \circ \mu_{w_1,w_2}] = \{w_2\}$. Since $|w_1| \leq |w_2|$ we add to w_1 a set x of new atoms, where $|x| = |w_2| - |w_1|$, and denote $w'_1 = w_1 \cup x$. Applying $\mathrm{R}_{\mathtt{A}}$ we obtain that $[\varepsilon \circ \mu_{w'_1,w_2}] = \{w_2\}$ and, using a renaming r defined, as in the previous direction, such that $r(w_2) = w'_1$ and $r(w'_1) = w_2$, and applying $\mathrm{R}_{\mathtt{N}}$ to r and $\varepsilon \circ \mu_{w'_1,w_2}$, we obtain that $[\varepsilon \circ \mu_{w'_1,w_2}] = \{w'_1\}$, leading to a contradiction. □

Lemma 1 shows that, in the very particular case in which the prior belief is ε, we can ensure that the result of revision coincides with the result delivered by the Dalal operator. The next move consists in using the Flipping postulate $\mathrm{R}_{\mathtt{F}}$ to extend this fact to complete formulas.

Lemma 2. *If a revision operator $\circ$ satisfies postulates* R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$ *and* $\mathrm{R}_{\mathtt{F}}$, *then, for any formula μ and complete formula φ_v, it holds that* $[\varphi_v \circ \mu] = \mathrm{argmin}_{w \in [\mu]} d_{\mathrm{H}}(v, w)$.

Proof. By postulate $\mathrm{R}_{\mathtt{F}}$ it holds that $f_v(\varphi_v \circ \mu) \equiv f_v(\varphi_v) \circ f_v(\mu)$. Note, now, that $[f_v(\varphi_v)] = \{v \triangle v\} = \{\emptyset\}$, and thus $f_v(\varphi_v) \equiv \varepsilon$, while $[f_v(\mu)] = \{w \triangle v \mid w \in [\mu]\}$. By Lemma 1, it holds that $[\varepsilon \circ f_v(\mu)] = \min_{w \triangle v \in [f_v(\mu)]} |w \triangle v|$ and, since $d_{\mathrm{H}}(v, w) = |w \triangle v|$, we derive the conclusion. □

Lemma 2 shows that it is not just the formula ε that makes choices consistent with the Dalal operator, but any complete formula φ_v. The intuition driving Lemma 2 is that the situation where v chooses between w_1 and w_2 is equivalent, through the Flipping postulate $\mathrm{R}_{\mathtt{F}}$, to a scenario where $\emptyset$ chooses between $w_1 \triangle v$ and $w_2 \triangle v$: and we know that in this situation postulates $\mathrm{R}_{\mathtt{N}}$ and $\mathrm{R}_{\mathtt{A}}$ guide $\emptyset$ to choose the interpretation $w_i \triangle v$ of minimal cardinality, which corresponds to w_i being at minimal Hamming distance to v.

The next step involves pushing this intuition even further, to the case of any propositional formula φ. As anticipated, the Best-of-Best postulate $\mathrm{R}_{\mathtt{BOB}}$ is the postulate that facilitates this move, and the proof goes through the intermediary observation that the best-of-best formula $\beta_{\varphi,\mu}$ selects interpretations corresponding to the desired redult.

Lemma 3. *If $\circ$ is a revision operator that satisfies postulates* R_1, R_3-R_6 R_4, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$ *and* $\mathrm{R}_{\mathtt{F}}$ *then, for any formulas φ and μ and interpretations w and v, it holds that $w \triangle v \in [\beta_{\varphi,\mu}]$ if and only if $w \in \mathrm{argmin}_{w \in [\mu]} \min_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$.*

Proof. By Lemma 1, $[\beta_{\varphi,\mu}]$ chooses exactly those interpretations $w_i \triangle v_j$, for $w_i \in [\mu]$ and $v_j \in [\varphi]$, that are of minimal cardinality. Since $|w_i \triangle v_j| = d_{\mathrm{H}}(w_i, v_j)$, the conclusion follows immediately. □

By Lemma 3, the result of the Dalal operator $\circ^{d_{\mathrm{H}}, \min}$ applied to φ and μ consists of those interpretations $w \in [\mu]$ such that $w \triangle v \in [\beta_{\varphi,\mu}]$, for some $v \in [\varphi]$. The Best-of-Best postulate $\mathrm{R}_{\mathtt{BOB}}$ instructs us that these are exactly the

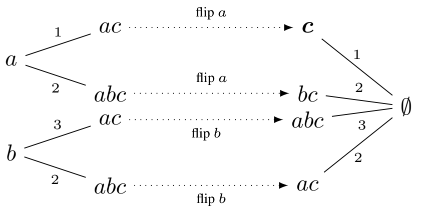


Figure 1: By flipping the atoms of $v \in [\varphi]$ in a model w of μ we get an interpretation $w \triangle v$ whose size corresponds to the Hamming distance between v and w, i.e., $|v \triangle w| = d_{\mathrm{H}}(v, w) = d_{\mathrm{H}}(\emptyset, v \triangle w) = d_{\mathrm{H}}(\emptyset, f_v(w))$. In this way, flipped models that get chosen by ε corresponds to models of μ that minimize overall Hamming distance to φ.

models of μ that should be chosen by an operator $\circ$, and provides the final piece in the sought after characterization.

Theorem 1. *A revision operator* $\circ$ *satisfies postulates* R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ *and* $\mathrm{R}_{\mathtt{BOB}}$ *if and only if* $\circ \equiv \circ^{d_{\mathrm{H}},\min}$.

Proof. For one direction, we take as known that the Dalal operator $\circ^{d_{\mathrm{H}},\min}$ satisfies postulates R_1, R_{3-6} (Katsuno and Mendelzon 1992) and $\mathrm{R}_{\mathtt{N}}$ (Haret and Woltran 2019). For postulate $\mathrm{R}_{\mathtt{N}}$, given Lemma 3, satisfaction of postulates $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ and $\mathrm{R}_{\mathtt{BOB}}$ follows straightforwardly.

For the other direction, we have to show that if $\circ$ satisfies all the stated postulates, then $[\varphi \circ \mu] = \mathrm{argmin}_{w \in [\mu]} \min_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$, for any formulas φ and μ. Lemma 3 already gives us that $\beta_{\varphi,\mu}$ selects those interpretations $w_i \triangle v_j$ for which $d_{\mathrm{H}}(w_i, v_j)$ is minimal among the set $\{w \triangle v \mid w \in [\mu], v \in [\varphi]\}$ of symmetric differences between models of φ and of μ. This means that if $w_i \in \mathrm{argmin}_{w \in [\mu]} \min_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$, then $w_i \triangle v_j \in [\beta_{\varphi,\mu}]$, for some $v_j \in [\varphi]$, and hence $(w_i \triangle v_j) \triangle v_j = w_i \in [f_{v_j}(\beta_{\varphi,\mu})]$, i.e., if w_i is selected by the Dalal operator then it shows up in $[(\bigvee_{v \in [\varphi]} f_v(\beta_{\varphi,\mu})) \wedge \mu]$. Conversely, suppose there is an interpretation $w_i \in [(\bigvee_{v \in [\varphi]} f_v(\beta_{\varphi,\mu})) \wedge \mu]$ that is not at minimal distance to φ. This means that $w_i = (w_j \triangle v_k) \triangle v_l$, where $w_j \in [\mu]$ corresponds to a model of μ that is at minimal Hamming distance to φ and $v_k, v_l \in [\varphi]$. We infer from this that $w_i \triangle v_l = ((w_j \triangle v_k) \triangle v_l) \triangle v_l = w_j \triangle v_k$, and thus $|w_i \triangle v_l| = |w_j \triangle v_k|$. But this contradicts the assumed minimality of $w_j \triangle v_k$. □

Note that postulate R_2 is not present in Theorem 1, even though the Dalal operator satisfies it, as it follows from the other postulates.

Theorem 1 can be read not just as a characterization of the Dalal operator, but also as a recipe, or a step-by-step argument, for constructing $\varphi \circ \mu$ from a set of simpler problems, in a srquence of steps guided by the transformations inherent in postulates $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ and $\mathrm{R}_{\mathtt{BOB}}$. The form such an argument could take is illustrated in the following example.

Example 4. *Consider formulas* $[\varphi] = \{a, b\}$ *and* $[\mu] = \{ac, abc\}$ *and note, first, that* $[\varphi \circ^{d_{\mathrm{H}},\min} \mu] = \{ac\}$*, as ac minimizes overall distance to* φ via $d_{\mathrm{H}}(a, ac) = 1$. *Assume, however, that we are given a revision operator* $\circ$ *that is not defined using distances, but is presented only as satisfying postulates* R_1, R_{3-6}, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ *and* $\mathrm{R}_{\mathtt{BOB}}$. *An agent revising according to* $\circ$ *can use the postulates to work its way toward* $[\varphi \circ^{d_{\mathrm{H}},\min} \mu]$ *without knowing anything about distances. This can be done by, first, splitting the problem into two revision problems, one for each model of* φ: $\varphi_a \circ \mu$ *and* $\varphi_b \circ \mu$, *where* $[\varphi_a] = \{a\}$ *and* $[\varphi_b] = \{b\}$. *The next step consists in reducing both problems to the common denominator of revising with prior belief* ε, *where* $[\varepsilon] = \{\emptyset\}$. *This is done by flipping a and b, respectively, in the two problems, to obtain the revision scenarios* $\varepsilon \circ f_a(\mu)$ *and* $\varepsilon \circ f_b(\mu)$, *with* $[f_a(\mu)] = \{f_a(ac), f_a(abc)\} = \{ac \triangle a, abc \triangle a\} = \{c, bc\}$ *and, likewise,* $[f_b(\mu)] = \{abc, ac\}$ *(see Figure 1). This move preserves Hamming distances in a crucial way: to take one instance,* $d_{\mathrm{H}}(a, ac) = 1$, *where* $a \in [\varphi]$ *and* $ac \in [\mu]$, *coincides with the Hamming distance between* $\emptyset$ *and* $f_a(ac) = c$, *and this distance coincides with the number of atoms in* $f_a(ac) = c$. *The operator* $\circ$, *of course, knows nothing of this: it performs these transformations solely because postulate* $\mathrm{R}_{\mathtt{BOB}}$ *warrants them. Thus, in the next step* ε *chooses among the models obtained from the successive flips of* μ, *i.e., it solves the revision problem* $\varepsilon \circ (f_a(\mu) \vee f_b(\mu))$. *Postulates* R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$ *and* $\mathrm{R}_{\mathtt{A}}$, via *the argument in Lemma 1, dictate that* ε *chooses the interpretation of minimal cardinality, such that* $[\beta_{\varphi,\mu}] = [\varepsilon \circ (f_a(\mu) \vee f_b(\mu))] = \{c\}$. *The result obtained, i.e., interpretation c, is the result of flipping the atom a in the interpretation* $ac \in [\mu]$*: to recover ac from c, we 'reverse' the original flips: one flip by a and one by b, to get* $[f_a(\beta_{\varphi,\mu}) \vee f_b(\beta_{\varphi,\mu})] = \{ac, bc\}$. *By postulate* $\mathrm{R}_{\mathtt{BOB}}$, *we have that* $[\varphi \circ \mu] = [f_a(\beta_{\varphi,\mu}) \vee f_b(\beta_{\varphi,\mu}) \wedge \mu] = \{ac\}$, *i.e., exactly the result produced by the Dalal operator* $\circ^{d_{\mathrm{H}},\min}$.

5 Characterizing the Hamming Distance Min-Max Operator

The postulates put forward in Section 4 for characterizing the Dalal operator prove their worth in an additional sense, as they can be put to use, with minimal modifications, in characterizing the Hamming distance min-max operator $\circ^{d_{\mathrm{H}},\max}$. This is the topic of the current section.

Of the newly proposed postulates, the Neutrality, Addition and Flipping postulates ($\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$ and $\mathrm{R}_{\mathtt{F}}$, respectively) can be used as stated in Section 4, while the Best-of-Best postulate $\mathrm{R}_{\mathtt{BOB}}$ has to be modified. Intuitively, this makes sense: postulates $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$ and $\mathrm{R}_{\mathtt{F}}$ are used in regulating what happens when the prior information is a complete formula φ_v (alternatively, for what happens in the ranking that corresponds to the v-column in the table of distances, e.g., Table 1), in which case, as per Proposition 2, all operators presented here coincide, whereas postulate $\mathrm{R}_{\mathtt{BOB}}$ instructs us how to choose when the prior information consists of more than one model (alternatively, across different columns of the table of distances). Correspondingly, postulate $\mathrm{R}_{\mathtt{BOB}}$ encodes the constraint that revision should pick the best of the best models across all of the φ_v's, for $v \in [\varphi]$, but this is not the rule that defines operator $\circ^{d_{\mathrm{H}},\max}$. For $\circ^{d_{\mathrm{H}},\max}$ we need a principle that mandates picking the best of the worst models across the φ_v's. The key fact allowing us to do this relies on a certain duality specific to the Hamming distance that will guide us in designing an appropriate postulate for $\circ^{d_{\mathrm{H}},\max}$, and which is summarized in the following result. Recall that A is the set of all atoms.

Lemma 4. *If v and w are interpretations and $|A| = n$, then $d_{\mathrm{H}}(v, w) = n - d_{\mathrm{H}}(A \setminus v, w)$.*

Intuitively, Lemma 4 implies that the further away w is from v (in terms of Hamming distance), the closer w is to $A \setminus v$. In particular, we can infer that:

$$\begin{aligned} d_{\mathrm{H}}(v, w) &= d_{\mathrm{H}}(\emptyset, |v \triangle w|) \\ &= d_{\mathrm{H}}(\emptyset, f_v(w)) \\ &= n - d_{\mathrm{H}}(A, f_v(w)). \qquad (1) \end{aligned}$$

Hence, $w \in [\mu]$ is among the models of μ at maximal Hamming distance to v if and only if $f_v(w)$ is, among the models of $f_v(\mu)$, the closest to A, or, more intuitively, the worst model of μ according to v is the best model of $f_v(\mu)$ according to α, where $[\alpha] = A$. We can thus define the *best-of-worst formula $\gamma_{\varphi,\mu}$ with respect to φ and μ* as:

$$\gamma_{\varphi,\mu} = \varepsilon \circ \Bigg(\bigvee_{v \in [\varphi]} \Big(\alpha \circ f_v(\mu) \Big) \Bigg),$$

i.e., as the result of revising the null formula ε by a disjunction made up of the results obtained from a sequence of revisions of the full formula α. In this sequence α is revised, in turn, by $f_v(\mu)$, for every model $v \in [\varphi]$.

Thus, similarly as for $\beta_{\varphi,\mu}$ from Section 4, $\gamma_{\varphi,\mu}$ simulates the process of going through the table of Hamming distances (e.g., Table 1), except that in this case we are interested in (i) selecting the worst elements according to each φ_v, for $v \in [\varphi]$, an operation reflected by the revision $\alpha \circ f_v(\mu)$, and (ii) selecting the best among these worst elements, an operation reflected by submitting the results obtained previously to ε for an additional round of revision. A bespoke postulate, called the *Best-of-Worst* postulate $\mathrm{R}_{\mathtt{BOW}}$, recovers the models of μ from the models of $\gamma_{\varphi,\mu}$:

($\mathrm{R}_{\mathtt{BOW}}$) $\varphi \circ \mu \equiv \left(\bigvee_{v \in [\varphi]} f_v(\gamma_{\varphi,\mu}) \right) \wedge \mu.$

Postulate $\mathrm{R}_{\mathtt{BOW}}$ stipulates that the result of revising φ by μ consists of those models of μ that come out of flipping $\gamma_{\varphi,\mu}$ by each model of φ, in this way reversing the initial flips that delivered the revision formula posed to ε.

The proof that the postulates put forward actually characterize the $\circ^{d_{\mathrm{H}},\max}$ operator hinges on $\gamma_{\varphi,\mu}$ selecting interpretations corresponding to models w of μ that minimize maximal Hamming distance to φ.

Lemma 5. *If $\circ$ is a revision operator that satisfies postulates R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ and $\mathrm{R}_{\mathtt{BOW}}$, then, for any formulas φ and μ and interpretations w and v, it holds that $w \triangle v \in [\gamma_{\varphi,\mu}]$ if and only if $w \in \operatorname{argmin}_{w \in [\mu]} \max_{v \in [\varphi]} d_{\mathrm{H}}(v, w)$.*

Proof. Using postulates $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$ and $\mathrm{R}_{\mathtt{F}}$ we can prove that α selects the models of μ that minimize Hamming distance to A, in a way completely analogous to Lemmas 1 and Lemma 2. Thus, using Equality 1, $\alpha \circ f_v(\mu)$ selects interpretations $w \triangle v$ such that $d_{\mathrm{H}}(v, w) = \max_{w' \in [\mu]} d_{\mathrm{H}}(v, w')$. Then, using Lemma 1, we obtain that $\gamma_{\varphi,\mu}$ selects interpretations $w \triangle v$ where w minimizes max-distance to φ. □

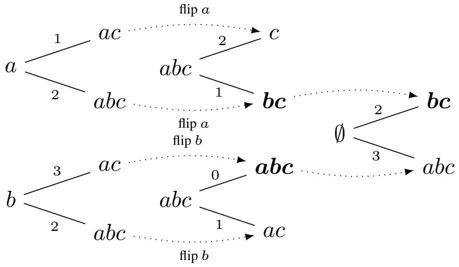


Figure 2: To get the best of the worst models of μ according to a and b we got through two rounds of revision: first, flip μ by a and by b. The results of $\alpha \circ f_a(\mu)$ and $\beta_{\circ} f_b(\mu)$ correspond to the models of μ at maximal distance to a and b, respectively. This result is further refined by passing it to ε for revision.

With Lemma 5 the characterization of $\circ^{d_{\mathrm{H}},\max}$ follows immediately.

Theorem 2. *If $\circ$ is a revision operator, then $\circ$ satisfies postulates R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ and $\mathrm{R}_{\mathtt{BOW}}$ iff $\circ \equiv \circ^{d_{\mathrm{H}},\max}$.*

The proof is similar, in its essentials, to the proof of Theorem 1 and is therefore omitted. The following example, however, illustrates how the mechanism works on a concrete case.

Example 5. *Consider formulas $[\varphi] = \{a, b\}$ and $[\mu] = \{ac, abc\}$, as in Example 4, over the set $A = \{a, b, c\}$ of atoms. Using the $\circ^{d_{\mathrm{H}},\max}$ operator we obtain that $[\varphi \circ^{d_{\mathrm{H}},\max} \mu] = \{abc\}$, but we can show that a (putatively different) revision operator $\circ$ known only to satisfy the stated postulates arrives at the same conclusion. It does so by first figuring out, using postulates R_1, R_3-R_6, $\mathrm{R}_{\mathtt{N}}$, $\mathrm{R}_{\mathtt{A}}$, $\mathrm{R}_{\mathtt{F}}$ that $[\alpha \circ f_a(\mu)] = \{bc\}$ and $[\alpha \circ f_b(\mu)] = \{abc\}$, with α, in this case, such that $[\alpha] = \{abc\}$ (see Figure 2 for an illustration). At this point, we have obtained the (flipped versions of) the models of μ at maximal Hamming distance to a and b, respectively. FOllowing this, we get that $[\gamma_{\varphi,\mu} = [\varepsilon \circ \big((\alpha \circ f_a(\mu)) \vee (\alpha \circ f_b(\mu))\big)] = \{bc\}$, where bc was obtained from abc by flipping a. Postulate* `BOW` *then be recovers abc through an extra flip of a.*

6 Characterizing the Hamming Surprise Min-Max Operator

Finally, we return to the operator $\circ^{s_{\mathrm{H}},\max}$ and, using the wisdom gained in Section 4 and 5, provide it with an axiomatic foundation. In doing so we pursue that same strategy as in the previous sections: (i) establish, axiomatically, what the revision result should be in the 'base' case in which the prior belief is of a simple type, which can be decided by appeal to an argument using appealing notions of symmetry; (ii) reduce, axiomatically, an arbitrary instance $\varphi \circ \mu$ of revision to the base case, in a manner that preserves the result of $\circ^{s_{\mathrm{H}},\max}$ on the given instance.

The base case for this section consists, as for the $\circ^{d_{\mathrm{H}},\max}$ operator, of revision when prior information is either ε or α, and we want to make sure we employ a set of postulates that deliver the expected result: since $\circ^{s_{\mathrm{H}},\max}$ behaves exactly like the Dalal and $\circ^{d_{\mathrm{H}},\max}$ operators when prior information is complete, postulates $\mathrm{R}_{\texttt{N}}$, $\mathrm{R}_{\texttt{A}}$ and $\mathrm{R}_{\texttt{F}}$ can be used without modification (the assumption of completeness made in Section 4 pays off here). We can also use the standard postulates R_1 and R_3-R_4, which we already know $\circ^{s_{\mathrm{H}},\max}$ satisfies (see Proposition 1). Postulates R_5-R_6 are, however, problematic, since $\circ^{s_{\mathrm{H}},\max}$ does not satisfy them in their unrestricted form (also Proposition 1). However, the equivalence of $\circ^{s_{\mathrm{H}},\max}$ with the Dalal and $\circ^{d_{\mathrm{H}},\max}$ operators when prior information is complete means that we can use postulates R_5 and R_6, restricted to the case when φ is complete. The restrictions are denoted R_5^c and R_6^c, respectively.

The next step involves engineering a choice situation focused on α and ε that is equivalent, in terms of what gets chosen, to the mechanics of $\circ^{s_{\mathrm{H}},\max}$. This is done using a few intermediary notions, as follows. If φ and μ are formulas such that $[\varphi] = \{v_1, \dots, v_n\}$, the *adjunction interpretations* x_1, …, x_n are interpretations consisting of completely new atoms such that $|x_i| = d_{\mathrm{H}}(v_i, \mu)$. For $v_i \in [\varphi]$, the *corrected interpretation* v_i^* is defined as $v_i^* = v_i \cup (x_1 \cup \dots x_{i-1} \cup x_{i+1} \cup \dots \cup x_n)$, i.e., as the result of adding to v_i all the adjunction interpretations, except x_i. Then, the *best-surprise formula* $\sigma_{\varphi,\mu}$ *with respect to* φ *and* μ is defined as:

$$\sigma_{\varphi,\mu} = \varepsilon \circ \left(\bigvee_{v_i \in [\varphi]} \Big(\alpha \circ f_{v_i^*}(\mu) \Big) \right).$$

In words, inside the main parenthesis we repeatedly revise α by a flipped version of μ: one revision for every model v_i of φ, flipping μ by the atoms in the corrected interpretation v_i^*. The disjunction of all these revisions is then passed on to α for another round of revision.

The reasoning behind this definition is that it recasts the surprise min-max revision scenario for $[\varphi] = \{v_1, \dots, v_n\}$ and μ into a min-max distance revision scenario for $[\varphi^*] = \{v^*, \dots, v_n^*\}$ and μ (which we know how to axiomatize from Section 5), while keeping the relative ranking of the models of μ intact. The following result makes this precise.

Lemma 6. *If* φ *and* μ *are propositional formulas,* $v_i, v_k \in [\varphi]$ *and* $w_j, w_\ell \in [\mu]$, *then* $s_{\mathrm{H}}^{\mu}(v_i, w_j) \leq s_{\mathrm{H}}^{\mu}(v_k, w_\ell)$ *iff* $d_{\mathrm{H}}(v_i^*, w_j) \leq d_{\mathrm{H}}(v_j^*, w_\ell)$.

Proof. Take $[\varphi] = \{v_1, \dots, v_n\}$, and $m_i = d_{\mathrm{H}}(v_i, \mu)$, for $v_i \in [\mu]$. We have that:

$$\begin{aligned} &s_{\mathrm{H}}^{\mu}(v_i, w_j) \leq s_{\mathrm{H}}^{\mu}(v_k, w_\ell) \text{ iff} \\ &d_{\mathrm{H}}(v_i, w_j) - m_i \leq d_{\mathrm{H}}(v_k, w_\ell) - m_k. \end{aligned}$$

We now add $\sum_{1 \leq r \leq n} m_r$ on both sides, to get an equivalence with $d_{\mathrm{H}}(v_i, w_j) + \sum_{1 \leq r \leq n, r \neq i} m_r \leq d_{\mathrm{H}}(v_k, w_\ell) + \sum_{1 \leq r \leq n, r \neq k} m_r$. This, in turn, is equivalent to $d_{\mathrm{H}}(v_i \cup (\bigcup_{1 \leq r \leq n, r \neq i} x_r), w_j) \leq d_{\mathrm{H}}(v_k \cup (\bigcup_{1 \leq r \leq n, r \neq k} x_r), w_\ell)$, which can be rewritten as $d_{\mathrm{H}}(v_i^*, w_j) \leq d_{\mathrm{H}}(v_i^*, w_\ell)$ □

Intuitively, the table of Hamming distances for $[\varphi^*] = \{v^*, \dots, v_n^*\}$ and μ can be thought of as obtained from the surprise table for φ and μ (see, e.g., Table 2) by adding a constant term (i.e., $\sum_{1 \leq r \leq n} m_r$) to every entry, a transformation that does not modify the relationships between the values: the v_i^* are the interpretations that induce the appropriate distances. This ensures that the models of $\sigma_{\varphi,\mu}$, obtained through a min-max distance type of postulate, correspond to models of μ that minimize maximum surprise with respect to φ and relative to μ, and warrants the following postulate, called *Best-of-Worst-Surprise*:

$$(\mathrm{R}_{\texttt{BOWS}}) \;\; \varphi \circ \mu \equiv \left(\bigvee\nolimits_{v \in [\varphi]} f_{v^*}(\sigma_{\varphi,\mu}) \right) \wedge \mu.$$

As expected, the $\mathrm{R}_{\texttt{BOWS}}$ postulate delivers exactly those models of μ that minimize maximum surprise, and underpins the final characterization result.

Theorem 3. *A revision operator* $\circ$ *satisfies postulates* R_1, R_3-R_4, R_5^c-R_6^c, $\mathrm{R}_{\texttt{N}}$, $\mathrm{R}_{\texttt{A}}$, $\mathrm{R}_{\texttt{F}}$ *and* $\mathrm{R}_{\texttt{BOWS}}$ *iff* $\circ \equiv \circ^{s_{\mathrm{H}},\max}$.

The following example illustrates the way in which postulate $\mathrm{R}_{\texttt{BOWS}}$ obtains the revision result.

Example 6. *Consider, again, formulas* $[\varphi] = \{a, b\}$ *and* $[\mu] = \{ac, abc\}$. *We have that* $[\varphi \circ^{s_{\mathrm{H}},\max} \mu] = \{ac, abc\}$. *Assuming we are working with an operator* $\circ$ *of which the only thing we know is that it satisfies the postulates in Theorem 3, we notice that* $d_{\mathrm{H}}(a, \mu) = 1$ *and* $d_{\mathrm{H}}(b, \mu) = 2$. *The postulates then direct us to compute the Hamming distance min-max result for* $[\varphi^*] = \{ayz, bx\}$ *and* μ, *with* x *and* yz *as the adjunction interpretations. The result obtained in this way is exactly* $\{ac, abc\}$.

7 Conclusion

We have introduced the Hamming surprise min-max operator $\circ^{s_{\mathrm{H}},\max}$, a revision operator that minimizes surprise relative to the prior belief as well as the newly acquired information. We have shown that, even though $\circ^{s_{\mathrm{H}},\max}$ does not satisfy all standard KM revision postulates, it is underpinned, in its choice behavior, by principles similar to those guiding established revision operators, among them appealing symmetry notions such as invariance under renamings and flips. When unearthed and formulated as logical postulates, these principles (or slight variations thereof) turned out to be powerful enough to fully characterize not just the surprise operator, but also the existing Dalal and Hamming distance min-max operator.

One obvious direction for future work lies in taking the idea of context dependence further: what other aspects of the environment influence an agent's plausibility rankings? Things that come to mind are issues of trust, the 'strangeness' of the new information, or peer effects. An alternative is to exploit the bottom-up, DIY nature of some of the postulates presented here in order to construct a framework, similar to that employed in collective decision-making (Cailloux and Endriss 2016), for offering *justifications* for revision results, i.e., human-readable and at the same time rigorous step-by-step arguments for how to obtain a particular result, starting from a specific set of postulates. Finally, the assumptions embedded in the present treatment call for taking the epistemic stance seriously, and investigating the relative worth of the various revision operators with respect to recovering the ground truth.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *J. Symb. Log.* 50(2):510–530.

Aravanis, T. I.; Peppas, P.; and Williams, M. 2021. An investigation of parametrized difference revision operators. *Ann. Math. Artif. Intell.* 89(1-2):7–28.

Cailloux, O., and Endriss, U. 2016. Arguing about Voting Rules. In Jonker, C. M.; Marsella, S.; Thangarajah, J.; and Tuyls, K., eds., *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, 287–295. ACM.

Dalal, M. 1988. Investigations into a Theory of Knowledge Base Revision. In *Proceedings of the 7th National Conference on Artificial Intelligence, 1988*, 475–479.

Darwiche, A., and Pearl, J. 1997. On the Logic of Iterated Belief Revision. *Artificial Intelligence* 89(1-2):1–29.

del Val, A. 1993. Syntactic Characterizations of Belief Change Operators. In Bajcsy, R., ed., *Proceedings of IJCAI 1993*, 540–547. Morgan Kaufmann.

Fermé, E. L., and Hansson, S. O. 2018. *Belief Change: Introduction and Overview*. Springer Briefs in Intelligent Systems. Springer.

Friston, K. 2010. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience* 11(2):127–138.

Gärdenfors, P., and Makinson, D. 1988. Revisions of Knowledge Systems Using Epistemic Entrenchment. In *Proceedings of TARK 1988*, 83–95.

Goldrei, D. 2005. *Propositional and Predicate Calculus*. Springer.

Grove, A. 1988. Two modellings for theory change. *Journal of Philosophical Logic* 17(2):157–170.

Hansson, S. O. 2017. Logic of Belief Revision. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2017 edition.

Haret, A., and Woltran, S. 2019. Belief Revision Operators with Varying Attitudes Towards Initial Beliefs. In *Proceedings of IJCAI 2019*, 1726–1733.

Herzig, A., and Rifi, O. 1999. Propositional Belief Base Update and Minimal Change. *Artificial Intelligence* 115(1):107–138.

Hohwy, J. 2016. The Self-Evidencing Brain. *Noûs* 50(2):259–285.

Katsuno, H., and Mendelzon, A. O. 1992. Propositional Knowledge Base Revision and Minimal change. *Artificial Intelligence* 52(3):263–294.

Lave, C. A., and March, J. G. 1993. *An Introduction to Models in the Social Sciences*. University Press of America.

Marquis, P., and Schwind, N. 2014. Lost in translation: Language independence in propositional logic-application to belief change. *Artificial Intelligence* 206:1–24.

Milnor, J. 1954. Games against nature. In Thrall, R.; Coombs, C.; and Davis, R., eds., *Decision Processes*. New York: Wiley.

Parikh, R. 1999. Beliefs, Belief Revision, and Splitting Languages. *Logic, Language and Computation* 2(96):266–268.

Peppas, P., and Williams, M. 2016. Kinetic Consistency and Relevance in belief revision. In Michael, L., and Kakas, A. C., eds., *Proceedings of JELIA 2016*, volume 10021 of *Lecture Notes in Computer Science*, 401–414.

Peppas, P., and Williams, M. 2018. Parametrised Difference Revision. In Thielscher, M.; Toni, F.; and Wolter, F., eds., *Proceedings of KR 2018*, 277–286. AAAI Press.

Peppas, P.; Williams, M.; Chopra, S.; and Foo, N. Y. 2015. Relevance in belief revision. *Artif. Intell.* 229:126–138.

Peppas, P. 2008. Belief Revision. In van Harmelen, F.; Lifschitz, V.; and Porter, B. W., eds., *Handbook of Knowledge Representation*, volume 3. Elsevier. 317–359.

Peterson, M. 2017. *An Introduction to Decision Theory*. Cambridge University Press, Second edition.

Pozos-Parra, P.; Liu, W.; and Perrussel, L. 2013. Dalal's Revision without Hamming Distance. In *Mexican International Conference on Artificial Intelligence*, 41–53. Springer.

Rott, H. 1992. Modellings for Belief Change: Base Contraction, Multiple Contraction, and Epistemic Entrenchment. In *Proceedings of JELIA '92*, 139–153.

Sen, A. 1993. Internal Consistency of Choice. *Econometrica* 61(3):495–521.

Sen, A. K. 2017. *Collective Choice and Social Welfare: Expanded Edition*. Penguin UK.

Using Conditional Independence for Belief Revision

Matthew James Lynn[1] , **James P. Delgrande**[1] , **Pavlos Peppas**[2]

[1] Simon Fraser University, Canada
[2] University of Patras, Greece

mlynn@cs.sfu.ca, jim@cs.sfu.ca, pavlos@upatras.gr

Abstract

We present an approach for incorporating qualitative conditional independence into belief revision. Our stance is that, as with probability, conditional independence arises far more frequently than the unconditional independence studied in previous work. Our approach uses multivalued dependencies to represent domain-dependent conditional independence assertions. In particular, the multivalued dependency $X \twoheadrightarrow Y$ expresses that assertions over the subvocabularies Y and $\overline{Y}$ are independent whenever complete information is known about the subvocabulary X. We introduce the class of partially compliant revision operators, wherein revising a KB satisfying $X \twoheadrightarrow Y$ by a formula expressed over Y results in the part of the KB expressed over $\overline{Y}$ remaining unchanged. This helps ensure that partially compliant revision operators result in minimal changes to existing beliefs, as irrelevant existing beliefs are left unchanged. Furthermore, we identify a subclass of partially compliant operators, called fully compliant operators, for which the same is true when revising by a formula expressed over XY rather than just Y. For both classes, we provide representation results which characterise compliance semantically in terms of faithful rankings. Finally, we compare our use of multivalued dependencies to existing work on independence in belief revision.

1 Introduction

Belief revision is concerned with the situation in which an agent is confronted with a new fact to incorporate into its belief set. If the new fact is inconsistent with the current belief set, the challenge is to revise these beliefs so that as many of the current beliefs as possible are retained while incorporating the new fact and maintaining consistency. This process is formalised as a *belief revision operator* $*$ which takes a current knowledge base K and a formula for revision ϕ and produces a revised knowledge base $K * \phi$.

In order to formalise the requirement that revision should result in a minimal change to existing beliefs, a number of authors have turned to *irrelevance*, suggesting that those beliefs irrelevant to the formula for revision should remain unchanged (Gardenfors 1990). This also has the potential advantage of opening a pathway to more efficient belief revision operators, by being able to exclude irrelevant beliefs from the revision process. However, so far, these notions of irrelevance have been extremely strict, considering beliefs as irrelevant only when there is no connection, however indirect, between them.

To see the issue, consider the following situation: an agent is informed that refrigerators require power, power is generated in the local area by wind turbines, and wind turbines kill birds. It would seem that information about birds would be independent of information concerning refrigerators; however, this is not the case, given the link between refrigerators and birds mediated by wind turbines. Consequently, existing approaches would consider refrigerators relevant to birds. However, when revising our beliefs about birds there would seem to be no reason for our beliefs about refrigerators to change. Hence it seems we need a more nuanced and general notion of irrelevance.

This situation has a parallel in probability theory. In practice, random variables are rarely independent. However, they are frequently conditionally independent. As a result, Bayesian networks have been developed to exploit conditional independence properties, thereby overcoming the otherwise seemingly-intractable complexity of probabilistic inference (Pearl 2014).

In this paper we take a suitable analogue of conditional independence for determining which beliefs may be considered irrelevant to others in a given context. We then apply this notion to belief revision, and we study those revision operators which comply with this formulation of conditional independence. Our approach is given in terms of the Katsuno-Mendelzon approach for belief revision. In our approach, we assume that conditional independence is a property of the underlying domain, and we consequently assume that a knowledge engineer has provided a collection of such conditional independence assertions. These assertions can then be taken into account in the belief revision process. To this end, we study two related notions of what it means for a belief revision operator to take into account conditional independencies. We provide postulates that characterise conditional independence in revision, and which generalise previous approaches to (non-conditional, absolute) independence. Furthermore, we provide representation results, giving conditions on faithful rankings which correspond to the sets of postulates characterising conditional independence in revision.

The next section covers background material: we first present useful definitions and notation, after which we give background material on belief revision, including existing approaches to independence in belief revision, along with conceptions of conditional independence in logic. Section 3 introduces the class of belief revision operators which *partially comply* with a multivalued dependency, and characterises partial compliance in terms of faithful rankings. Section 4 studies the stronger property of *full compliance* with a multivalued dependency, again with a characterisation in terms of faithful rankings. In Section 5 we examine and clarify the relationship between logical conditional independence, multivalued dependencies, and syntax splitting. Finally, Section 6 discusses our approach, related work, and future work, after which we have a brief conclusion.

2 Background Material

2.1 Preliminaries and Notation

Let $V = \{p, q, r, \dots\}$ be a finite set of propositional variables, arbitrary subsets of which are denoted by X, Y, and Z. We sometimes juxtapose these subsets to represent unions, e.g. $XY = X \cup Y$. The relative complement $V - X$ will be denoted by $\overline{X}$. Every subset X of V induces a propositional language $L(X)$ consisting of formulae constructed from the elements of X by applying the propositional connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. We write L for the entire propositional language $L(V)$.

Lower case Greek letters $\phi, \psi, \gamma, \dots$ will be used to range over formulae in a propositional language, with K playing a special role of a formula thought of as representing the knowledge base of an agent.

Also associated to every subset X of V is the set Ω_X of functions $v : X \rightarrow \{T, F\}$ referred to as *models* or *possible worlds* over X. We will freely think of these possible worlds as either these functions, or as conjunctions of the literals satisfied by them. Hence, for us, $\{x \mapsto T, y \mapsto F\}$ is the same thing as $x \wedge \neg y$. Given a possible word u over V alongside a subset X of V, we write u_X for the reduct of u to a possible world over X, that is the function $u_X : X \rightarrow \{T, F\}$ agreeing with u.

When ϕ is a formula we write $[\phi]$ for the set of models over V satisfying ϕ, so that $[\phi] \subseteq \Omega_V$. We write $\phi \vdash \psi$ to indicate $[\phi] \subseteq [\psi]$, and $\phi \equiv \psi$ to indicate $[\phi] = [\psi]$.

We write $V(\phi)$ for the minimal set of propositional variables for which there exists a formula ψ logically equivalent to ϕ containing only occurrences of variables in $V(\phi)$, for instance $V(q \wedge (p \vee \neg p)) = \{q\}$.

2.2 Projections of a Propositional Formula

In order to speak about components of a knowledge base K expressed in various subvocabularies we will introduce the following analogue of the *projection operator* from the relational algebra (Abiteboul, Hull, and Vianu 1995).

Definition 2.1. *If ϕ is a propositional formula, and $X \subseteq V$, then the **projection** ϕ_X **of** ϕ **onto** X is defined up to logical equivalence as the formula ϕ_X such that*

$$[\phi_X] = \{u \in \Omega_V \mid \exists v \in [\phi],\ v_X = u_X\}.$$

Example 2.1. *The projection of $(p \rightarrow q) \wedge (q \rightarrow r)$ onto $\{p, q\}$ is $(p \rightarrow q)$, whereas the projection of $(p \rightarrow q) \wedge (q \rightarrow r)$ onto $\{q, r\}$ is $(q \rightarrow r)$.*

Regarding a set of possible worlds as tuples in a relation, it follows that ϕ_X defines the set of worlds resulting from projecting this "relation" onto the "attributes" in X, then taking the Cartesian product of this with all possible interpretations of the remaining variables. This operator also appears as the notion of a uniform interpolant, a model-theoretic reduct (Hodges 1993), or as the dual of a forgetting operator[1] (Delgrande 2017). For our purposes, we will rely on the following property of projections:

Theorem 2.1. *If $\phi \vdash \psi$ and $V(\psi) \subseteq X$ then $\phi \vdash \phi_X$ and $\phi_X \vdash \psi$.*

2.3 Revision Operators and Faithful Rankings

A belief revision operator, as formalised by Alchourron, Gärdenfors, and Makinson (1985), is a binary function $*$ which maps a belief set K and a formula ϕ and produces a revised belief set $K * \phi$ in a manner satisfying the *AGM postulates*. These postulates attempt to capture the requirement that $K * \phi$ must include ϕ alongside as many beliefs from K as possible, while maintaining consistency. In other words, $K * \phi$ results from a minimal change to the existing belief set K which results in ϕ being believed. Note that belief revision captures an agent revising its beliefs about the present state of affairs, whereas updating its beliefs when the state of the world changes is the subject of *belief update operators*, cf. (Peppas 2008).

In our setting of a finite vocabulary, we can simplify matters by working instead with the Katsuno-Mendelzon approach wherein the belief sets K and $K * \phi$ are represented as single formulas, and the AGM postulates are rephrased in the following manner (Katsuno and Mendelzon 1991).

Definition 2.2. *A binary function $* : L \times L \rightarrow L$ is a **belief revision operator** if it satisfies the following **basic postulates**:*

R1. $K * \psi \vdash \psi$;

R2. *If $K \wedge \phi$ is satisfiable then $K * \phi \equiv K \wedge \phi$;*

R3. *If ϕ is satisfiable then $K * \phi$ is satisfiable;*

R4. *If $K_1 \equiv K_2$ and $\phi_1 \equiv \phi_2$ then $K_1 * \phi_1 \equiv K_2 * \phi_2$.*

*We will say that a belief revision operator $*$ satisfies the **supplementary postulates** when it satisfies the following:*

R5. $(K * \phi) \wedge \psi \vdash K * (\phi \wedge \psi)$;

R6. *If $(K*\phi) \wedge \psi$ is satisfiable then $K*(\phi \wedge \psi) \vdash (K*\phi) \wedge \psi$.*

Unless we explicitly specify that a belief revision operator satisfies the supplementary postulates, we will assume only that the basic postulates are satisfied. Note that this partitioning of the Katzuno-Mendelzon postulates into basic and supplementary postulates exactly mirrors the organisation of the original AGM postulates into basic and supplementary postulates.

When working with belief revision operators satisfying the basic and supplementary KM postulates, Katsuno and Mendelzon (1991) show that we may semantically characterise the belief revision operator as determining $K * \phi$ by selecting those worlds in $[\phi]$ which are minimally implausible with respect to a ranking on worlds. To this end, they

[1] In the sense that $\phi_Y \equiv \text{forget}(\phi, V - Y)$.

introduce binary relations $\leq_K$ on worlds referred to as *faithful rankings* wherein $u \leq_K v$ means that v is at least as implausible as u from the perspective of an agent knowing only K.

Definition 2.3. *A **faithful ranking** for K is a binary relation $\leq_K$ on possible worlds which satisfies the following properties:*

1. *$w \leq_K w'$ and $w' \leq_K w''$ implies $w \leq_K w''$.*
2. *Either $w \leq_K w'$ or $w' \leq_K w$.*
3. *$w \leq_K w'$ for all w' if and only if $w \models K$.*

If W is a set of possible worlds and $\leq$ is a faithful ranking, we write $\min(W, \leq)$ for the set of worlds in W which are minimal under $\leq$. That is to say, $x \in \min(W, \leq)$ if and only if $x \in W$ and $x \leq y$ for all $y \in W$.

Theorem 2.2 ((Katsuno and Mendelzon 1991))**.** *A binary function $* : L \times L \to L$ is a belief revision operator satisfying the supplementary postulates if and only if for every K there exists a faithful ranking $\leq_K$ for K such that $[K * \phi] = \min([\phi], \leq_K)$.*

2.4 Relevance in Belief Revision

Although the general consensus is that a belief revision operator must satisfy the KM postulates, these postulates place few constraints on the behaviour of belief revision operators. For instance, they fail to rule out the belief revision operator defined by setting $K * \phi = K \wedge \phi$ if $K \wedge \phi$ is consistent and $K * \phi = \phi$ otherwise[2]. This is in tension with the objective of belief revision to preserve as many of the original beliefs as possible.

In (Parikh 1999) the notion of minimal change is addressed via considering an additional postulate asserting that whenever the knowledge base is divisible into two unrelated components, then revision by a formula pertaining to only one of those components should leave the other component unchanged. For a KM belief revision operator $*$, Parikh's postulate can be expressed as follows:

P If $K \equiv K_1 \wedge K_2$ where $V(K_1) \subseteq X_1$, $V(K_2) \subseteq X_2$, $X_1 \cap X_2 = \emptyset$, and ϕ is such that $V(\phi) \subseteq X_1$ then

$$K * \phi \equiv (K_1 \circledast \phi) \wedge K_2$$

where $\circledast$ is a belief revision operator for the language X_1.

The statement of Parikh's postulate admits a weak reading wherein $\circledast$ varies as a function of K, as well as a strong reading wherein $\circledast$ is fixed. In order to clarify this situation, Peppas et al.(2015) introduced the following variations (P1) and (P2) of (P) which we state here in the KM setting:

P1. If $V(K_1) \cap V(K_2) = \emptyset$ and $V(\phi) \subseteq V(K_1)$ then $((K_1 \wedge K_2) * \phi)_{V(K_2)} \equiv K_2$.

P2. If $V(K_1) \cap V(K_2) = \emptyset$ and $V(\phi) \subseteq V(K_1)$ then $((K_1 \wedge K_2) * \phi)_{V(K_1)} \equiv (K_1 * \phi)_{V(K_1)}$.

Intuitively, (P1) states that when revising K by ϕ, only the part of K relevant to ϕ is revised. The role of (P2) is to ensure that whenever K_1 and K_2 agree on the beliefs relevant to ϕ, then the revisions $K_i * \phi$ change this part in the same way.

Using these clarified postulates, Peppas et al. (2015) develop a characterisation of those belief operators satisfying (P1) and (P2), and show that Dalal's belief revision operator satisfies the basic and supplementary KM postulates as well as (P1) and (P2). Subsequent work has extended these results to epistemic states (Kern-Isberner and Brewka 2017), to belief contraction operators (Haldimann, Kern-Isberner, and Beierle 2020), to epistemic entrenchments and selection functions (Aravanis, Peppas, and Williams 2019), and to preferential entailment relations (Kern-Isberner, Beierle, and Brewka 2020).

Rather than considering belief revision operators that satisfy (P1), Delgrande and Pappas (2018) consider belief revision operators which satisfy an analogue of Parikh's postulate for only certain theories and a subset of possible syntax splittings. The idea is that the knowledge engineer will specify a number of *irrelevance assertions* $\sigma \twoheadrightarrow Y$[3], and belief revision operators will be required to comply with these assertions in the following sense:

Definition 2.4. *A belief revision operator $*$ **complies** with $\sigma \twoheadrightarrow Y$ at K when either $K \nvdash \sigma$ or for every consistent ϕ with $V(\phi) \subseteq Y$ the following postulate is satisfied:*

R *If $K \vdash \neg\phi$ then $K * \phi \equiv (K * \phi)_Y \wedge K_{\overline{Y}}$.*

For a belief revision operator $*$ induced from a family of faithful rankings $\{\leq_K\}_K$, Delgrande and Pappas (2018) show that complying with $\sigma \twoheadrightarrow Y$ is equivalent to stating that, for every K entailing σ, the following postulates are satisfied:

S1. If $u_Y = v_Y$, $K \vdash \neg u_Y$, and $K_{\overline{Y}} \nvdash \neg u$ then $u \leq_K v$;

S2. If $u_Y = v_Y$, $K \vdash \neg u_Y$, $K_{\overline{Y}} \nvdash \neg u$, and $K_{\overline{Y}} \vdash \neg v$ then $u <_K v$;

2.5 Conditional Independence

Parikh's postulate, and the majority of approaches descending from it, suffers from the limitation that the knowledge base must be able to be split into disjoint components in order for the postulate to apply. This limitation is already noted in (Chopra and Parikh 2000) which attempts to overcome this limitation by introducing the notion of a *belief structure*, which splits a knowledge base into a number of compartments which may overlap in vocabulary. However this compartmentalisation is fixed which can lead to information being lost.

This situation has an analogue in probability theory, where unconditional independence is a powerful but rarely applicable assumption. Rather, it is conditional independence which arises most frequently, and in fact has become a central component of modern probabilistic modelling and inference.

Inspired by probability theory, Darwiche (1997) introduces a notion of conditional logical independence together

[2] Consider the rankings $\leq_K$ where $u \leq_K v$ for all $u, v \notin [K]$.

[3] For the reader familiar with multivalued dependencies, the similarity of this notation was a deliberate choice in (Delgrande and Peppas 2018).

with a number of equivalent characterisations tailored for different reasoning problems. We will adopt the following notion, adapted from (Lang and Marquis 1998) and (Lang, Liberatore, and Marquis 2002).

Definition 2.5. *If X, Y_1, and Y_2 are pairwise disjoint subsets of V and K is a propositional formula over V then Y_1 and Y_2 are* ***conditionally independent given*** *X* ***modulo*** *K when for any world u and formulae ϕ_1 and ϕ_2 with $V(\phi_1) \subseteq Y_1$ and $V(\phi_2) \subseteq Y_2$ such that $K \wedge u_X \vdash \phi_1 \vee \phi_2$ either $K \wedge u_X \vdash \phi_1$ or $K \wedge u_X \vdash \phi_2$.*

Example 2.2. *The sets $\{p\}$ and $\{r\}$ are conditionally independent given $\{q\}$ modulo $K := (p \to q) \wedge (q \to r)$. This follows from Theorem 5.2 below. To verify this for a specific case, let u be an arbitrary possible world and consider that $K \wedge u_{\{q\}} \vdash \neg p \vee r$. Either $u(q) = F$ in which case $K \wedge u_{\{q\}} \vdash \neg p$, or $u(q) = T$ in which case $K \wedge u_{\{q\}} \vdash r$, as required.*

Taking inspiration instead from database theory, we can regard the worlds satisfying a propositional formula K as constituting a database table wherein the attributes are the propositional variables in V. Then, we may consider the notion of a multivalued dependency:

Definition 2.6. *A propositional formula K satisfies the* ***multivalued dependency*** *$X \twoheadrightarrow Y$ when for any models v and u of K such that $v_X = u_X$ there exists a model w of K such that $w_Y = v_Y$ and $w_{\overline{Y}} = u_{\overline{Y}}$.*

Example 2.3. *The formula $K = (p \to q) \wedge (q \to r) \wedge (q \wedge r \to s)$ satisfies the multivalued dependencies $\{q\} \twoheadrightarrow \{p\}$ and $\{q\} \twoheadrightarrow \{r, s\}$.*

In Section 5 we show that multivalued dependencies are equivalent to a restricted case of conditional independence, and that both are equivalent to a generalisation of Parikh's syntax-splittings.

3 Compliance with Multivalued Dependencies

Parikh's original postulate considers only unconditional independence. However unconditional independence is a strong condition which is unrealistic to expect to hold often. Consider even a seemingly clear situation, such as a knowledge base containing knowledge about birds and knowledge about refrigerators. These topics would seem to be independent. However, suppose we have that refrigerators require power, power is generated in the local area by wind turbines, and wind turbines often kill birds. Now, the ability to split the knowledge base is gone. However, we can observe that if the only link between birds and refrigerators passes through the language of wind turbines, then when revising knowledge about birds, our knowledge concerning refrigerators is not impacted, provided that our knowledge of wind turbines is unaffected.

In our approach, the knowledge engineer will represent their understanding of conditional independencies between components of the knowledge base as a collection of multivalued dependencies. The intuitive interpretation being that a multivalued dependency $X \twoheadrightarrow Y$ captures that the only connections between knowledge over Y and knowledge over $\overline{Y}$ arise from knowledge over X. In our example scenario, knowledge about turbines comprises the only connection between birds and refrigerators, so the knowledge engineer would represent this via the multivalued dependencies $TurbineVocabulary \twoheadrightarrow BirdVocabulary$ and $TurbineVocabulary \twoheadrightarrow RefrigeratorVocabulary$.

Once the knowledge engineer has selected a collection of multivalued dependencies which capture the conditional independence relations between different areas of knowledge being worked with, these multivalued dependencies are incorporated into the belief revision process by requiring *compliance* in the following sense:

Definition 3.1. *If X and Y are disjoint subsets of V then a belief revision operator $*$* ***partially complies with*** *$X \twoheadrightarrow Y$ if the following postulate holds:*

PCR. *If K is consistent and satisfies $X \twoheadrightarrow Y$, $V(\phi) \subseteq Y$, and ϕ is consistent then*

$$K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}.$$

Any belief revision operator partially complying with $X \twoheadrightarrow Y$ must, when revising a knowledge base satisfying $X \twoheadrightarrow Y$ by a consistent formula over Y, preserve the $\overline{Y}$ component of the knowledge base unchanged. Returning to our example, supposing our knowledge base K satisfies $TurbineVocabulary \twoheadrightarrow BirdVocabulary$ and we revise by some formula ϕ in the bird vocabulary, we would have that knowledge over $\overline{BirdVocabulary}$ is preserved. In particular, our beliefs concerning the relationship between turbines and refrigerators could not be changed by any formula ϕ only referring to birds.

We refer to this as only partial compliance, for in the next section we will introduce a postulate which applies to suitable ϕ with $V(\phi) \subseteq XY$ rather than just for ϕ with $V(\phi) \subseteq Y$.

3.1 Representation via Faithful Rankings

Those belief revision operators which partially comply with a multivalued dependency can be characterised semantically by conditions on their corresponding faithful rankings.

Definition 3.2. *If $\leq_K$ is a faithful ranking for K then $\leq_K$* ***partially respects*** *$X \twoheadrightarrow Y$ if either K does not satisfy $X \twoheadrightarrow Y$ or the following conditions are satisfied:*

PCS1. *If $u_{XY} = v_{XY}$, $K \vdash \neg u_Y$, $u \in [K_{\overline{Y}}]$, and $v <_K u$ then there exists w such that $w_Y = u_Y$ and $w <_K v$.*

PCS2. *If $K_{\overline{Y}} \vdash \neg v$ then there exists a world $u \in [K_{\overline{Y}}]$ such that $u_Y = v_Y$ and $u <_K v$.*

Condition (PCS1) states that when worlds u and v with $u_{XY} = v_{XY}$ are ruled out by K on the basis of u_Y, yet u is consistent with $K_{\overline{Y}}$, then either u is at least as plausible as v or there is some world w with $w_Y = u_Y$ strictly more plausible than both u and v. Condition (PCS2) further states that a possible world v inconsistent with $K_{\overline{Y}}$ is always less plausible than some possible world u satisfying $K_{\overline{Y}}$, and furthermore such a u may be obtained from v by modifying only the variables in $\overline{Y}$.

Theorem 3.1. *If $*$ is a belief revision operator satisfying the supplementary postulates which partially complies with $X \twoheadrightarrow Y$, then there exist faithful rankings $\{\leq_K\}_K$ which partially respect $X \twoheadrightarrow Y$ such that $[K * \phi] = \min([\phi], \leq_K)$ for all K and ϕ.*

Proof. By Theorem 2.2 there exist faithful rankings $\{\leq_K\}_K$ such that $[K * \phi] = \min([\phi], \leq_K)$ for all K and ϕ. Suppose $*$ partially complies with $X \twoheadrightarrow Y$ and consider a consistent formula K. In the case K does not satisfy $X \twoheadrightarrow Y$ then $\leq_K$ partially respects $X \twoheadrightarrow Y$ in the trivial sense. Otherwise, K satisfies $X \twoheadrightarrow Y$ and we must demonstrate that $\leq_K$ satisfies (PCS1) and (PCS2).

Part 1. Suppose that (PCR) holds. In order to verify (PCS1), suppose that u and v are worlds such that $u_{XY} = v_{XY}$, $K \vdash \neg u_Y$, and $u \in [K_{\overline{Y}}]$, and $v <_K u$. Applying (PCR) it follows that

$$[(K * u_Y)_{XY}] \cap [K_{\overline{Y}}] = [K * u_Y].$$

Assume for the sake of contradiction that $u \in [K * u_Y]$. As $u \in [u_Y]$ and $u_{XY} = v_{XY}$ it follows that $v \in [u_Y]$, which means that $u \leq_K v$. However, this contradicts our assumption that $v <_K u$, so it must be the case that $u \notin [K * u_Y]$. Therefore, as $u \in [K_{\overline{Y}}]$, it follows that $u \notin [(K * u_Y)_{XY}]$, and thus $v \notin [(K * u_Y)_{XY}]$ as $u_{XY} = v_{XY}$. Theorem 2.1 implies that $K * u_Y \vdash (K * u_Y)_{XY}$, hence $v \notin [K * u_Y]$. However, as $[u_Y] \neq \emptyset$ there must exist some world $w \in [K * u_Y]$. It follows that $w_Y = u_Y$, and furthermore as $v \in [u_Y]$ yet $v \notin [K * u_Y]$ it follows that $w <_K v$ as required. Therefore, (PCS1) is satisfied.

Part 2. In order to verify (PCS2) suppose that v is a world such that $K_{\overline{Y}} \vdash \neg v$. Applying (PCR) it follows that

$$[K * v_Y] = [(K * v_Y)_{XY}] \cap [K_{\overline{Y}}].$$

By our supposition that $K_{\overline{Y}} \vdash \neg v$ it follows that $v \notin [K_{\overline{Y}}]$, and therefore $v \notin [K * v_Y]$. However, as $[v_Y] \neq \emptyset$ it follows that $[K * v_Y] \neq \emptyset$. Let $u \in [K * v_Y]$ be arbitrary, and observe that $u \in [v_Y]$ meaning $u_Y = v_Y$. As $v \in [v_Y]$ but $v \notin [K * v_Y]$ it follows then that $u <_K v$ as required. Therefore, (PCS2) holds. □

Theorem 3.2. *If $\{\leq_K\}_K$ are faithful rankings which partially respect $X \twoheadrightarrow Y$, then the binary function defined by $[K * \phi] = \min([\phi], \leq_K)$ is a belief revision operator satisfying the supplementary postulates which partially complies with $X \twoheadrightarrow Y$.*

Proof. By Theorem 2.2 it follows that $*$ is a belief revision operator. Suppose K is a consistent formula such that $\leq_K$ partially respects $X \twoheadrightarrow Y$. In the case K does not satisfy $X \twoheadrightarrow Y$ there is nothing to check, so assume K satisfies $X \twoheadrightarrow Y$. This means that $\leq_K$ satisfies (PCS1) and (PCS2). Using this, we must demonstrate that $[K * \phi] = [(K*\phi)_{XY}] \cap [K_{\overline{Y}}]$ whenever $V(\phi) \subseteq Y$ and ϕ is consistent. In the case $K \wedge \phi$ is consistent then $K * \phi \equiv K \wedge \phi$, and K satisfying $X \twoheadrightarrow Y$ means $K \equiv K_{XY} \wedge K_{\overline{Y}}$ (cf. Theorem 5.3 below), hence $K * \phi \equiv K_{XY} \wedge K_{\overline{Y}} \wedge \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$. Therefore, we will assume $K \vdash \neg\phi$, in which case our proof has two parts:

Part 1. In order to show $[(K * \phi)_{XY}] \cap [K_{\overline{Y}}] \subseteq [K * \phi]$ suppose that $u \in [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$. Being that $u \in [(K * \phi)_{XY}]$ it follows that there exists $v \in [K * \phi]$ such that $u_{XY} = v_{XY}$. By our assumption that $K \vdash \neg\phi$ and the observation that $v_Y \vdash \phi$, it follows that $K \vdash \neg v_Y$. Being that $u_{XY} = v_{XY}$ it follows that $K \vdash \neg u_Y$. Assume for the sake of contradiction that $v <_K u$. It then follows from (PCS1) that there exists w with $w_Y = u_Y = v_Y$ and $w <_K v$. However, $v \in [\phi]$ and $w_Y = v_Y$ implies $w \in [\phi]$, and therefore $v \in [K * \phi]$ implies $v \leq_K w$ which contradicts our assumption that $w <_K v$. Therefore, our assumption was wrong, so it must be the case that $u \leq_K v$. This means that $u \in [\phi]$ and $v \in \min([\phi], \leq_K)$ which implies $u \in \min([\phi], \leq_K) = [K * \phi]$. Thus, as u was arbitrary, it follows that $[(K * \phi)_{XY}] \cap [K_{\overline{Y}}] \subseteq [K * \phi]$.

Part 2. In order to show $[K * \phi] \subseteq [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$ start by observing that $K * \phi \vdash (K * \phi)_{XY}$ by Theorem 2.1. Therefore, it suffices to verify that $[K * \phi] \subseteq [K_{\overline{Y}}]$. Suppose that $v \in [K * \phi]$ but assume for the sake of contradiction that $v \notin [K_{\overline{Y}}]$. It follows that $K_{\overline{Y}} \vdash \neg v$, and therefore by (PCS2) there exists a world $u \in [K_{\overline{Y}}]$ such that $u_Y = v_Y$ and $u <_K v$. Observing that $v \in [\phi]$, $V(\phi) \subseteq Y$, and $u_Y = v_Y$ it follows that $u \in [\phi]$. However, by our assumption that $v \in [K * \phi]$ this implies $v \leq_K u$ which is a contradiction as $u <_K v$. Therefore, it must be that $v \in [K_{\overline{Y}}]$. As v was arbitrary, it follows that $[K * \phi] \subseteq [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$ as required.

It follows that $K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$, showing that (PCR) holds. □

3.2 Existence of Partially Compliant Operators

Parikh (1999) demonstrates the existence of a belief revision operator satisfying postulate **P** as follows: Given a knowledge base K and a formula ϕ to revise by which is inconsistent with K, first K is split as $K_Y \wedge K_{\overline{Y}}$ where Y is the smallest subset of V with $V(\phi) \subseteq Y$ and K satisfies $\emptyset \twoheadrightarrow Y$. K is then replaced by $\phi \wedge K_{\overline{Y}}$. In order to mirror this, we need to show that we can construct such an analogous Y, which we refer to here as a *section*:

Definition 3.3. *An X-**section** of ϕ is a subset $Y \subseteq V(\phi)$ disjoint from X such that ϕ satisfies $X \twoheadrightarrow Y$.*

In order to construct a smallest section, we will make use of the following properties of multivalued dependencies:

Lemma 3.1 (Abiteboul, Hull, and Vianu (1995)). *1. If $X \twoheadrightarrow Y$ then $X \twoheadrightarrow V - Y$;*

2. If $Y \subseteq X$ then $X \twoheadrightarrow Y$;

3. If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow X$ then $X \twoheadrightarrow Z$;

4. If $X \twoheadrightarrow Y$ then $XZ \twoheadrightarrow YZ$;

For any set of variables X we can consider the set $d_K(X)$ of Y such that K satisfies $X \twoheadrightarrow Y$, that is

$$d_K(X) := \{Y \mid K \text{ satisfies } X \twoheadrightarrow Y\}.$$

An important consequence of Lemma 3.1 is that $d_K(X)$ forms a Boolean algebra:

Corollary 3.1 (Abiteboul, Hull, and Vianu (1995)). *For any K and $X \subseteq V$ it follows that $d_K(X)$ is a Boolean algebra, i.e. $d_K(X)$ is closed under unions, intersections, complementation, and it contains X.*

Theorem 3.3 (Conditional Sectioning Theorem)**.** *If there is an X-section of K containing $V(\phi)$ then there is a unique smallest X-section of K containing $V(\psi)$.*

Proof. Simply take the intersection of all $Y \in d_K(X)$ such that $V(\phi) \subseteq Y$. □

Theorem 3.4. *For every $X \subseteq V$ there exists a belief revision operator $*$ which satisfies the basic postulates and partially complies with every $X \twoheadrightarrow Y$ where $Y \subseteq V$ is disjoint from X.*

Proof. Construct a belief revision operator $*$ as follows. For every K and ϕ define $K * \phi$ as $K \wedge \phi$ in the case $K \wedge \phi$ is consistent. Otherwise, if there is an X-section of K containing $V(\phi)$ choose the smallest X-section Y of K containing $V(\phi)$ and define $K*\phi$ as $(K)_{\overline{Y}} \wedge \phi$. Otherwise, define $K*\phi$ as ϕ. □

4 Full Compliance with Multivalued Dependencies

Consider again an agent aware of wind turbines killing birds, and powering refrigerators, but with no knowledge directly linking birds and refrigerators. Suppose that this agent is given a new fact that modern wind turbines stop momentarily when an approaching bird is detected, in order to allow its safe passage, and consider how the agent may revise its knowledge base. A revision operator that partially complies with $TurbineVocabulary \twoheadrightarrow BirdVocabulary$ is not useful here, since we are revising by a formula in the language of both turbines and birds. However, since the new knowledge is consistent with the fact that turbines power refrigerators, it seems that there is no reason why knowledge about refrigerators should be changed. Thus, we can consider a stronger notion of compliance wherein we can revise by knowledge containing the shared variables about turbines.

Definition 4.1. *If X and Y are disjoint subsets of V then a belief revision operator $*$* ***fully complies with*** *$X \twoheadrightarrow Y$ if the following postulate holds:*

CR. *If K is consistent and satisfies $X \twoheadrightarrow Y$, $V(\phi) \subseteq XY$, and $\phi \wedge K_{\overline{Y}}$ is consistent then*

$$K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}.$$

Requiring that a belief revision operator fully comply with $X \twoheadrightarrow Y$ is stronger than requiring that it partially comply with $X \twoheadrightarrow Y$, for the reason that (CR) applies to a broader class of formulae. Consequently, we obtain the following relationship between full and partial compliance:

Theorem 4.1. *If X and Y are disjoint subsets of V and $*$ is a belief revision operator which fully complies with $X \twoheadrightarrow Y$, then $*$ partially complies with $X \twoheadrightarrow Y$.*

Proof. Suppose K is a consistent formula satisfying $X \twoheadrightarrow Y$, and ϕ is a consistent formula with $V(\phi) \subseteq Y$. As K and ϕ are consistent and $V(K_{\overline{Y}}) \cap V(\phi) = \emptyset$ it follows that $K_{\overline{Y}} \wedge \phi$ is consistent, and hence we may apply (CR) to write $K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$. Which is exactly what was required to show (PCR) is satisfied. Hence, $*$ partially complies with $X \twoheadrightarrow Y$. □

4.1 Representation via Faithful Rankings

As with (PCR), the postulate (CR) can be characterised in terms of conditions (CS1), (CS2), and (CS3) on faithful rankings. The stronger nature of (CR) will result in (CS1) and (CS2) appearing much closer to the original conditions (S1) and (S2) introduced in (Delgrande and Peppas 2018).

Definition 4.2. *If $\leq_K$ is a faithful ranking for K then $\leq_K$* ***fully respects*** *$X \twoheadrightarrow Y$ if either K does not satisfy $X \twoheadrightarrow Y$ or the following conditions are satisfied:*

CS1. *If $u_{XY} = v_{XY}$, $K \vdash \neg u_{XY}$, and $K_{\overline{Y}} \nvdash \neg u$ then $u \leq_K v$.*

CS2. *If $u_{XY} = v_{XY}$, $K \vdash \neg u_{XY}$, $K_{\overline{Y}} \nvdash \neg u$, and $K_{\overline{Y}} \vdash \neg v$ then $u <_K v$.*

CS3. *If $K \vdash \neg u_{XY}$, $K \vdash \neg v_{XY}$, and $K_{\overline{Y}} \nvdash \neg u_{XY}$ and $K_{\overline{Y}} \vdash \neg v_{XY}$ then there exists w with $w_{XY} = u_{XY}$ and $w <_K v$.*

Condition (CS1) states that whenever worlds u and v incompatible with K are such that $u_{XY} = v_{XY}$, and u is consistent with $K_{\overline{Y}}$, then v cannot be more plausible than u. In the case v is itself inconsistent with $K_{\overline{Y}}$, then (CS2) strengthens this to say that u is strictly more plausible than v. Finally, (CS3) ensures that whenever u is compatible with $K_{\overline{Y}}$ and v is not, then u can be modified to be strictly more plausible than v by modifying variables not in XY.

Demonstrating that a belief revision operator fully complying with $X \twoheadrightarrow Y$ results in the conditions (CS1), (CS2), and (CS3) being satisfied for the corresponding faithful rankings proceeds along lines strongly reminiscent to Theorem 2 of (Delgrande and Peppas 2018).

Theorem 4.2. *If $*$ is a belief revision operator satisfying the supplementary postulates which fully complies with $X \twoheadrightarrow Y$, then there exist faithful rankings $\{\leq_K\}_K$ which fully respects $X \twoheadrightarrow Y$ such that $[K * \phi] = \min([\phi], \leq_K)$ for all K and ϕ.*

Proof. By Theorem 2.2 there exist faithful rankings $\{\leq_K\}_K$ such that $[K * \phi] = \min([\phi], \leq_K)$ for all K and ϕ. Suppose that $*$ fully complies with $X \twoheadrightarrow Y$, and consider K satisfying $X \twoheadrightarrow Y$. We must show that $\leq_K$ satisfies the conditions (CS1), (CS2), and (CS3).

Part 1. Suppose u and v are worlds such that $u_{XY} = v_{XY}$, $K \vdash \neg u_{XY}$, and $K_{\overline{Y}} \nvdash \neg u$. This last assumption implies that u_{XY} is consistent with $K_{\overline{Y}}$, hence we may apply the postulate (CR) to write

$$\begin{aligned}[K * u_{XY}] &= [(K * u_{XY})_{XY}] \cap [K_{\overline{Y}}] \\ &= [u_{XY}] \cap [K_{\overline{Y}}].\end{aligned}$$

As $K_{\overline{Y}} \nvdash \neg u$ it follows that $u \in [K_{\overline{Y}}]$, and tautologically $u \in [u_{XY}]$, so it follows that $u \in [K * u_{XY}]$. Hence, as $v \in [u_{XY}]$ it follows that $u \leq v$ verifying (CS1).

Part 2. In order to see (CS2) suppose further that $K_{\overline{Y}} \vdash \neg v$. In this case, $v \notin [K * u_{XY}]$ hence $u < v$ verifying (CS2).

Part 3. In order to verify (CS3) suppose u and v are worlds such that $K \vdash \neg u_{XY}$, $K \vdash \neg v_{XY}$, $K_{\overline{Y}} \nvdash u_{XY}$ and $K_{\overline{Y}} \vdash \neg v_{XY}$. Construct the formula $\phi = u_{XY} \vee v_{XY}$ and observe that v_{XY} is consistent with $K_{\overline{Y}}$ and hence ϕ

is consistent with $K_{\overline{Y}}$. However, ϕ is inconsistent with K by our hypothesis. Therefore, we may apply (CR) to write $K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$. By the success postulate, $K * \phi \vdash \phi \equiv u_{XY} \vee v_{XY}$. By (CR) we also know $K * \phi \vdash K_{\overline{Y}}$. However, we also know $K_{\overline{Y}} \vdash \neg v_{XY}$, and therefore it follows that $K * \phi \vdash u_{XY}$. Hence, choosing any $w \in [K * \phi]$ it follows that $w_{XY} = u_{XY}$ and $w \leq v$. Being that $v \notin [K * \phi]$ yet $v \in [\phi]$ it follows that $w < v$. □

Theorem 4.3. *If $\{\leq_K\}_K$ are faithful rankings which fully respects $X \twoheadrightarrow Y$, then the binary function defined by $[K * \phi] = \min([\phi], \leq_K)$ is a belief revision operator satisfying the supplementary postulates which fully complies with $X \twoheadrightarrow Y$.*

Proof. By Theorem 2.2 it follows that $*$ is a belief revision operator. Suppose $\leq_K$ fully respects $X \twoheadrightarrow Y$. In the case K does not satisfy $X \twoheadrightarrow Y$ then there is nothing to verify. Assume K satisfies $X \twoheadrightarrow Y$, so that $\leq_K$ satisfies (CS1), (CS2), and (CS3). We must demonstrate that $[K * \phi] = [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$ whenever $V(\phi) \subseteq XY$ and $\phi \wedge K_{\overline{Y}}$ is consistent. In the case $K \wedge \phi$ is consistent then $K * \phi \equiv K \wedge \phi$, and K satisfying $X \twoheadrightarrow Y$ means $K \equiv K_{XY} \wedge K_{\overline{Y}}$ (cf. Theorem 5.3 below), hence $K * \phi \equiv K_{XY} \wedge K_{\overline{Y}} \wedge \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$. Therefore, we will assume $K \vdash \neg\phi$, in which case our proof has two parts:

Part 1. In order to show $[(K * \phi)_{XY}] \cap [K_{\overline{Y}}] \subseteq [K * \phi]$ suppose that $u \in [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$. Being that $u \in [(K * \phi)_{XY}]$ it follows that there exists $v \in [K * \phi]$ such that $u_{XY} = v_{XY}$. Observe that $K \vdash \neg\phi$, and furthermore $\neg\phi \vdash \neg u_{XY}$ as $u_{XY} = v_{XY}$ and $v \in [\phi]$. Hence, $K \vdash \neg u_{XY}$. However, $u \in [K_{\overline{Y}}]$ so $K_{\overline{Y}} \nvdash \neg u$. Therefore, by (CS1) it follows that $u \leq_K v$. However, $u \in [\phi]$ and $v \in \min([\phi], \leq_K)$ so it follows that $u \in \min([\phi], \leq_K) = [K * \phi]$. With u being arbitrary, it follows that $[(K * \phi)_{XY}] \cap [K_{\overline{Y}}] \subseteq [K * \phi]$ as required.

Part 2. In order to show $[K * \phi] \subseteq [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$ consider a world $v \in [K * \phi] = \min([\phi], \leq_K])$, and observe that $v \in [(K * \phi)_{XY}]$ hence it suffices to show $v \in [K_{\overline{Y}}]$. Assume for the sake of contradiction that $v \notin [K_{\overline{Y}}]$, which is to say that $K_{\overline{Y}} \vdash \neg v$. We have two cases:

1. In the case there exists a world u with $u_{XY} = v_{XY}$, and $K_{\overline{Y}} \nvdash \neg u$, argue as follows. As $u_{XY} = v_{XY}$ and $v \in [\phi]$ it follows that $u \in [\phi]$, and hence $\neg\phi \vdash \neg u_{XY}$. Observing that $K \vdash \neg\phi$ it follows that $K_{\overline{Y}} \vdash \neg u_{XY}$. As $v \notin [K_{\overline{Y}}]$ by our assumption, it follows that $K_{\overline{Y}} \vdash \neg v$. Hence, by (CS2), it follows that $u <_K v$. However, $u \in [\phi]$ and $v \in \min([\phi], \leq_K)$ so this is a contradiction.
2. In the other case, $K_{\overline{Y}} \vdash \neg v_{XY}$. Recalling that $K_{\overline{Y}}$ is consistent with ϕ, it follows that there exists a world $u \in [K_{\overline{Y}} \wedge \phi]$, for which we know that $K \vdash \neg u_{XY}$ and $K_{\overline{Y}X} \nvdash \neg u_{XY}$. However, by (CS3) we may conclude that there exists w with $w_{XY} = u_{XY}$ such that $w < v$. As ϕ is expressed over the vocabulary XY and $u \in [\phi]$ it follows that $w \in [\phi]$ and $w <_K v$. However, this contradicts $v \in \min([\phi], \leq_K)$.

In both cases, a contradiction is achieved, so our assumption that $v \notin [K_{\overline{Y}}]$ must have been false. Hence, $v \in [K_{\overline{Y}}]$ as well. With v being arbitrary, we have shown $[K * \phi] \subseteq [(K * \phi)_{XY}] \cap [K_{\overline{Y}}]$. □

4.2 Existence of Fully Compliant Operators

With this representation result in hand, the next question is whether there exists a belief revision operator which fully complies with an arbitrary multivalued dependency $X \twoheadrightarrow Y$ where X need not be empty. Fortunately, the answer is affirmative:

Theorem 4.4. *If X and Y are disjoint then there exists a belief revision operator $*$ satisfying the supplementary postulates which fully complies with $X \twoheadrightarrow Y$.*

Proof. It suffices to construct a family of faithful rankings $\{\leq_K\}_K$ where each $\leq_K$ fully respects $X \twoheadrightarrow Y$, in which case the corresponding belief revision operator $*$ with $[K * \phi] = \min([\phi], \leq_K)$ will fully comply with $X \twoheadrightarrow Y$. Given K define the function $\rho_K : \Omega \to \mathbb{N}$ given by

$$\rho_K(u) := \begin{cases} 0 & \text{if } u \in [K] \\ 1 & \text{if } u \notin [K] \text{ and } K_{\overline{Y}} \nvdash \neg u \\ 2 & \text{otherwise} \end{cases}$$

The ranking $\leq_K$ is defined by setting $u \leq_K v$ if and only if $\rho_K(u) \leq \rho_K(v)$. As $\rho_K(u) = 0$ if and only if $u \in [K]$, it follows that the minimal worlds under $\leq_K$ are exactly the worlds satisfying K. Hence, $\leq_K$ is a faithful ranking for K.

In order to argue $\leq_K$ fully respects $X \twoheadrightarrow Y$ assume that K satisfies $X \twoheadrightarrow Y$, and verify (CS1), (CS2), and (CS3) as follows:

1. (CS1) Suppose that $u_{XY} = v_{XY}$, $K \vdash \neg u_{XY}$, and $K_{\overline{Y}} \nvdash \neg u$. It follows that $r_K(u) = 1$ and $r_K(v) \geq 1$, hence $u \leq_K v$ as required.
2. (CS2) Suppose that $u_{XY} = v_{XY}$, $K \vdash \neg u_{XY}$, $K_{\overline{Y}} \nvdash \neg u$, and $K_{\overline{Y}} \vdash \neg v$. It follows that $r_K(u) = 1$ and $r_K(v) = 2$, hence $u <_K v$ as required.
3. (CS3) Suppose that $K \vdash \neg u_{XY}$, $K \vdash \neg v_{XY}$, $K_{\overline{Y}} \nvdash \neg u_{XY}$, and $K_{\overline{Y}} \vdash \neg v_{XY}$. As a consequence of $K_{\overline{Y}} \vdash \neg v_{XY}$ it follows that $\rho_K(v) = 2$. As $K_{\overline{Y}} \nvdash \neg u_{XY}$ there exists a world w with $w_{XY} = u_{XY}$ and $w \in [K_{\overline{Y}}]$, so that $\rho_K(w) = 1$ and hence $w <_K u$.

□

This leaves the open question of whether any set of multivalued dependencies can be simultaneously fully complied with by some belief revision operator.

5 Syntax Splitting and MVDs

5.1 Syntax Splitting and Conditional Independence

In this section we demonstrate that Parikh's syntax splitting generalises naturally into the framework of multivalued dependencies and conditional independence. We start by showing that syntax splitting gives rise to conditional logical independence via leveraging Craig's Interpolation Theorem (Craig 1957), which is stated as follows:

Theorem 5.1 (Craig's Interpolation Theorem). *If $K \vdash \psi$ then there exists ϕ with $V(\phi) \subseteq V(K) \cap V(\psi)$ such that $K \vdash \phi$ and $\phi \vdash \psi$.*

Theorem 5.2 (The Splitting Criterion). *If Y_1, Y_2, and X are pairwise disjoint sets of propositional variables then for any propositional formulae K_1 and K_2 such that $V(K_1) \subseteq Y_1X$ and $V(K_2) \subseteq Y_2X$ it follows that Y_1 and Y_2 are independent given X modulo $K_1 \wedge K_2$.*

Proof. Suppose u is a world, and ϕ_1 and ϕ_2 are propositional formulae with $V(\phi_1) \subseteq Y_1$ and $V(\phi_2) \subseteq Y_2$ such that $K_1 \wedge K_2 \wedge u_X \vdash \phi_1 \vee \phi_2$. We must demonstrate that either $K_1 \wedge K_2 \wedge u_X \vdash \phi_1$ or $K_1 \wedge K_2 \wedge u_X \vdash \phi_2$ holds.

It follows from our hypotheses that $K_1 \wedge u_X \wedge \neg\phi_1 \vdash \phi_2 \vee \neg K_2 \vee \neg u_X$. Applying Craig's Interpolation Theorem there exists an interpolant δ such that $V(\delta) \subseteq V(K_1 \wedge u_X \wedge \neg\phi_1) \cap V(\phi_2 \vee \neg K_2 \vee \neg u_X)$ and furthermore $K_1 \wedge u_X \wedge \neg\phi_1 \vdash \delta$ and $\delta \vdash \phi_2 \vee \neg K_2 \vee \neg u_X$.

Observing that $V(K_1 \wedge u_X \wedge \neg\phi_1) \cap V(\phi_2 \vee \neg K_2 \vee \neg u_X) \subseteq (Y_1X) \cap (Y_2X) = X$ it follows that $V(\delta) \subseteq X$. As every variable in X appears as a literal in u_X, it follows that either $u_X \vdash \delta$ or $u_X \vdash \neg\delta$. This gives two cases:

1. In the case $u_X \vdash \delta$ recall that $\delta \vdash \phi_2 \vee \neg K_2 \vee \neg u_X$ which means $K_2 \wedge u_X \wedge \delta \vdash \phi_2$ and hence $K_2 \wedge u_X \vdash \phi_2$.
2. In the case $u_X \vdash \neg\delta$ recall that $K_1 \wedge u_X \wedge \neg\phi_1 \vdash \delta$ hence $K_1 \wedge u_X \wedge \neg\phi_1 \vdash \bot$ and thus $K_1 \wedge u_X \vdash \phi_1$.

In either case, we can conclude either $K_1 \wedge K_2 \wedge u_X \vdash \phi_1$ or $K_1 \wedge K_2 \wedge u_X \vdash \phi_2$. With ϕ_1 and ϕ_2 being arbitrary, it follows that Y_1 and Y_2 are independent given X modulo $K_1 \wedge K_2$. □

The previous Theorem can be regarded as a special case of Darwiche's results on *structured databases*, which are graphs similar to Bayesian networks whose vertices are labelled by components of a knowledge base in such a way that conditional independencies may be read directly off the graph itself (Darwiche 1997; Darwiche and Pearl 1994).

5.2 Relationship to Multivalued Dependencies

Our attention now turns to showing that multivalued dependencies for propositional formulae arise as a special case of Darwiche's logical conditional independence.

Theorem 5.3 (Projection Criterion). *Given a propositional formula K and disjoint sets Y_1, Y_2, and X of propositional variables, it follows that Y_1 and Y_2 are independent given X modulo K if and only if $K_{Y_1X} \wedge K_{Y_2X} \vdash K_{Y_1Y_2X}$ holds.*

Proof. Suppose that Y_1 and Y_2 are independent given X modulo K, and consider a world u satisfying both K_{Y_1X} and K_{Y_2X}. We must demonstrate that u satisfies $K_{Y_1Y_2X}$. Assume for the sake of contradiction that u satisfies $\neg K_{Y_1Y_2X}$ as well. Construct the formulae ϕ_1 and ϕ_2 by choosing ϕ_1 as the conjunction of literals over Y_1 satisfied by u, and ϕ_2 as the conjunction of literals over Y_2 satisfied by u. Also choose u_X to be the conjunction of literals over X satisfied by u. It follows that $K \wedge u_X \vdash K_{Y_1Y_2X} \wedge u_X$ and $K_{Y_1Y_2X} \wedge u_X \vdash \neg\phi_1 \vee \neg\phi_2$ hence $K_{Y_1Y_2X} \wedge u_X \vdash \neg\phi_1 \vee \neg\phi_2$, for otherwise there would exist a model of $K_{Y_1Y_2X} \wedge u_X$ equivalent to u on Y_1Y_2X. Being that Y_1 and Y_2 are independent given X modulo K, and u_X is X-complete, it follows that $K \wedge u_X \vdash \neg\phi_1$ or $K \wedge u_X \vdash \neg\phi_2$. However, this means that either $K_{Y_1X} \wedge u_X \vdash \neg\phi_1$ or $K_{Y_2X} \wedge u_X \vdash \neg\phi_2$ which is a contradiction as u satisfies ϕ_1, ϕ_2, u_X, and both projections of K. Therefore, u is a model of $K_{Y_1Y_2X}$ showing that $K_{Y_1X} \wedge K_{Y_2X} \vdash K_{Y_1Y_2X}$ holds.

Conversely, suppose that $K_{Y_1X} \wedge K_{Y_2X} \vdash K_{Y_1Y_2X}$ holds. Consider formulae ϕ_1 and ϕ_2 such that $V(\phi_1) \subseteq Y_1$ and $V(\phi_2) \subseteq Y_2$ along with a world u such that $K \wedge u_X \vdash \phi_1 \vee \phi_2$. We must show that either $K \wedge u_X \vdash \phi_1$ or $K \wedge u_X \vdash \phi_2$. Observe that $K_{Y_1X} \wedge K_{Y_2X} \wedge u_X \vdash u_X \vdash \phi_1 \vee \phi_2$ by the Projection Theorem, and by the Splitting Criterion Y_1 and Y_2 are independent given X modulo $K_{Y_1X} \wedge K_{Y_2X} \wedge u_X$. Thus, either $K_{Y_1X} \wedge K_{Y_2X} \wedge u_X \vdash \phi_1$ or $K_{Y_1X} \wedge K_{Y_2X} \wedge u_X \vdash \phi_2$, hence either $K \wedge u_X \vdash \phi_1$ or $K \wedge u_X \vdash \phi_2$ as required. □

It is worthwhile making two observations: as $K_{Y_1Y_2X} \vdash K_{Y_1X} \wedge K_{Y_2X}$ always holds, so this projection criterion can be rephrased as asserting independence if and only if $K_{Y_1Y_2X} \equiv K_{Y_1X} \wedge K_{Y_2X}$. Furthermore, $K_{Y_1X} \wedge K_{Y_2X}$ is effectively a splitting of $K_{Y_1Y_2X}$ which implies a converse to the Splitting Criterion.

Theorem 5.4. *If X and Y are disjoint subsets of V then a propositional theory K satisfies $X \twoheadrightarrow Y$ if and only if Y and $V - (XY)$ are independent given X modulo K.*

Proof. By the Projection Criterion and our observation it follows that Y and $V - (XY)$ are independent given X if and only if $K \equiv K_{XY} \wedge K_{\overline{Y}}$. Observe that a world $w \in [K_{XY} \wedge K_{\overline{Y}}]$ if and only if there exists $u \in [K_{XY}]$ and $v \in [K_{\overline{Y}}]$ such that $w_{XY} = u_{XY}$ and $w_{\overline{Y}} = v_{\overline{Y}}$. This is in turn equivalent to having K satisfy $X \twoheadrightarrow Y$. □

It is now clear that multivalued dependencies, logical conditional independence, and syntax splitting are different aspects of the same underlying phenomenon. As corollaries of the Splitting Criterion, we see that (PCR) and (CR) ensure that belief revision operators preserve the satisfaction of multivalued dependencies which are partially or fully complied with.

Theorem 5.5. *If $*$ is a belief revision operator which partially complies with $X \twoheadrightarrow Y$, K satisfies $X \twoheadrightarrow Y$, and $V(\phi) \subseteq Y$ then $K * \phi$ satisfies $X \twoheadrightarrow Y$.*

Proof. Observe that when writing $K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$ we have $V((K * \phi)_{XY}) \subseteq XY$ and $V(K_{\overline{Y}}) \subseteq \overline{Y}$ hence the resulting theory satisfies $X \twoheadrightarrow Y$ via the Splitting Criterion. □

Theorem 5.6. *If $*$ is a belief revision operator which fully complies with $X \twoheadrightarrow Y$, K satisfies $X \twoheadrightarrow Y$, and $V(\phi) \subseteq XY$ then $K * \phi$ satisfies $X \twoheadrightarrow Y$.*

Proof. Observe that when writing $K * \phi \equiv (K * \phi)_{XY} \wedge K_{\overline{Y}}$ we have $V((K * \phi)_{XY}) \subseteq XY$ and $V(K_{\overline{Y}}) \subseteq \overline{Y}$ hence the resulting theory satisfies $X \twoheadrightarrow Y$ via the Splitting Criterion. □

6 Discussion

6.1 Sources of Multivalued Dependencies

In our approach we consider multivalued dependencies to be specified by the knowledge engineer as part of the domain knowledge, rather than extracted automatically from the knowledge base. This avoids using possibly-spurious conditional independencies that just happen to hold. As well, we also avoid the cost of determining all potential conditional independencies prior to a revision, given that checking whether a single conditional independence holds is known to be in Π_2^p (Lang, Liberatore, and Marquis 2002).

This raises the question of how a knowledge engineer might determine appropriate multivalued dependencies This question (in the analogous case of conditional irrelevance assertions) is discussed in (Delgrande and Peppas 2018), where a number of sources are suggested: knowledge about the domain (e.g. birds and refrigerators are unrelated), a causal theory, a Bayesian network, or some structural features of a knowledge base which the knowledge engineer deems essential.

In our setting, we can make this a bit more precise. Using the notion of a symbolic causal network introduced by Darwiche and Pearl (1994), it follows from (Darwiche 1997) that conditional independence properties can be read off directly from these networks just as they are for Bayesian networks in probability theory (Pearl 2014). Any multivalued dependency obtained by this method will be non-spurious since it would arise from the causal structure of the domain, as given in the causal network. We believe further investigation of revision operators which comply with the entire structure of a symbolic causal network is worthwhile.

6.2 Related Work

The approach of (Delgrande and Peppas 2018) is closest to our work, which raises the question of whether the independence assertions studied there are related to the conditional independence assertions considered here. Clearly our multivalued dependencies have no mechanism for encoding the selective behaviour of the condition σ in an assertion $\sigma \twoheadrightarrow Z$ unless σ is tautologous, in which case it becomes equivalent to the multivalued dependency $\emptyset \twoheadrightarrow Z$.

In the reverse direction, suppose a multivalued dependency $X \twoheadrightarrow Y$ were encoded via an independence assertion $\sigma \twoheadrightarrow Z$. There are two natural-appearing approaches to consider:

1. If $Z = Y$ then when revising K with $K \vdash \sigma$ by ϕ with $V(\phi) \subseteq Z = Y$ it would follows that $K*\phi \equiv (K*\phi)_Y \wedge K_{\overline{Y}}$. Hence, we would have $K * \phi$ satisfies $\emptyset \twoheadrightarrow Y$. This is far too strong, for this means that all beliefs relating X and Y have been lost in the revision process, whereas we know that (PCR) and (CR) would result in them having been preserved.
2. If $Z = XY$ then when revising K with $K \vdash \sigma$ by ϕ with $V(\phi) \subseteq Z = XY$ it would follow that $K * \phi \equiv (K*\phi)_{XY} \wedge K_{\overline{XY}}$. Hence, we would have $K*\phi$ satisfies $\emptyset \twoheadrightarrow XY$. This is again far too strong, for this means that all beliefs relating X and $\overline{Y}$ have been lost in the revision process, whereas we know that (PCR) and (CR) would result in them having been preserved.

Neither of these are tenable, which suggests that conditional independence assertions cannot in general simulate the multivalued dependencies we consider in this work.

Our results on the relationship between multivalued dependencies and syntax splitting apply as well in the unconditional setting. As an application, the postulates (P1) and (P2) from (Peppas et al. 2015) can be restated as follows:

Theorem 6.1. *Let $*$ be a belief revision operator.*

- *(P1) is equivalent to the following: if K satisfies $\emptyset \twoheadrightarrow Y$ and $V(\phi) \subseteq Y$ then $(K * \phi)_{\overline{Y}} \equiv K_{\overline{Y}}$.*
- *(P2) is equivalent to the following: if K satisfies $\emptyset \twoheadrightarrow Y$ and $V(\phi) \subseteq Y$ then $(K * \phi)_Y \equiv (K_Y * \phi)_Y$.*

6.3 Future Work

There are a number of opportunities for future work deriving from the above. One immediate observation is that although we demonstrate the classes of operators partially complying, or fully complying, with an arbitrary multivalued dependency are non-empty, we have not demonstrated that any reasonable-looking, "natural" belief revision operator reside within these classes. Hence, the question remains of finding interesting belief revision operators which satisfy our postulates.

Another line of inquiry would be to ask how we can take advantage of partial or full compliance to reduce the computational cost of belief revision. One possibility is to develop efficient representations for rankings analogous to Bayesian networks for probability distributions, which use the ranking conditions (CS1), (CS2), and (CS3) to factor a ranking into smaller components.

There are also a number of natural variations on our postulates which seem to merit consideration:

1. Study a "parallelised" variant of our postulates, wherein we consider revising by $\phi \wedge \psi$ with $V(\phi) \subseteq XY$ and $V(\psi) \subseteq \overline{Y}$, with our postulate saying something like $K * (\phi \wedge \psi) = (K * \phi)_{XY} \wedge (K * \psi)_{\overline{Y}}$.
2. Study a "prioritised" variant of our postulates, wherein we consider revising by ϕ with $\phi \wedge K_{\overline{Y}}$ is not necessarily consistent, with our postulate saying something like $K * \phi = K_{\overline{Y}} \circledast (K * \phi)_{XY}$ for some operator $\circledast$.
3. Study belief revision operators which fully comply with all multivalued dependencies simultaneously, and consider the analogues of (P1) and (P2) in this case which would amount to the following:

   CP1. If K satisfies $X \twoheadrightarrow Y$ with $Y \cap X = \emptyset$ and $V(\phi) \subseteq Y$ then $(K * \phi)_{\overline{Y}} \equiv K_{\overline{Y}}$.

   CP2. If K satisfies $X \twoheadrightarrow Y$ with $Y \cap X = \emptyset$ and $V(\phi) \subseteq Y$ then $(K * \phi)_Y \equiv (K_Y * \phi)_Y$.
4. Study postulates which make use of conditional independencies in the sense of Darwiche, which unlike multivalued dependencies need not partition the entire vocabulary.

Finally, it would be interesting to investigate whether these postulates can be extended to nonmonotonic logics in a manner analogous to the extension of Parikh's syntax splitting paradigm in (Kern-Isberner, Beierle, and Brewka 2020).

7 Conclusion

The central challenge of belief revision is to efficiently and plausibly restore consistency to a knowledge base after incorporating a contradictory proposition, and in a manner which causes only minimal changes to existing beliefs. With the standard postulates for belief revision failing to rule out rather pathologically-destructive or bizarre operators, the problem of formalising this requirement of minimality remains an ongoing challenge. We believe that enforcing the requirement that irrelevant beliefs are unchanged is an important aspect of minimal change.

In this work we have extended the previous study of unconditional independence in belief revision to accommodate conditional independence in the form of multivalued dependencies. We have introduced two notions by which a belief revision operator may comply with a multivalued dependency, and characterised these postulates in terms of conditions on faithful rankings. Further, we have endorsed the perspective of (Delgrande and Peppas 2018) that conditional independencies should be provided by the knowledge engineer, rather than read off of the knowledge base. This both avoids enforcing spurious conditional independencies, and means that our operators are not required to carry out the expensive task of checking for conditional independence themselves.

Our hope is that these postulates will assist in identifying those belief revision operators which can be truly said to result in minimal changes to existing beliefs, and that these operators will admit computationally efficient implementations by merit of being able to limit the amount of work required to perform revisions.

Acknowledgements

We would like to thank our reviewers for their insightful comments, as well as the Natural Sciences and Engineering Research Council of Canada for financial support.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*, volume 8. Addison-Wesley Reading.

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 510–530.

Aravanis, T.; Peppas, P.; and Williams, M.-A. 2019. Full characterization of Parikh's relevance-sensitive axiom for belief revision. *Journal of Artificial Intelligence Research* 66:765–792.

Chopra, S., and Parikh, R. 2000. Relevance sensitive belief structures. *Annals of Mathematics and Artificial Intelligence* 28(1):259–285.

Craig, W. 1957. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *The Journal of Symbolic Logic* 22(3):269–285.

Darwiche, A., and Pearl, J. 1994. Symbolic causal networks. In *AAAI*, 238–244.

Darwiche, A. 1997. A logical notion of conditional independence: Properties and applications. *Artificial Intelligence* 97(1-2):45–82.

Delgrande, J., and Peppas, P. 2018. Incorporating relevance in epistemic states in belief revision. In *International Conference on Principles of Knowledge Representation and Reasoning*.

Delgrande, J. P. 2017. A knowledge level account of forgetting. *Journal of Artificial Intelligence Research* 60:1165–1213.

Gardenfors, P. 1990. Belief revision and relevance. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 349–365. Philosophy of Science Association.

Haldimann, J. P.; Kern-Isberner, G.; and Beierle, C. 2020. Syntax splitting for iterated contractions. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, 465–475.

Hodges, W. 1993. *Model Theory*. Cambridge University Press.

Katsuno, H., and Mendelzon, A. O. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52:263–294.

Kern-Isberner, G., and Brewka, G. 2017. Strong syntax splitting for iterated belief revision. In *IJCAI*, 1131–1137.

Kern-Isberner, G.; Beierle, C.; and Brewka, G. 2020. Syntax splitting= relevance+ independence: New postulates for nonmonotonic reasoning from conditional belief bases. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, 560–571.

Lang, J., and Marquis, P. 1998. Complexity results for independence and definability. In *Proc. the 6th International Conference on Knowledge Representation and Reasoning*, 356–367.

Lang, J.; Liberatore, P.; and Marquis, P. 2002. Conditional independence in propositional logic. *Artificial Intelligence* 141(1-2):79–121.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, Language and Computation* 2(96):266–268.

Pearl, J. 2014. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier.

Peppas, P.; Williams, M.-A.; Chopra, S.; and Foo, N. 2015. Relevance in belief revision. *Artificial Intelligence* 229:126–138.

Peppas, P. 2008. Belief revision. *Foundations of Artificial Intelligence* 3:317–359.

An AGM Approach to Revising Preferences

Adrian Haret[1] , **Johannes P. Wallner**[2]
[1]ILLC, University of Amsterdam
[2]TU Graz
a.haret@uva.nl, wallner@ist.tugraz.at

Abstract

We look at preference change arising out of an interaction between two elements: the first is an initial preference ranking encoding a pre-existing attitude; the second element is new preference information signaling input from an authoritative source, which may come into conflict with the initial preference. The aim is to adjust the initial preference and bring it in line with the new preference, without having to give up more information than necessary. We model this process using the formal machinery of belief change, along the lines of the well-known AGM approach. We propose a set of fundamental rationality postulates, and derive the main results of the paper: a set of representation theorems showing that preference change according to these postulates can be rationalized as a choice function guided by a ranking on the comparisons in the initial preference order. We conclude by presenting operators satisfying our proposed postulates. Our approach thus allows us to situate preference revision within the larger family of belief change operators.

1 Introduction

Preferences play a central role in theories of decision making, as part of the mechanism underlying intentional behavior and rational choice, both in economic models of rational agency as well as in formal models of artificial agents supposed to interact with the world and each other (Boutilier et al. 2004; Domshlak et al. 2011; Rossi, Venable, and Walsh 2011; Pigozzi, Tsoukiàs, and Viappiani 2016). Since such interactions take place in dynamic environments, it can be expected that preferences change in response to new developments.

In this paper we are interested in preference change occurring when new preference information, denoted by o, becomes available and has to be taken at face value, thereby prompting a change in prior preference information, denoted by π. The change, we require, should preserve as much useful information from π as can be afforded.

Preference change thus described is a pervasive phenomenon, arising in many contexts spanning the realms of both human and artificial agency. Thus, there is a distinguished tradition in Economics and Philosophy that looks at examples of conflict between an agent's subjective preference (what we call here the initial, or prior preference π) and a second-order preference, often standing for a commitment or moral rule (what we call here the new preference information o): subjective versus 'ethical' preferences (Harsanyi 1955), lack of will, or *akrasia* (Jeffrey 1974), moral commitments (Sen 1977), second-order volitions (Frankfurt 1988) and second-order preferences (Nozick 1994) all fall under this heading.

The same challenge can occur in technological applications, from updating CP-nets (Cadilhac et al. 2015) to changing the order in which search results are displayed on a page in response to user provided specifications. At the same time, similar topics are emerging in the discussion on ethical decision making for artificial agents (Rossi and Mattei 2019) and in issues related to the *alignment problem* (Russell 2019): an artificial agent dealing with humans will have to learn their preferences, but as it cannot do so instantaneously, it must presumably acquire the relevant information in intermediate steps, revising along the way.

Thus, whether it is the internal conflict between an agent's private leanings and the better angels of its nature, or a content provider wanting to tailor its products for a better user experience, many cases of preference change involve a conflict between two types of preferences, one of which is perceived as having priority over the other. However, even though the need to reconcile conflicting preferences in favor of one of them is widely acknowledged, a concrete mechanism for resolving preference conflicts, that works for general preference orders, is often overlooked.

In keeping with prominent approaches to belief change, which model rational change using a plausibility relation over the states of affairs undergoing revision, and echoing a suggestion of Amartya Sen to the effect that conflicts among preferences can be understood using rankings over the preferences themselves (Sen 1977), we propose formalizing preference change using preferences over the basic elements of a preference order, as illustrated in the following example.

Example 1. *The initial preference π is such that, as a result of explicit assertion, item 1 is ranked better than 2 and 2 is ranked better than 3; by virtue of transitivity, it is also inferred that 1 is considered better than 3. We want to revise π by a preference o, according to which 3 is better than 1 (see Figure 1). The simplest solution is to add o to π (i.e., include the comparisons contained in both), but the transitivity requirement leads to a cycle between 1, 2 and 3, which we*

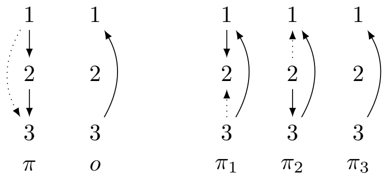


Figure 1: Revising preference order π by o: simply adding o to π leads to a cycle, so if o is accepted then a choice needs to be made regarding which of the initial comparisons of π to keep; potential candidates for the revised order are π_1, π_2 or π_3. A direct comparison ranking i better than j is depicted by a solid arrow from i to j, with comparisons inferred by transitivity depicted by dotted arrows.

would like to avoid. We are thus in a situation where π and o cannot be jointly accepted, but since o, we stipulate, must be accepted, something must be given up from π (though, we ask, no more than strictly necessary). How is the decision to be made? We suggest that an implicit preference over the comparisons of π that were explicitly provided can provide an answer: if the comparison of 1-*vs*-2 *(the edge from* 1 *to* 2 *in Figure 1) is preferred to the one of* 2-*vs*-3 *then the result is π_1, which holds on to* 1-*vs*-2 *from π and together with o infers, by transitivity, that* 3 *is better than* 2*; alternatively, a preference for* 2-*vs*-3 *over* 1-*vs*-2 *leads to π_2 as the result, while indifference between the two comparisons means that both are given up, resulting in π_3. Thus, preference over comparisons in π translates as choice over how to go about revising π. Interestingly, we may also reason in the opposite direction: observing choice behavior across different instances of revision allows us to infer preferences over comparisons in π, e.g., revising to π_1, rather than to π_2 or π_3, can be rationalized as saying that the comparison of* 1-*vs*-2 *is considered better than* 2-*vs*-3*.*

Our purpose here is to formalize the type of reasoning illustrated in Example 1 by rationalizing preference change as a type of choice function on what we will call the *direct comparisons of* π, i.e., the explicit preferences assumed to be given in π. Since a conflict between π and o forces some of the direct comparisons of π to be renounced, additional information in the form of a preference order over the direct comparisons of π will serve as guide to the choice function. The aim, in this, is not legislate on what is the right choice to make; rather, it is to make sure that whatever the choice is, it is made in a coherent way.

Contributions. We present a mechanism for revising a preference order π that is based on an underlying preference relation over the basic, atomic comparisons of π. This mechanism proceeds sequentially, by working its way through the underlying preference relation and adding as many of the direct comparisons of π as possible, while avoiding a conflict with o. We present a set of conditions under which the preference order on direct comparisons of π exists and has desired properties, and characterize the revision mechanism using a set of intuitive normative principles, i.e., rationality postulates in the AGM mould (Alchourrón, Gärdenfors, and Makinson 1985). The significance of our approach lies in laying bare the theoretical requirements and basic assumptions for mechanisms intended to revise preferences.

Related work. Our work complements existing research, but manages to occupy a distinct niche in a broader landscape. Some previous work labeled as preference revision (Bradley 2007; Lang and van der Torre 2008; Liu 2011), looks at changes in preferences prompted by a change in beliefs. Here we abstract away from the source of the new information, choosing to focus exclusively on a mechanism that can be used for resolving conflicts: the rational thing to do when knowing that, for some reason or other, one's preference has to change. Other work (Cadilhac et al. 2015) describes preference change when preferences are represented using CP-nets (Boutilier et al. 2004), or dynamic epistemic logic (Benthem and Liu 2014), in the context of declarative debugging (Dell'Acqua and Pereira 2005), or databases (Chomicki 2003), and therefore comes with additional structural constraints. In contrast, we have opted to represent preferences as strict partial orders over a set of items: we believe this straightforward formulation allows the basic issue signaled by Amartya Sen (Sen 1977), to be visible and tackled head on.

Apart from the issues raised in the Economics literature about second-order desires (Harsanyi 1955; Jeffrey 1974; Sen 1977; Frankfurt 1988; Nozick 1994), the basic phenomenon of preference change has also been raised in explicit connection to belief change (Hansson 1995; Grüne-Yanoff and Hansson 2009b; Grüne-Yanoff 2013), but a representation in terms of preferences on the comparisons present in the preference orders, along the lines suggested here, has, to the best of our knowledge, not yet been given. Much existing work proceeds by putting forward some concrete preference revision mechanism, possibly by shifting some elements of the original preference around, and occasionally with a remark on the similarity between this operation and a belief revision operation (Freund 2004; Chomicki and Song 2005; Liu 2011; Ma, Benferhat, and Liu 2012). What our work adds to these models is an analysis in terms of postulates and representation results.

The postulates we put forward for preference revision bear a distinct resemblance to the AGM postulates employed for belief revision (Alchourrón, Gärdenfors, and Makinson 1985; Katsuno and Mendelzon 1992; Fermé and Hansson 2018): given that changing one's mind involves choosing some parts of a belief to keep and some to remove, this is no coincidence. Indeed, the two problems are similar, though the structural particularities of preferences (in particular, the requirement that they are transitive) mean that transfer of insights from belief revision to preference revision is by no means straightforward.

Outline. The rest of the paper is structured as follows. In Section 2 we introduce notation and the basic elements of our model. In Section 3 we provide a constructive way of revising a preference order, based on rankings of the direct comparisons. In Section 4 we provide a set of intuitive postulates together with their motivations, and discuss their appropriateness for the purpose of modelling preference revision. In Section 5 we identify a set of conditions under which these postulates can be applied. In Section 6 we show that the postulates presented in Section 4 characterize the procedure described in Section 3. Section 7 discusses concrete preference revision operators, and Section 8 offers concluding remarks.

2 Preliminaries

We assume a finite set V of items, standing for the objects an agent can have preferences over. If π is a binary relation on a set V of items, then π is a *strict partial order (spo) on V* if π is transitive and irreflexive, and we write $\mathcal{O}_V$ for the set of strict partial orders on V. If π is an spo on a set V of items, then π is a *strict linear order on V* if π is also total, in addition to being transitive and irreflexive. A *chain on V* is a strict linear order on a subset of V. We write $\mathcal{C}_V$ for the set of chains on V. FInally, π is a total preorder on V if π is transitive and total, with $\mathcal{T}_V$ being the set of total preorders on V. Note that in the following we will typically be interested in total preorders on $V \times V$, i.e., total preorders on the set of comparisons of items in V.

If π is an spo on a set of items V, then a *comparison (i, j) of π* is an element $(i, j) \in \pi$, for some items $i, j \in V$, interpreted as saying that, in the context of π, i is considered strictly better than j. To simplify notation, we sometimes also refer to comparisons with the letter c. We often have to consider the union $\pi_1 \cup \pi_2$ of two spos, which is not guaranteed to be an spo, since transitivity is not preserved under unions. If this is the case, we typically have to substitute $\pi_1 \cup \pi_2$ for its *transitive closure*, denoted by $(\pi_1 \cup \pi_2)^+$. Since preferences are required to be transitive, we write a sequence of comparisons $\{(1, 2), (2, 3) \ldots , (m-1, m)\}^+$ as $(1, \ldots , m)$.

If $\pi = (i_1, \ldots , i_m)$ is a chain on V, a *direct comparison of π* is a comparison $(i_k, i_{k+1}) \in \pi$, i.e., a comparison between i_k and its direct successor in π, with δ_π being the set of direct comparisons of π. The assumption is that direct comparisons are the result of explicit information, and are basic in the sense that they cannot be inferred by transitivity using other comparisons in π. Given preference orders $\pi \in \mathcal{C}_V$ and $o \in \mathcal{O}_V$, we want to carve out the possible options for the revision of π by o. For this we use the set $\lfloor o \rfloor_\pi$ of *π-completions of o*, defined as:

$$\lfloor o \rfloor_\pi = \{(o \cup \delta)^+ \in \mathcal{O}_V \mid \delta \subseteq \delta_\pi\}.$$

The intuition is that a π-completion of o is a preference order constructed from o using some, and only, direct comparisons in π, i.e., information originating exclusively from the two sources given as input. We will expect that a preference revision operator selects one element of this set as the revision result.

Though taking $(\pi \cup o)^+$ as the result of revising π by o is not, in general, feasible, we still want to identify parts of $(\pi \cup o)^+$ that are uncontroversial. To that end, the *cycle-free part α^o_π of $(\pi \cup o)^+$* is defined as:

$$\alpha^o_\pi = \{(i, i{+}1) \in (\pi \cup o)^+ \mid (i{+}1, i) \notin (\pi \cup o)^+\},$$

i.e., the set of comparisons of $(\pi \cup o)^+$ not involved in a cycle with the comparisons of o. The *cyclic part κ^o_π of π with respect to o* is defined as:

$$\kappa^o_\pi = \{(i, i{+}1) \in \delta_\pi \mid (i{+}1, i) \in (\pi \cup o)^+\},$$

i.e., the set of direct comparisons of π involved in a cycle with o.

Example 2. *For π and o as in Example 1, we have that $\delta_\pi = \{(1, 2), (2, 3)\}$, while the π-completions of o are $\lfloor o \rfloor_\pi = \{(3, 1, 2), (2, 3, 1), (3, 1)\}$, i.e., the spos obtained by adding to o either of the elements of δ_π, or none (corresponding to π_1, π_2 and π_3). The cyclic part of π with respect to o is $\kappa^o_\pi = \delta_\pi = \{(1, 2), (2, 3)\}$ and the cycle-free part of π with respect to o is $\alpha^o_\pi = \emptyset$.*

3 A General Method for Revising Preferences

A *preference revision operator* $\triangleright$ is a function $\triangleright\colon \mathcal{C}_V \times \mathcal{O}_V \rightarrow \mathcal{O}_V$ taking a chain π and an spo o as input, and returning an spo $\pi \triangleright o$ as output.

The choice of input and output can be motivated by imagining that π stands for an existing priority ranking, e.g., the ordering of items on a webpage, whereas the new information o is provided by a user and is more likely to be incomplete.

In addition, we may look at this in light of the material that is to come: since we will be rationalizing preference revision operators using preferences (i.e., preorders) on comparisons, an spo as output reflects the fact that certain comparisons are considered equally good, and must be given up together. The unfortunate effect of this is that the input and output formats do not match, which makes it unclear, at this point, whether we can iterate the revision operation. That being said, the output can (and will) be tightened to a chain: provided that the preferences guiding revision are a linear order (i.e., there are no ties). We touch on this aspect at the end of Section 6.

We start, then, by presenting a general procedure for revising preferences that, as advertised, utilizes total preorders on the set δ_π of direct comparisons of π: thus, a *preference assignment a* is a function $a\colon \mathcal{C}_V \rightarrow \mathcal{T}_{V \times V}$ mapping every preference $\pi \in \mathcal{C}_V$ to a total preorder $\leq_\pi$ on elements of $V \times V$, i.e., on pairwise comparisons on the items of V, of which we are interested only in the preorder on δ_π. In typical AGM manner, a comparison $c_i \leq_\pi c_j$ in the context of a preorder $\leq_\pi$ on δ_π means that c_i is *better* than c_j.

If $\pi \in \mathcal{C}_V$, $o \in \mathcal{O}_V$ and $\leq_\pi$ is a total preorder on δ_π, then, for $i \geq 1$, the *$\leq_\pi$-level i of δ_π*, denoted $lvl^i_\leq(\delta_\pi)$, contains the i^{th} best elements of δ_π according to $\leq_\pi$, i.e., $lvl^1_{\leq_\pi}(\delta_\pi) = \min_{\leq_\pi}(\delta_\pi)$, $lvl^{i+1}_{\leq_\pi}(\delta_\pi) = \min_{\leq_\pi}(\delta_\pi \setminus \bigcup_{1 \leq j \leq i} lvl^j_{\leq_\pi}(\delta_\pi))$, etc. Note that the $\leq_\pi$-levels of δ_π partition δ_π and, since δ_π is finite, there exists a $j > 0$ such that $lvl^i_{\leq_\pi}(\delta_\pi) = \emptyset$, for all $i \geq j$. The *addition operator* $\text{add}^i_{\leq_\pi}(o)$ is defined, for any $o \in \mathcal{O}_V$ and $i \geq 0$, as follows:

$$\text{add}^0_{\leq_\pi}(o) = (o \cup \alpha^o_\pi)^+,$$

$$\text{add}^i_{\leq_\pi}(o) = \begin{cases} (\text{add}^{i-1}_{\leq_\pi}(o) \cup (lvl^i_{\leq_\pi}(\delta_\pi) \cap \kappa^o_\pi))^+, & \text{if in } \mathcal{O}_V, \\ \text{add}^{i-1}_{\leq_\pi}(o), & \text{otherwise.} \end{cases}$$

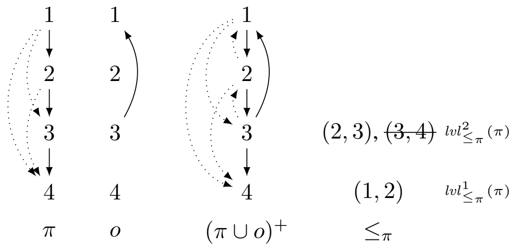


Figure 2: Preference revision by adding direct comparisons from π to o, using the preorder $\leq_\pi$. In $\leq_\pi$ lower means better; the comparison $(3, 4)$ is ignored by the addition operator because it is not involved in a cycle with o (and is added at the beginning anyway).

Intuitively, the addition operator starts by adding to o all the direct comparisons of π that are not involved in a cycle with it, i.e., which are not under contention by the accrual of new preference information. Then, at every further step $i > 0$, the addition operator tries to add all comparisons on level i of δ_π that are involved in a cycle with o: if the resulting set of comparisons can be construed as a spo (by taking its transitive closure) the operation is successful, and the new comparisons are added; if not, the addition operator does nothing. Since the addition of new comparison follows the order $\leq_\pi$, this ensures that better quality comparisons are considered before lower quality ones.

Note that this procedure guarantees that there are always *some* comparisons in $\pi \triangleright o$, i.e., we have that $o \subseteq \pi \triangleright o$, regardless of anything else. Note, also, that the number of non-empty levels in δ_π is finite and the addition operation eventually reaches a fixed point, i.e., there exists $j \geq 0$ such that $\mathrm{add}^i_{\leq_\pi}(o) = \mathrm{add}^j_{\leq_\pi}(o)$, for any $i \geq j$. We denote by $\mathrm{add}^*_{\leq_\pi}(o)$ the fixed point of this operator and take it as the defining expression of a preference revision operator: if a is a preference assignment, then the *a-induced preference revision operator* $\triangleright^a$ is defined, for any $\pi \in \mathcal{C}_V$ and $o \in \mathcal{O}_V$, as:

$$\pi \triangleright^a o = \mathrm{add}^*_{\leq_\pi}(o).$$

Note that, by design, $\mathrm{add}^*_{\leq_\pi}(o) \in \mathcal{O}_V$, i.e., the operator $\triangleright$ is well defined.

Example 3. *Consider initial preference order* $\pi = (1, 2, 3, 4)$ *and new information* $o = (3, 1)$. *We obtain that the direct comparisons of* π *are* $\delta_\pi = \{(1, 2), (2, 3), (3, 4)\}$. *Suppose, now, that there is a total preorder* $\leq_\pi$ *on* δ_π *according to which* $(1, 2) <_\pi (2, 3) \approx_\pi (3, 4)$, *as depicted in Figure 2. To construct* $\pi \triangleright o$, *the addition operator starts from* $\mathrm{add}^0_{\leq_\pi}(o) = (\{(3, 1)\} \cup \{(1, 4), (2, 4), (3, 4)\})^+$, *i.e., o itself together with* α^o_π, *the cycle-free part of* π *with respect to o. At the next step the addition operator tries to add* $(1, 2)$, *which it can do successfully; at the next step it attempts to add* $(2, 3)$, *which creates a conflict with* $(3, 1)$ *and* $(1, 2)$, *added previously. After this there are no more comparisons to add.*

4 Postulates for Preference Revision

We show now that the procedure described in Section 3 can be characterized with a set of AGM-like postulates that do not reference any concrete revision procedure and are, by themselves, intuitive enough to provide reasonable constraints on any preference revision operator.

The first two postulates we consider apply to any chain $\pi \in \mathcal{C}_V$, spo $o \in \mathcal{O}_V$ and preference revision operator $\triangleright \colon \mathcal{C}_V \times \mathcal{O}_V \to \mathcal{O}_V$, and are as follows:

(P_1) $\pi \triangleright o \in \lfloor o \rfloor_\pi$.

(P_2) $\alpha^o_\pi \subseteq \pi \triangleright o$.

Postulates P_{1-2} require the result to be formed by adding elements from π to the new information o, and to be of a certain admissible type, i.e., an spo. They are meant to capture preference revision in its most uncontroversial aspects, yet they still require some careful unpacking.

Postulate P_1 states that $\pi \triangleright o$ is a π-completion of o, i.e., a preference order constructed only by adding direct comparisons from π to o. Unfolding its consequences, postulate P_1 ensures that:

(i) $\pi \triangleright o \in \mathcal{O}_V$, i.e., $\pi \triangleright o$ is a chain on V,

(ii) $o \subseteq \pi \triangleright o$, i.e., $\pi \triangleright o$ contains all the information present in o, and

(iii) $\pi \triangleright o \subseteq (\pi \cup o)^+$, i.e., $\pi \triangleright o$ is contained in the binary relation obtained by simply adding o to π, and adding all the comparisons inferred by transitivity.

In terms of AGM propositional belief revision, postulate P_1 does the same duty as the *Closure*, *Success*, *Inclusion* and *Consistency* postulates (Hansson 2017; Fermé and Hansson 2018). These postulate mandate that the revision result should be a propositional theory (i.e., have a required format), that the new information should be accepted, and that, unless the new information is inconsistent, the revision result should be consistent.

Given this observation, a question emerges as to why not use conditions (i)-(iii) as postulates instead of the proposed P_1. The reason is that P_1 contains an element that lacks from conditions (i)-(iii): what P_1 adds is the requirement that $\pi \triangleright o$ is to be constructed using only direct comparisons of π (in addition to o), and the reason why such a condition is desirable is to prevent $\pi \triangleright o$ from having opinions on items over which no opinion had been expressed before revision. The issue is illustrated in Example 4.

Example 4. *Consider preferences* π *and o as in Example 1, and an additional spo* $\pi_4 = \{(3, 1), (3, 2)\}$. *Note that* π_4 *is such that* $o \subseteq \pi_4 \subseteq (\pi \cup o)^+$ *and therefore satisfies conditions (i)-(iii) expressed above, so that according to conditions (i)-(iii) preference* π_4 *is a viable revision result.*

At the same time, we do not want to consider π_4 *as a potential candidate for the revision result: the comparison* $(3, 2)$ *occurs neither in* π *nor in o as a direct comparison, and there is reason to think that adding it would be unjustified: a rational preference revision operator should not be allowed to return* π_4 *when revising* π *by o. By contrast, when* *the comparison* $(3,2)$ *does occur, e.g., in the desirable preference order* $\pi_1 = (3,1,2)$*, it occurs as the result of inference from* $(3,1)$*, which is added from o, and* $(1,2)$*, which is preserved from* π.

Postulate P_2 says that the cycle-free part of π with respect to o is to be preserved in $\pi \triangleright o$, and is meant to preserve the parts of $(\pi \cup o)^+$ that are not up for dispute. Note that in the case when $(\pi \cup o)^+$ does not contain a cycle then $\alpha^o_\pi = (\pi \cup o)^+$, and P_2 together with P_1 imply that $\pi \triangleright o = (\pi \cup o)^+$: this is the case when revision is easy, and nothing special needs to be done. Throughout all this, postulate P_2 serves the same function as the *Vacuity* postulate in propositional revision (Hansson 2017; Fermé and Hansson 2018): in the ideal case, when o can simply be added to π, applying postulate P_2 results in the union of the two structures.

So far we have established that, if there is no conflict between π and o, i.e., no cycle arises by adding o to π, then we can simply add o to π; and if there is a conflict, then $\triangleright$ must choose between the direct comparisons of π involved in the cycle. This choice, however, must be coherent in a precise sense: we expect the choices to be indicative of an underlying preference over direct comparisons, which remains stable across different instances of revision. This sense of coherence is illustrated by Example 5.

Example 5. *Consider revising* $\pi = (1,2,3,4)$*, depicted in Figure 2, by* $o_1 = (4,1)$*. SInce adding* $(\pi \cup o)^+$ *contains a cycle, revision requires a choice between comparisons* $(1,2)$*,* $(2,3)$ *and* $(3,4)$*: assume* $(1,2)$ *is chosen, suggesting* $(1,2)$ *is better than* $(2,3)$ *and* $(3,4)$*. Suppose, now, that we add* $o_2 = \{(3,4)\}$ *and revise by* $(o_1 \cup o_2)^+ = \{(3,4),(4,1),(3,1)\}$*: another cycle is formed, and a choice is necessary, this time only between* $(1,2)$ *and* $(2,3)$*. In accordance with the previous decision,* $(1,2)$ *should be chosen here as well.*

The choice behavior of a revision operator has to reflect an implicit preference order over the direct comparisons of π, and this is handled by the following postulates, meant to apply to any chain $\pi \in \mathcal{C}_V$, spos $o_1, o_2 \in \mathcal{O}_V$ such that $(o_1 \cup o_2)^+ \in \mathcal{O}_V$, and a preference revision operator $\triangleright$:

(P_3) $\pi \triangleright (o_1 \cup o_2)^+ \subseteq ((\pi \triangleright o_1) \cup o_2)^+$.

(P_4) If $((\pi \triangleright o_1) \cup o_2)^+ \in \mathcal{O}_V$, then $((\pi \triangleright o_1) \cup o_2)^+ \subseteq \pi \triangleright (o_1 \cup o_2)^+$.

There is a similarity between postulates P_3 and P_4 and the *Superexpansion* and *Subexpansion* postulates, respectively, from propositional belief revision (Hansson 2017; Fermé and Hansson 2018), which ensure that the choice between two options is stable and independent of alternatives not directly involved. Postulates P_{3-4} are meant to ensure the same here. However, it turns out that in the context of preference revision this happens only under a specific set of conditions, which we elaborate on in the following section.

5 Coordination

In this section we identify the precise conditions under which it makes sense to apply postulates P_{3-4}, presented in Section 4. Before doing so, we introduce some additional notation.

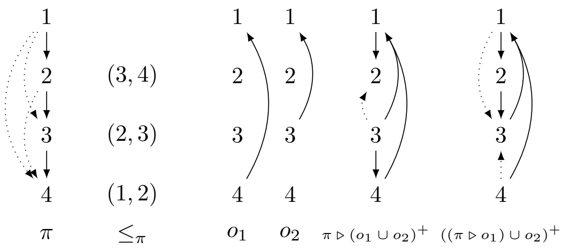


Figure 3: Postulates P_{3-4} are satisfied only if o_1 and o_2 are coordinated with respect to π.

If o_1 and o_2 are spos, we say that o_1 and o_2 are *coordinated with respect to* π if for any set $\delta \subseteq \kappa^{o_1}_\pi$ such that for every direct comparison $(i, i{+}1) \in \delta$, neither $(i, i{+}1)$ nor $(i{+}1, i)$ is in $(o_1 \cup o_2)^+$, it holds that if $(o_1 \cup \delta)^+ \in \mathcal{O}_V$, then $((o_1 \cup o_2)^+ \cup \delta)^+ \in \mathcal{O}_V$. In other words, if π and o_1 form a cycle and we want to add o_2 as well, then we direct our attention to the direct comparisons in π that are not directly ruled out by $(o_1 \cup o_2)^+$, i.e., such that neither these comparisons nor their inverses are contained in $(o_1 \cup o_2)^+$. The property of coordination says that if we can consistently add some of these comparisons to o_1, then it must be the case that we can also add them to $(o_1 \cup o_2)^+$. Intuitively, coordination means that adding extra information o_2 does not step on o_1's toes, by rendering unviable any comparisons that were previously viable. The following example makes this clearer.

Example 6. *Take* $\pi = (1,2,3,4)$ *and* $o_1 = (4,1)$*,* $o_2 = (3,1)$*. The direct comparisons of* π *that are involved in a cycle with* o_1 *are* $\kappa^{o_1}_\pi = \{(1,2),(2,3),(3,4)\}$*, so that revision by* o_1 *requires making a choice between these comparisons. This choice, we expect, is done on the basis of some implicit preference over the comparisons, which guides revision even when we add additional information in the form of* o_2*. Notice, now, that neither of* $(1,2)$*,* $(2,3)$ *and* $(3,4)$ *is individually ruled out by* $(o_1 \cup o_2)^+$*: we have, for instance, that* $(1,2) \notin (o_1 \cup o_2)^+$ *and* $(2,1) \notin (o_1 \cup o_2)^+$*; the same holds for* $(2,3)$ *and* $(3,4)$*. The significance of this is that adding* o_2 *to* o_1 *does not alter the menu: the choice is still one over comparisons* $(1,2)$*,* $(2,3)$ *and* $(3,4)$*.*

The problem, however, is that whereas with o_1 *the choice is relatively unconstrained, meaning we can choose any proper subset of* $\{(1,2),(2,3),(3,4)\}$ *to add to* $(4,1)$*, adding the additional comparison* $(3,1)$ *complicates things. To see how, consider the set of comparisons* $\delta = \{(1,2),(2,3)\}$*. These comparisons can be consistently added to* o_1*, i.e.,* $(o_1 \cup \delta)^+ \in \mathcal{O}_V$*, but not to* $(o_1 \cup o_2)^+$*, i.e.,* $((o_1 \cup o_2)^+ \cup \delta)^+ \notin \mathcal{O}_V$*. According to our definition, this implies that* o_1 *and* o_2 *are not coordinated with respect to* π*. Thus, whereas with* o_1 *can be augmented with both* $(1,2)$ *and* $(2,3)$*,* o_1 *and* o_2 *do not allow adding both comparisons* together. *This, then, has a knock-down effect in that it makes it possible to add comparison* $(3,4)$*, irrespective of where it is in the preorder on comparisons.*

In such a situation, then, the specific details of how the choice problem is constructed makes the position of $(3,4)$ *in the overall preference order over comparisons irrelevant. Consequently, expecting our axioms to take the preference order into account will land us into trouble. To see this, onsider preorder* $\leq_\pi$ *in Figure 3, where* $(3,4)$ *is the least preferred comparison, and the revision operator* $\triangleright$ *induced by it. We have that* $(3,4) \in \pi \triangleright (o_1 \cup o_2)^+$*, but* $(3,4) \notin ((\pi \triangleright o_1) \cup o_2)^+$*, i.e., postulate* P_3 *is not satisfied.*

This fact is related with the lack of coordination between o_1 *and* o_2*, as the addition of* o_2 *tampers with the choice problem: though we can still add either one of the three comparisons, as mentioned above, we cannot add* $(1,2)$ *and* $(2,3)$ *together anymore, which in turn means that* $(3,4)$ *can be added regardless of its position in* $\leq_\pi$ *the preorder.*

Example 6 is a case in which lack of coordination creates a situation where postulate P_3 is not satisfied. We do not mean to imply, however, that there is anything wrong with postulate P_3, or with uncoordinated preference information. Rather, we take the moral to be that we need postulates tailored to cases that do *not* look like the one in Example 6, in which preference information over the direct comparisons is rendered unusable by the overriding structural constraints of working with preference orders.

In other words, we want the behavior of a revision operator to reflect the preference information over the direct comparisons: however, the requirement of transitivity means that, in the interest of consistency, we sometimes have to add comparisons that were not explicitly chosen, and this can interfere with the preference information over the comparisons of π. Thus, the significance of coordination, as the following theorem shows, is that it is needed in order for postulates P_{3-4} to be effective at ensuring that choice across different types of incoming preferences is coherent.

Theorem 1. *If* $a\colon \mathcal{C}_V \to \mathcal{T}_{V\times V}$ *is a preference assignment and* $\triangleright^a$ *is the* a*-induced revision operator, then,* $\triangleright^a$ *satisfies postulates* P_{3-4} *if and only if, for any chain* $\pi \in \mathcal{C}_V$ *and spos* $o_1, o_2 \in \mathcal{O}_V$*, it holds that* o_1 *and* o_2 *are coordinated with respect to* π*.*

Proof. ("$\Leftarrow$") Take $o_1, o_2 \in \mathcal{O}_V$ that are coordinated with respect to π. We will show that, for any preorder $\leq_\pi$ on δ_π, the a-induced revision operator $\triangleright^a$ satisfies postulates P_{3-4}. Since $\triangleright^a$ satisfies postulates P_{3-4} trivially if $(\pi \cup o_1)^+ \in \mathcal{O}_V$, we look at the case when $\kappa_\pi^{o_1} \neq \emptyset$, i.e., when $(\pi \cup o_1)^+$ contains a cycle.

For postulate P_3, assume there is a comparison $c^\star \in \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$ such that $c^\star \notin (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$. If $c^\star \in (o_1 \cup o_2)^+$ then a contradiction follows immediately. We thus have to look at the case when $c^\star \notin (o_1 \cup o_2)^+$, which contains two subcases of its own.

Case 1. If $c^\star \in \delta_\pi$, then by our assumption we have that $c^\star \in \kappa_\pi^{o_1}$, i.e., $c^\star$ is involved in some cycle with o_1. From $c^\star \notin \mathrm{add}^*_{\leq_\pi}(o_1)$ we infer that there must be a set $\delta \subseteq \delta_\pi$ of direct comparisons of π that precede $c^\star$ in $\leq_\pi$, are added to o_1 before it, and prevent $c^\star$ itself from being added. In particular, this means that $(o_1 \cup \delta)^+ \in \mathcal{O}_V$, but $((o_1 \cup \delta)^+ \cup \{c^\star\})^+ \notin \mathcal{O}_V$. At the same time, we know that $c^\star \in \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$, i.e., $c^\star$ can be consistently added to $(o_1 \cup o_2)^+$. Note that this happens after all the comparisons in δ, which precede it in $\leq_\pi$, have been considered as well. This implies that not all of the comparisons in δ can be added to $(o_1 \cup o_2)^+$, since if they could, then the cycle formed with o_1, δ and $c^\star$ would be reproduced here as well. If not all of the comparisons in δ can be added to $(o_1 \cup o_2)^+$, this must be because $((o_1 \cup o_2)^+ \cup \delta)^+$ contains a cycle, i.e., $((o_1 \cup o_2)^+ \cup \delta)^+ \notin \mathcal{O}_V$. This now contradicts the fact that o_1 and o_2 are coordinated with respect to π.

Case 2. If $c^\star$ is not a direct comparison of π, then it is inferred by transitivity using at least one direct comparison of π added previously. We apply the reasoning in Case 1 to these direct comparisons to show that they are in $(\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$, which implies that $c^\star \in (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$ as well.

For postulate P_4, take $c^\star \in (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$ and assume $c^\star \notin \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$. As before, the non-obvious case is when $c^\star \notin (o_1 \cup o_2)^+$. If $c^\star \in \delta_\pi$, then from the assumption that $c^\star \notin \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$ we conclude that there is a set $\delta \subseteq \kappa_\pi^{o_1}$ of comparisons that precede $c^\star$ in $\leq_\pi$, are added to $(o_1 \cup o_2)^+$ before it and, in concert with $(o_1 \cup o_2)^+$, block $c^\star$ from being added, i.e., such that:

$$((o_1 \cup o_2)^+ \cup \delta)^+ \in \mathcal{O}_V,$$

but $((o_1 \cup o_2)^+ \cup \delta')^+ \notin \mathcal{O}_V$, where $\delta' = \delta \cup \{c_\star\}$. From the second to last result we infer that δ can be added consistently to $(o_1 \cup o_2)^+$ and, since we have that $c^\star \in (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$ as well, we obtain that and $c^\star$ can be added consistently to o_1. In other words, it holds that $(o_1 \cup \delta')^+ \in \mathcal{O}_V$. Together with the previous result this contradicts the fact that o_1 and o_2 are coordinated with respect to π.

The case when $c^\star \notin (o_1 \cup o_2)^+$ is treated analogously as for postulate P_3.

("$\Rightarrow$") Assume that there are $o_1, o_2 \in \mathcal{O}_V$ not coordinated with respect to π, i.e., there exists a set $\delta \subseteq \kappa_\pi^{o_1}$ of direct comparisons of π that are involved in a cycle with o_1 and are such that $(o_1 \cup \delta)^+ \in \mathcal{O}_V$ and $((o_1 \cup o_2)^+ \cup \delta)^+ \notin \mathcal{O}_V$. Additionally, we have that neither of the comparisons in δ, or their inverses, are in $(o_1 \cup o_2)^+$. We infer that there must exist a comparison $c^\star \in (\kappa_\pi^{o_1} \setminus \delta)$ that completes the cycle. We will show that there exists a preorder $\leq_\pi$ such that the revision operator induced by it does not satisfy P_3. Take a preorder $\leq_\pi$ on δ_π that arranges the elements of δ in a linear order at the bottom of $\leq_\pi$, i.e., such that $c_j <_\pi c_l$, for any $c_j \in \delta$ and $c_l \notin \delta$, and $c^\star$ the maximal element in $\leq_\pi$, i.e., $c_j <_\pi c^\star$, for any $c_j \in \delta$. This implies, in particular, that $c_j <_\pi c^\star$, for any $c_j \in \delta$.

Note, now, that $c^\star \in \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$: this is because, by assumption, not all of the comparisons in δ can be added to $(o_1 \cup o_2)^+$, and this makes it possible for $c^\star$ to be added. On the other hand, $c^\star \notin (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$: this is because here we can, again by assumption, consistently add δ to o_1 and, since $c^\star$ is the last in line to be added, the inevitability of creating a cycle with δ and the rest of the comparisons of o_1 makes it impossible to do so consistently. We obtain that $c^\star \in \mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$ but $c^\star \notin (\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$, i.e., postulate P_3 is not satisfied. Concurrently, there will be a

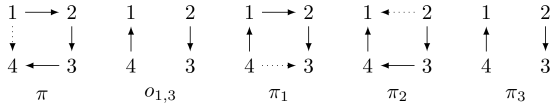


Figure 4: Revision of π by $o_{1,3}$ forces a choice between direct comparisons $(1,2)$ and $(3,4)$: since keeping both $(1,2)$ and $(3,4)$ is not possible, at least one of them, potentially both, must be discarded. Depending on the choice made, possible results are π_1, π_2 and π_3.

comparison in δ that occurs in $(\mathrm{add}^*_{\leq_\pi}(o_1) \cup o_2)^+$ that does not make it into $\mathrm{add}^*_{\leq_\pi}(o_1 \cup o_2)^+$, showing that P_4 is not satisfied either. □

Theorem 1 shows that coordination is needed in order to make sure that postulates P_{3-4} work, and we will henceforth assume that o_1 and o_2 are coordinated with respect to π whenever we apply these postulates.

6 Characterizing Preference Revision as Choice Over Comparisons

We show now that the procedure described in Section 3 is characterized by the postulates introduced in Section 4, under the restrictions established through Theorem 1. Theorem 2 shows that the procedure in Section 3 yields preference revision operators that satisfy postulates P_{1-4}.

Theorem 2. *If $a\colon \mathcal{C}_V \to \mathcal{T}_{V\times V}$ is a preference assignment, then the revision operator $\triangleright^a$ induced by it satisfies postulates P_{1-4}, for any $\pi \in \mathcal{C}_V$ and $o, o_1, o_2 \in \mathcal{O}_V$ such that o_1, o_2 are coordinated with respect to π.*

Proof. Satisfaction of postulates P_{1-2} is straightforward. For P_1, since at every step $\mathrm{add}^i_{\leq_\pi}$ selects some direct comparisons in π to add to o, the end result satisfies the condition for being in $\lfloor o \rceil_\pi$. For P_2, note that $(\pi \cup o)^+ \subseteq \mathrm{add}^0_{\leq_\pi}(o) \subseteq \mathrm{add}^*_{\leq_\pi}(o)$. Since o_1 and o_2 are assumed to be coordinated with respect to π, satisfaction of postulates P_{3-4} is guaranteed by Theorem 1. □

For the converse, we want to show that any preference revision operator satisfying P_{1-4} can be rationalized using a preference assignment.

To that end, we will construct the preorder $\leq_\pi$ from binary comparisons, but we must first figure out how to compare two direct comparisons $(k, k{+}1)$ and $(l, l{+}1)$. This is done by creating a situation where we cannot add both and hence one has to be given up. We will use a special type of preference order to induce a choice between these comparisons. If $\pi \in \mathcal{C}_V$ is a chain and $(k, k{+}1), (l, l{+}1) \in \delta_\pi$ are direct comparisons of π, the *choice inducing preference $o_{k,l}$ for $(k, k{+}1)$ and $(l, l{+}1)$* is defined as $o_{k,l} = \{(k{+}1, l), (l{+}1, k)\}$. The following example illustrates this notion.

Example 7. *To induce a choice between direct comparisons $(1,2)$ and $(3,4)$ in Figure 4, revise by $o_{1,3} = \{(2,3),(4,1)\}$. Note that effectiveness of this maneuver hinges on the choice being confined to the direct comparisons of π: if inferred comparisons were allowed to be part of the choice, $o_{1,3}$ loses its power to discriminate between $(1,2)$ and $(3,4)$: if, for instance, $(1,3)$ and $(2,4)$ are chosen, then $(2,1)$ and $(4,3)$ have to be inferred, leaving no space for a choice between $(1,2)$ and $(3,4)$, i.e., $o_{1,3}$ would tell us nothing about the implicit preference between $(1,2)$ and $(3,4)$. We can also see that comparison of $(1,2)$ and $(2,3)$ is done by revising by $(3,1)$.*

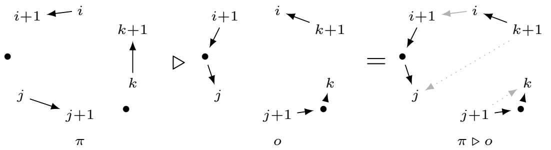


Figure 5: To show that $\leq^{\triangleright}_\pi$ is transitive, we show first that $(k, k{+}1) \notin \pi \triangleright o$. Bullets indicate other potential items in π; faded arrows indicate comparisons that may not be in $\pi \triangleright o$, but can be consistently added to it.

Conversely, if $(k, k{+}1), (l, l{+}1) \in \delta_\pi$ and $\triangleright$ is a preference revision operator, then the *revealed order $\leq^{\triangleright}_\pi$ between $(k, k{+}1)$ and $(l, l{+}1)$* is defined as:

$$(k, k{+}1) \leq^{\triangleright}_\pi (l, l{+}1) \text{ if } (l, l{+}1) \notin \pi \triangleright o_{k,l}.$$

Intuitively, $(l, l{+}1)$ being discarded from $\pi \triangleright o_{k,l}$ signals that it is considered less important than $(k, k{+}1)$.

The primary question at this point is whether the revealed preference relation $\leq^{\triangleright}_\pi$, as defined above, is transitive. We show next that the answer is yes.

Lemma 1. *If $\triangleright$ satisfies postulates P_{1-4}, then the revealed preference relation $\leq^{\triangleright}_\pi$ is transitive.*

Proof. Take $\pi \in \mathcal{C}_V$ and $(i, i{+}1), (j, j{+}1), (k, k{+}1) \in \delta_\pi$ such that $(i, i{+}1) \leq^{\triangleright}_\pi (j, j{+}1) \leq^{\triangleright}_\pi (k, k{+}1)$ (we can assume that $i < j < k$). To show that $(i, i{+}1) \leq^{\triangleright}_\pi (j, j{+}1)$, take $o \in \mathcal{O}_V$ that contains all direct comparisons in π up to k, except $(i, i{+}1)$, $(j, j{+}1)$ and $(k, k{+}1)$, plus the comparison $(k{+}1, i)$. In other words, o is such that if $(i, i{+}1)$, $(j, j{+}1)$ and $(k, k{+}1)$ were added to it, a cycle would form. The first step involves showing that $(k, k{+}1) \notin \pi \triangleright o$. To see why this is the case, note first that, by design, not all of $(i, i{+}1)$, $(j, j{+}1)$ and $(k, k{+}1)$ can be in $\pi \triangleright o$, i.e., at least one of them must be left out. We now do a case analysis to show that, either way, $(k, k{+}1)$ ends up being left out.

Case 1. If $(k, k{+}1) \notin \pi \triangleright o$, the conclusion is immediate.

Case 2. If $(j, j{+}1) \notin \pi \triangleright o$, then we can safely add $(i, i{+}1)$ to $\pi \triangleright o$: this is because the inference of the opposite comparison, i.e., $(i{+}1, i)$, can be done only by adding all comparisons on the path from $i{+}1$ to i, and the absence of $(j, j{+}1)$ means this inference is blocked. Using P_{3-4} we can now conclude that $((\pi \triangleright o) \cup \{(i, i{+}1)\})^+ = \pi \triangleright (o \cup \{(i, i{+}1)\})^+$ (see Figure 5). Note, we can separate $o \cup \{(i, i{+}1)\}$ into $o_{j,k} = \{(k{+}1, j), (j{+}1, k)\}$ and all the comparisons on the path from $k{+}1$ to j, plus the comparisons on the path from $j{+}1$ to k. Call this latter preference

o'. We thus have that $(o \cup \{(i, i{+}1)\})^+ = (o_{j,k} \cup o')^+$ and, applying P_3, we obtain that:

$$\pi \rhd (o \cup \{(i, i{+}1)\})^+ = \pi \rhd (o_{j,k} \cup o')^+ \subseteq ((\pi \rhd o_{j,k}) \cup o')^+.$$

Since, by definition, $(k, k{+}1) \notin \pi \rhd o_{j,k}$ and $(k, k{+}1) \notin o'$, It follows that:

$$(k, k{+}1) \notin \pi \rhd (o \cup \{(i, i{+}1)\})^+,$$

then:

$$(k, k{+}1) \notin ((\pi \rhd o) \cup \{(i, i{+}1)\})^+,$$

and, finally, that:

$$(k, k{+}1) \notin \pi \rhd o.$$

Case 3. If $(i, i{+}1) \notin \pi \rhd o$, then we can safely add $(k, k{+}1)$ to $\pi \rhd o$ and, by reasoning similar to above, show that $(j, j{+}1) \notin \pi \rhd o$. Here we invoke Case 2.

With the fact that $(k, k{+}1) \notin \pi \rhd o$ in hand, we can add $(j, j{+}1)$ to $\pi \rhd o$ (by reasoning similar to above), because the path from $j{+}1$ to j in $\pi \rhd o$ is blocked by the absence of $(k, k{+}1)$. Using postulates P_{3-4}, we conclude that:

$$\begin{aligned}((\pi \rhd o) \cup \{(j, j{+}1)\})^+ &= \pi \rhd (o \cup \{(j, j{+}1)\})^+\\ &= \pi \rhd (\{(i{+}1, \ldots, k), (k{+}1, i)\})^+\\ &= ((\pi \rhd o_{i,k}) \cup \{(i{+}1, \ldots, k)\})^+.\end{aligned}$$

Since $(k, k{+}1) \notin ((\pi \rhd o) \cup \{(j, j{+}1)\})^+$, we conclude that $(k, k{+}1) \notin \pi \rhd o_{i,k}$, which implies that $(i, i{+}1) \leq^{\rhd}_{\pi} (k, k{+}1)$. □

Lemma 1 is crucial for the following representation result, as it indicates that we can identify the revealed preference relation with the underlying preference over direct comparisons of π driving revision.

Theorem 3. *If $\rhd$ is a revision operator satisfying postulates P_{1-4}, for any $\pi \in \mathcal{C}_V$ and $o, o_1, o_2 \in \mathcal{O}_V$ such that o_1, o_2 are coordinated with respect to π, then there exists a preference assignment a such that $\rhd$ is the a-induced revision operator.*

Proof. For any $\pi \in \mathcal{C}_V$, take $\leq_\pi$ to be the revealed preference relation $\leq^{\rhd}_{\pi}$. By Lemma 1, we know that $\leq_\pi$ is transitive, so the only thing left to is show is that $\pi \rhd o = \mathrm{add}^*_{\leq_\pi}(o)$. We do this in two steps.

("$\subseteq$") For one direction, Take $(j, k) \in \pi \rhd o$ and suppose $(j, k) \notin \mathrm{add}^*_{\leq_\pi}(o)$. Clearly, it cannot be the case that $(j, k) \in o$, so we conclude that (j, k) is either a direct comparison of π, or is inferred by transitivity using direct comparisons in π and o.

Case 1. If $(j, k) \in \delta_\pi$, then we can write (j, k) as $(j, j{+}1)$, Suppose that $(j, j{+}1)$ is on level i of δ_π: this means that if $(j, j{+}1)$ does not get added to $\mathrm{add}^*_{\leq_\pi}(o)$ at step i, then, since it cannot be inferred by transitivity, it does not get added at all. The fact that $(j, j{+}1) \notin \mathrm{add}^*_{\leq_\pi}(o)$ thus means that $(j, j{+}1)$ forms a cycle with some comparisons in o and comparisons in π on levels $l \leq i$. First, note that $(j, j{+}1)$ cannot form a cycle with elements of o only, since that would imply that $(j{+}1, j) \in o$ and that would exclude the possibility that $(j, j{+}1) \in \pi \rhd o$. Thus, at least one other comparison in the cycle must come from π. We can state, now, that, since $(j{+}1, j) \in \pi \rhd o$, then at least one of these comparisons must be absent in $\pi \rhd o$, i.e., there exists a direct comparison $(k, k{+}1) \in \delta_\pi$ such that $(k, k{+}1) \in \mathit{lvl}^{j}_{\leq_\pi}(\pi)$, for some $j \leq i$, $(k, k{+}1) \notin \pi \rhd o$ and $(j, j{+}1)$, $(k, k{+}1)$, plus some other comparisons in o and π form a cycle. This means that it is safe to add o' to $\pi \rhd o$, where o' contains all comparisons on the path from $k{+}1$ to j, plus the comparison on the path from $j{+}1$ to k. We can rewrite o' by separating out $(k{+}1, j)$ and $(j{+}1, k)$, i.e., $o' = (o_{j,k} \cup o')^+$. Applying postulates P_{3-4}, we now get that

$$\begin{aligned}((\pi \rhd o) \cup o')^+ &= \pi \rhd (o \cup o')^+\\ &= \pi \rhd (o_{j,k} \cup o')^+\\ &\subseteq ((\pi \rhd o_{j,k}) \cup o')^+.\end{aligned}$$

Using the assumption that $(j, j{+}1) \in \pi \rhd o$ and the fact that $(j, j{+}1) \notin o'$, we can thus infer that $(j, j{+}1) \in \pi \rhd o_{j,k}$. This, in turn, implies that $(j, j{+}1) <_\pi (k, k{+}1)$ and hence $(j, j{+}1)$ belongs to a lower level of δ_π than $(k, k{+}1)$: but this contradicts the conclusion drawn earlier that $(k, k{+}1)$ belongs to a level $l \leq i$, where i is the level of $(j, j{+}1)$.

Case 2. If (j, k) is not a direct comparison of π, then it is inferred from some direct comparisons of π that end up in $\pi \rhd o$, together with comparisons in o. We can now apply the reasoning from Case 1 to the direct comparisons of π that go into inferring (j, k), to show that they must be in $\mathrm{add}^*_{\leq_\pi}(o)$. This, in turn, implies that (j, k) will be in $\mathrm{add}^*_{\leq_\pi}(o)$ as well.

The reasoning for the other direction is similar. □

Theorems 2 and 3 describe preference revision operators that rely on total preorders $\leq_\pi$ on δ_π, where a tie between two direct comparisons means that if they cannot both be added, then they are both passed over. We can eliminate this indeciseveness by using *linear* orders on δ_π instead of preorders: this ensures that any two direct comparisons of π can be clearly ranked with respect to each other, and that a revision operator is always in a position to choose among them. On the postulate site, linear orders can be characterized by tightening the notion of a π-completion and, with it, postulate P_1. Thus, a *decisive π-completion of o* is defined as:

$$\lfloor o \rfloor^{\mathsf{D}}_{\pi} = \{(o \cup \delta)^+ \in \mathcal{O}_V \mid \emptyset \subset \delta \subseteq \delta_\pi\}.$$

Changing the format of the revision output requires changing the postulate that speaks about this format as well. The decisive version of P_1 is then written, for any $\pi \in \mathcal{C}_V$ and $o \in \mathcal{O}_V$, as:

(P_{D}) $\pi \rhd o \in \lfloor o \rfloor^{\mathsf{D}}_{\pi}$.

A *decisive preference assignment* a is a function $a \colon \mathcal{C}_V \to \mathcal{C}_{V \times V}$ mapping every $\pi \in \mathcal{C}_V$ to a linear preorder $<_\pi$ on δ_π. We can now show the following result.

Theorem 4. *A revision operator $\rhd$ satisfies postulates P_{D} and P_{2-4} if and only if there exists a decisive preference assignment a such that, for any $\pi \in \mathcal{C}_V$ and $o, o_1, o_2 \in \mathcal{O}_V$ such that o_1, o_2 are coordinated with respect to π, $\rhd$ is the a-induced preference revision operator.*

Proof. The proofs for Theorems 2 and 3 work here with minimal adjustments. Note that when choosing between two direct comparisons, postulate P_{D} does not allow $\triangleright$ to be indifferent anymore. This means that the revealed preference relation on δ_π ends up being linear. □

We can see, thus, that what seems like a weakness in the original formulation of the problem, i.e., the mismatch in type between the input (a chain) and the output (an spo) of a revision operator, can be resolved by requiring the ranking on comparisons to be strict. However, in the present setup this amounts to a less general result, which is why we presented our work in this manner.

7 Concrete Preference Revision Operators

Theorems 2, 3 and 4 articulate an important lesson: preference revision performed in a principled manner, i.e., in accordance with P_{1-4} or P_{D} and P_{2-4}, involves having preferences over comparisons. Thus, to obtain concrete operators one must look at ways of ranking the comparisons in a preference π. We sketch here two simple solutions, as proof of concept.

The *trivial assignment* a^{t} is defined by taking:

$$(i, i{+}1) \approx^{\mathrm{t}}_{\pi} (j, j{+}1),$$

while the *lexicographic assignment* a^{lex} is defined by taking:

$$(i, i{+}1) <^{\mathrm{lex}}_{\pi} (j, j{+}1),$$

if $i < j$, for any $\pi \in \mathcal{C}_V$ and $(i, i{+}1), (j, j{+}1) \in \pi$. Intuitively, the trivial assignment makes all direct comparisons of π equally desirable, while the lexicographic assignment orders them in lexicographic order.

These assignments induce the *trivial* and *lexicographic* operators $\triangleright^{\mathrm{t}}$ and $\triangleright^{\mathrm{lex}}$, respectively. It is straightforward to see that $\leq^{\mathrm{t}}_{\pi}$ is a preorder and $<^{\mathrm{lex}}_{\pi}$ is a linear order, prompting the following result.

Proposition 1. The operator $\triangleright^{\mathrm{t}}$ satisfies postulates P_{1-4}. The operator $\triangleright^{\mathrm{lex}}$ satisfies postulates P_{D} and P_{2-4}.

The following example illustrates that the two operators can give different results on the same input.

Example 8. *For π and o as in Example 1, the trivial operator ranks all direct comparisons of π, i.e.,* $(1, 2)$ *and* $(2, 3)$*, equally, and hence either adds all or none of them to o. Since adding both leads to a cycle, it ends up adding none and hence $\pi \triangleright^{\mathrm{t}} o = (3, 1)$.*

The lexicographic assignment ranks $(1, 2)$ *as better than* $(2, 3)$*, and hence adds* $(1, 2)$ *after which it runs out of options, i.e., $\pi \triangleright^{\mathrm{lex}} o = (3, 1, 2)$.*

8 Conclusion

We have presented a model of preference change according to which revising a preference π goes hand in hand with having preferences over the comparisons of π, thereby providing a rigorous formal treatment to intuitions found elsewhere in the literature (Sen 1977; Grüne-Yanoff and Hansson 2009a). Interestingly, the postulates describing preference revision are analogous to existing postulates offered for propositional enforcement (Haret, Wallner, and Woltran 2018), an operation used to model changes in Abstract Argumentation Frameworks (AFs) (Dung 1995).

Our treatment unearthed interesting aspects of preference revision, such as the issue of coordination between successive instances of new preference information (Section 4) and the non-obvious solution to the question of how to rank two comparisons relative to each other (Section 6). These aspects are taken for granted in regular propositional revision, but prove key to successful application of revision to the more specialized context of transitive relations on a set of items, i.e., preference orders. In this respect, preference revision is akin to revision for fragments of propositional logic (Delgrande, Peppas, and Woltran 2018; Creignou et al. 2018), and raises the possibility of exporting this approach to other formalisms in this family. The addition procedure in particular, lends itself to application in other formalisms by slight tweaking of the acceptance condition, and could thus supply some interesting lessons for revision in general, in particular to revision-like operators for specialized formalisms, such as that of AFs, mentioned above.

There is also ample space for future work with respect to the present framework itself. To facilitate exposition of the main ideas we imposed certain restrictions on the primary notions. Lifting these restrictions would yield broader results that would potentially cover more ground and apply to a more diverse set of inputs. We can consider, for instance, revising strict partial orders in general (not just linear orders), and using rankings that involve all comparisons of the initial preference order (not just the direct ones). As the space of possibilities becomes larger, the choice problems on this space become increasingly more complex as well. Finding the right conditions under which the choice mechanism corresponds to a set of appealing postulates requires a delicate balance of many elements, and holds the promise for interesting results.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic* 50(2):510–530.

Benthem, J., and Liu, F. 2014. Deontic Logic and Preference Change. *IfCoLog Journal of Logics and their Applications* 1(2):1–46.

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research (JAIR)* 21:135–191.

Bradley, R. 2007. The kinematics of belief and desire. *Synthese* 156(3):513–535.

Cadilhac, A.; Asher, N.; Lascarides, A.; and Benamara, F. 2015. Preference change. *Journal of Logic, Language and Information* 24(3):267–288.

Chomicki, J., and Song, J. 2005. Monotonic and Nonmonotonic Preference Revision. In *Proc. IJCAI 2005 Multidisciplinary Workshop on Advances in Preference Handling*.

Chomicki, J. 2003. Preference formulas in relational queries. *ACM Trans. Database Syst.* 28(4):427–466.

Creignou, N.; Haret, A.; Papini, O.; and Woltran, S. 2018. Belief Update in the Horn Fragment. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 1781–1787.

Delgrande, J. P.; Peppas, P.; and Woltran, S. 2018. General Belief Revision. *Journal of the ACM (JACM)* 65(5):29:1–29:34.

Dell'Acqua, P., and Pereira, L. M. 2005. Preference Revision Via Declarative Debugging. In *Portuguese Conference on Artificial Intelligence*, 18–28. Springer.

Domshlak, C.; Hüllermeier, E.; Kaci, S.; and Prade, H. 2011. Preferences in AI: An Overview. *Artificial Intelligence* 175(7-8):1037–1052.

Dung, P. M. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.* 77(2):321–358.

Fermé, E. L., and Hansson, S. O. 2018. *Belief Change: Introduction and Overview*. Springer Briefs in Intelligent Systems. Springer.

Frankfurt, H. G. 1988. Freedom of the Will and the Concept of a Person. In *What is a person?* Springer. 127–144.

Freund, M. 2004. On the revision of preferences and rational inference processes. *Artificial Intelligence* 152(1):105–137.

Grüne-Yanoff, T., and Hansson, S. O., eds. 2009a. *Preference Change: Approaches from Philosophy, Economics and Psychology*, volume 42 of *Theory and Decision Library A*. Springer.

Grüne-Yanoff, T., and Hansson, S. O. 2009b. From Belief Revision to Preference Change. In *Preference Change: Approaches from Philosophy, Economics and Psychology*. Springer. 159–184.

Grüne-Yanoff, T. 2013. Preference change and conservatism: comparing the bayesian and the AGM models of preference revision. *Synthese* 190(14):2623–2641.

Hansson, S. O. 1995. Changes in preference. *Theory and Decision* 38(1):1–28.

Hansson, S. O. 2017. Logic of Belief Revision. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition.

Haret, A.; Wallner, J. P.; and Woltran, S. 2018. Two Sides of the Same Coin: Belief Revision and Enforcing Arguments. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 1854–1860.

Harsanyi, J. C. 1955. Cardinal Welfare, Individualistic Ethics, and Interpersonal Comparisons of Utility. *Journal of Political Economy* 63(4):309–321.

Jeffrey, R. C. 1974. Preference among preferences. *Journal of Philosophy* 71(13):377–391.

Katsuno, H., and Mendelzon, A. O. 1992. Propositional Knowledge Base Revision and Minimal Change. *Artificial Intelligence* 52(3):263–294.

Lang, J., and van der Torre, L. W. N. 2008. Preference Change Triggered by Belief Change: A Principled Approach. In *Proceedings of the 8th International Conference on Logic and the Foundations of Game and Decision Theory (LOFT 8)*, 86–111.

Liu, F. 2011. *Reasoning About Preference Dynamics*, volume 354 of *Synthese Library*. Springer.

Ma, J.; Benferhat, S.; and Liu, W. 2012. Revising Partial Pre-Orders with Partial Pre-Orders: A Unit-Based Revision Framework. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, 633–637.

Nozick, R. 1994. *The Nature of Rationality*. Princeton University Press.

Pigozzi, G.; Tsoukiàs, A.; and Viappiani, P. 2016. Preferences in artificial intelligence. *Annals of Mathematics and Artificial Intelligence* 77(3-4):361–401.

Rossi, F., and Mattei, N. 2019. Building Ethically Bounded AI. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, 9785–9789.

Rossi, F.; Venable, K. B.; and Walsh, T. 2011. *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Russell, S. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin.

Sen, A. K. 1977. Rational Fools: A Critique of the Behavioral Foundations of Economic Theory. *Philosophy & Public Affairs* 317–344.

On the Relationship of Modularity Notions in Abstract Argumentation

Tom Friese[1] and **Markus Ulbricht**[2]
[1]TU Dresden
[2]TU Wien, Institute of Logic and Computation
mulbricht@informatik.uni-leipzig.de

Abstract

Abstract argumentation frameworks (AFs) as proposed by Dung in his seminal 1995 paper are by now a well-established and flourishing research area in knowledge representation and reasoning. Various aspects of AFs have been extensively studied over the last 25 years. Many of these are concerned with computational properties of reasoning problems in an AF, for example deciding whether a certain set of arguments is a so-called extension w.r.t. a given semantics. Recently, properties of semantics have been investigated in order to examine the potential of several divide & conquer techniques. For example, the notion of SCC decomposability formalizes how to calculate an extension by evaluating the strongly connected components individually. We extend this line of research and compare several notions with a special focus on the recently introduced modularization property.

1 Introduction

In his seminal 1995 paper (Dung 1995), Dung initiated the investigation of abstract argumentation frameworks (AFs). A Dung-style AF is a directed graph where nodes are interpreted as arguments and edges as attacks between them. In the literature, various *semantics* have been proposed, that is, mappings which assign to an AF a set of so-called *extensions*, i.e. sets of commonly acceptable arguments.

An important feature of knowledge representation formalisms is their expressive power, i. e. the questions which kind of knowledge can be expressed and which not. This is by no means limited to knowledge representation and reasoning, but an important question for various formalisms investigated in theoretical computer science in general. Much research is driven by the expressive power of the studied framework as it hints at the need to propose extensions in order to augment its ability to model certain application scenarios. There is however a natural trade-off since the search for "good" formalisms has to take both expressive power and computational complexity of natural decision problems into consideration.

Both expressive power and computational complexity are quite well understood for various extensions and semantics of AFs. We refer the reader to (Baroni, Caminada, and Giacomin 2011; 2018) for an overview of AF semantics and to (Dvořák and Dunne 2018) regarding the computational complexity. A common technique for finding algorithmic solutions to problems arising in computer science are so-called divide & conquer approaches. In a nutshell, the underlying idea is to partition the given problem into smaller subproblems and solving them individually. In order to formalize such approaches for reasoning with AFs, several modularity notions have been introduced in the literature (Baroni et al. 2014; Baroni, Giacomin, and Liao 2018); and similar work has also been done for e.g. logic programs (Lifschitz and Turner 1994) or default logic (Turner 1996). Since AFs are a non-monotonic logic, this is an inherently sophisticated endeavor. More specifically, given a partial solution to some problem (in our case a σ-extension E of a subframework of an AF F), non-monotonicity usually undermines attempts to calculate an additional partial solution in order to combine both of them to a single one of the whole problem. Driven by this observation, researchers have studied situations in which such approaches work due to the theoretical properties of the semantics. Most notably, notions like SCC-recursiveness, directionality, splitting (Baumann 2011), or full decomposability have been proposed and investigated.

More recently, a property called modularization has been introduced (Baumann, Brewka, and Ulbricht 2020a). The overall idea is as follows: Starting from an extension E, consider the subframework consisting of these arguments whose acceptance status is not yet decided. If in this situation E' is an extension of the aforementioned auxiliary subframework, then $E \cup E'$ shall be an extension of the whole AF. Although this is a rather simple property – both intuitively as well as technically – the results reported in (Baumann, Brewka, and Ulbricht 2020a) suggest it to be quite powerful: Among others, the accepted arguments w.r.t. the *grounded* extension can be characterized as well as the so-called strongly admissible sets (Caminada and Dunne 2019; Baumann, Linsbichler, and Woltran 2016) of an AF. Moreover semantics satisfying modularization can be computed step-wise in a way that at first glance seems unexpected for a non-monotonic logic like AFs.

In this paper we are going to compare modularization to well-established modularity notions (Baroni, Giacomin, and Liao 2018) on an abstract level. More specifically, we i) extend the list of semantics satisfying modularization in Section 3, ii) show that modularization is incomparable to the other notions in general by developing suitable counterexamples in Section 4, iii) establish that under mild assumptions full decomposability implies modularization for semantics refining co in Section 5, and iv) give a criterion under which modularization and directionality imply SCC-decomposability in Section 6. Due to space restrictions, we will introduce the modularity notions quite briefly. We refer the reader to (Baroni, Giacomin, and Liao 2018) for a gentle introduction of the concepts.

2 Background

2.1 Standard Concepts and Classical Semantics

We fix a non-finite background set $\mathcal{U}$. An AF (Dung 1995) is a directed graph $F = (A, R)$ where $A \subseteq \mathcal{U}$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. $\mathcal{F}$ denotes the set of all finite AFs over $\mathcal{U}$; we shall consider finite AFs only.

For $S \subseteq A$ we let $F{\downarrow_S} = (A \cap S, R \cap (S \times S))$. For $a, b \in A$, if $(a, b) \in R$ we say that a *attacks* b as well as a *attacks* (the set) E given that $b \in E \subseteq A$. Moreover, we use $E^+_F = \{a \in A \mid E \text{ attacks } a \text{ in } F\}$ and $E^{\oplus}_F = E \cup E^+_F$. The latter set is known as the *range* of E in F. When clear from the context, we omit the subscript F. A set E is conflict-free in F (for short, $E \in cf(F)$) iff for no $a, b \in E$, $(a, b) \in R$. We say a set E *defends* an argument a (in F) if any attacker of a is attacked by some argument of E, i. e. for each b with $(b, a) \in R$, there is $c \in E$ such that $(c, b) \in R$.

A *semantics* σ is a mapping $\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}}$ where $F \mapsto \sigma(F) \subseteq 2^A$, i. e. given an AF $F = (A, R)$ a semantics returns a subset of 2^A. Besides conflict-free and admissible sets (abbr. cf and ad) we consider stable, semi-stable, complete, preferred, grounded, ideal and eager semantics (abbr. co, gr, pr, stb, ss, il and eg, respectively).

Definition 2.1. Let $F = (A, R)$ be an AF and $E \in cf(A)$.

1. $E \in ad(F)$ iff E defends all its elements,
2. $E \in co(F)$ iff $E \in ad(F)$ and, for any x defended by E, we have $x \in E$,
3. $E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$,
4. $E \in pr(F)$ iff E is $\subseteq$-maximal in $ad(F)$,
5. $E \in stb(F)$ iff $E \in cf(A)$ and any $a \in A \setminus E$ is attacked by E,
6. $E \in ss(F)$ iff $E \in co(F)$ with $\subseteq$-maximal range $E^{\oplus}$,
7. $E \in il(F)$ iff $E \in co(F)$ and $E \subseteq \bigcap pr(F)$ and $\subseteq$-maximal wrt. the conjunction of both properties,
8. $E \in eg(F)$ iff $E \in co(F)$ and $E \subseteq \bigcap ss(F)$ and $\subseteq$-maximal wrt. the conjunction of both properties.

Some of our definitions and proofs make use of the labelling-approach to semantics. For brevity, we state that labellings and extensions are somewhat interchangeable for conflict-free semantics (Baroni, Caminada, and Giacomin 2011, Defintion 7, 8, 14, 15). We use the set of labellings $\{in, out, undec\}$. For an AF $F = (A, R)$ and some extension-based semantics σ, s.t. $E \in \sigma(F)$, we will say $Lab(e) = in$ iff $e \in E$, $Lab(e) = out$ iff $e \in E^+$ and $Lab(e) = undec$ iff $e \in A \setminus E^{\oplus}$. $Lab = \{(e, Lab(e))\}$ denotes the set of tuples of arguments and their labels for all $e \in A$. Essentially, this set corresponds to some extension E. Given some $X \subseteq A$, we define the restriction of Lab to X, denoted $Lab\downarrow_X$, as $Lab \cap (X \times \{in, out, undec\})$. For ease of notation, we will sometimes also refer to Lab as the tuple $(\{e \in A \mid Lab(e) = in\}, \{e \in A \mid Lab(e) = out\}, \{e \in A \mid Lab(e) = undec\})$.

2.2 Weak Admissible-Based Semantics

The reduct is the central notion in the definition of weak admissible semantics (Baumann, Brewka, and Ulbricht 2020b).

Definition 2.2. Let $F = (A, R)$ be an AF and let $E \subseteq A$. The E-reduct of F is the AF $F^E = (E^*, R \cap (E^* \times E^*))$ where $E^* = A \setminus E^{\oplus}_F$.

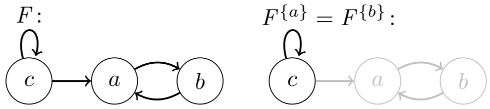
By definition, F^E is the subframework of F obtained by removing the range of E as well as corresponding attacks, i. e. $F^E = F{\downarrow_{A \setminus E^{\oplus}}}$. Intuitively, the E-reduct contains those arguments whose status still needs to be decided, assuming the arguments in E are accepted. This intuition is captured in the forthcoming central definition.

Definition 2.3. For an AF $F = (A, R)$, $E \subseteq A$ is called *weakly admissible* (or *w-admissible*) in F ($E \in ad^w(F)$) iff

1. $E \in cf(F)$ and
2. for any attacker y of E we have $y \notin \bigcup ad^w\left(F^E\right)$.

The major difference between the standard definition of admissibility and the "weak" one is that extensions do not have to defend themselves against *all* attackers: attackers which do not appear in any w-admissible set of the reduct can be neglected.

Example 2.4. Consider the following simple example:



While we observe $\{a\} \notin ad(F)$, we can verify weak admissibility of $\{a\}$ in F. Obviously, $\{a\}$ is conflict-free in F (condition 1). Since c is the only attacker of $\{a\}$ in $F^{\{a\}}$ we have to check $c \notin \bigcup ad^w\left(F^{\{a\}}\right)$ (condition 2). Since $\{c\}$ is not conflict-free in the reduct $F^{\{a\}} = (\{c\}, \{(c, c)\})$ we find $\{c\} \notin ad^w\left(F^{\{a\}}\right)$ yielding $\bigcup ad^w\left(F^{\{a\}}\right) = \emptyset$. Hence, $c \notin \bigcup ad^w\left(F^{\{a\}}\right)$, and thus $\{a\} \in ad^w(F)$. ◇

Following the classical Dung-style semantics, *weakly preferred* extensions are defined as $\subseteq$-maximal w-admissible extensions.

Definition 2.5. For an AF $F = (A, R)$, $E \subseteq A$ is called *weakly preferred* (or *w-preferred*) in F ($E \in pr^w(F)$) iff E is $\subseteq$-maximal in $ad^w(F)$.

For more details regarding the definition and basic properties of weak admissibility we refer the reader to (Baumann, Brewka, and Ulbricht 2020b).

2.3 Modularity Notions

Directionality. We call $U \subseteq A$ *unattacked* if there are no two arguments $a \in A \setminus U$ and $u \in U$ with $(a, u) \in R$. A semantics satisfies *directionality* if for any unattacked set U we have $\sigma(F{\downarrow_U}) = \{E \cap U \mid E \in \sigma(F)\}$.

Full Decomposability. Let us now define the notion of full decomposability (Baroni et al. 2014). Let $F = (A, R)$ be an AF. Let $E \subseteq A$. The *input* of E, which we will denote as E^{inp}, is defined as

$$E^{inp} := \{a \in A \setminus E \mid \exists e \in E : (a, e) \in R\}.$$

The *conditioning relation* of E, is $E^R := R \cap (E^{inp} \times E)$. An AF *with input* is a tuple (F, X, L_X, R_X) where X is some set of arguments s. t. $X \cap A = \emptyset$, L_X is some labeling for X, i. e. $L_X = \{(x, Lab(x)) \mid x \in X\}$, and R_X is a relation s. t. $R_X \subseteq X \times A$. Given an AF with input (F, X, L_X, R_X), the *standard AF* w.r.t. (F, X, L_X, R_X) is $F' = (A \cup X', R \cup R'_X)$ where

$$\begin{aligned} X' &= X \cup \{a' \mid (a, out) \in L_X\} \\ R'_X &= R_X \cup \{(a', a) \mid (a, out) \in L_X\} \\ &\quad \cup \{(a, a) \mid (a, undec) \in L_X\}. \end{aligned}$$

A *local function* L_F assigns to any argumentation framework with input a (possibly empty) set of labelings. Given some semantics $\sigma \subseteq cf$, we denote the labellings corresponding to $\sigma(F)$ as $\mathcal{L}_\sigma(F)$. The *canonical local function* of σ (also called the local function of σ) is defined as the following set of labellings: $F_\sigma(F, X, L_X, R_X) = \{Lab{\downarrow_A} \mid Lab \in \mathcal{L}_\sigma(F')\}$ where F' is the standard argumentation framework w.r.t. (F, X, L_X, R_X). A *partition* of A is a set $\{P_1, ..., P_n\}$, s. t. $\forall i \in \{1, ..., n\}$ we have i) $\emptyset \neq P_i \subseteq A$, ii) $\bigcup_{i=1...n} P_i = A$ and iii) $P_i \cap P_j = \emptyset$ for $i \neq j$.

We say that σ is *fully decomposable* if there is a local function F_σ s. t. for every AF F and every partition $P = \{P_1, ..., P_n\}$ of A, we have $\mathcal{L}_\sigma(F) = U(P, F, F_\sigma)$ where

$$U(P, F, F_\sigma) = \left\{\bigcup L_{P_i} \mid L_{P_i} \in F_\sigma(F{\downarrow_{P_i}}, P_i^{inp}, L, P_i^R)\right\}$$

with $L = \left(\bigcup_{j \neq i} L_{P_j}\right){\downarrow_{P_i^{inp}}}$.

SCC-recursiveness. Given an AF $F = (A, R)$, an SCC S is a maximal set of arguments s. t. in $F{\downarrow_S}$ the following condition holds: For any two $a_1, a_n \in A(F{\downarrow_S})$ there is a sequence $a_1, \ldots, a_n$ of arguments with $a_i \in S$ and $(a_i, a_j) \in R$. We denote by $SCCS_F$ the set of all SCCs of F. If there is $a \in P$ and $b \in S$ s. t. $(a, b) \in R$ for SCCs S, P, we call P a *parent* of S. By $S^\prec$ we denote all ancestors of S which are induced by this parent relation. We consider the following sets:

- $D_F(S, E) = \{a \in S \mid \exists P \in S^\prec, b \in E \cap P :\ b \to a\}$,
- $P_F(S, E) = \{a \in S \mid \exists b \in P \in S^\prec : (b, a) \in R,\ E \not\to b\} \setminus D_F(S, E)$,
- $U_F(S, E) = S \setminus (D_F(S, E) \cup P_F(S, E))$.

We let $UP_F(S, E) = U_F(S, E) \cup P_F(S, E)$. We say a semantics σ is SCC-recursive if for any AF $F = (A, R)$, we have $\sigma(F) = \bar{\sigma}(F, A)$, where for any AF $F = (A, R)$ and any $C \subseteq A$, $\bar{\sigma}(F, C) \subseteq 2^A$ is given as follows: $E \subseteq A$ satisfies $E \in \bar{\sigma}(F, C)$ iff

- if $|SCCS(F)| = 1$, then $E \in \bar{\sigma}_b(F, C)$ for a "base function" $\bar{\sigma}_b(F, C)$,
- otherwise, for all $S \in SCCS(F)$ it holds that $E \cap S \in \bar{\sigma}(F{\downarrow_{UP_F(S,E)}}, U_F(S, E) \cap C)$.

Modularization. Finally, we introduce the more recent modularization property from (Baumann, Brewka, and Ulbricht 2020a). Modularization does not restrict the structure of the given AF, but rather focuses on compatibility of extensions E and their reduct F^E in the following sense:

Definition 2.6. A semantics σ satisfies *modularization* if for any AF F we have: $E \in \sigma(F)$ and $E' \in \sigma\left(F^E\right)$ implies $E \cup E' \in \sigma(F)$.

We refer the reader to (Baumann, Brewka, and Ulbricht 2020a) for an in-depth discussion of modularization, including results demonstrating that Dung's semantics satisfy this property.

Proposition 2.7 (see (Baumann, Brewka, and Ulbricht 2020a))**.** *The semantics $\sigma \in \{ad, co, gr, pr, stb\}$ satisfy modularization.*

3 Warm Up

In this section we briefly show modularization for the semantics not mentioned in Proposition 2.7 in order to augment previous results. That is, we consider ss, eg, and il in this section. In all three cases we will show that the empty set is the only extension in the reduct F^E, i. e. $\sigma(F^E) = \{\emptyset\}$ for $\sigma \in \{ss, eg, il\}$ and $E \in \sigma(F)$. Then, modularization is clear since there is nothing to show whenever $E' = \emptyset$ in Definition 2.6.

We start with semi-stable semantics ss. Here, we can immediately show that the empty set is the only extension of the reduct, without any preparatory considerations.

Proposition 3.1. *If $E \in ss(F)$, then $ss\left(F^E\right) = \{\emptyset\}$.*

Proof. Let $F = (A, R)$ be an AF. Let $E \in ss(F)$. Assume $\emptyset \neq E' \in ss\left(F^E\right)$. Both E and E' are admissible by definition and we therefore infer $E \cup E' \in ad(F)$ from the modularization property of ad. Since $E' \subseteq A \setminus E^\oplus$ it is clear that $E^\oplus \subsetneq (E \cup E')^\oplus$ implying E was not a semi-stable extension of F. □

Corollary 3.2. *ss satisfies modularization.*

The proof for eg is more involved. We start by inferring some auxiliary results. We will state them explicitly as they will be of interest later on.

Proposition 3.3. *Let F be an AF and let $E \in ad(F)$ with $E = E' \dot{\cup} E''$ for some $E' \in ad(F)$. Then $E'' \in ad(F^{E'})$.*

Proof. Clearly, $E'' \in cf(F)$. Now assume a attacks E'' in $F^{E'}$. Due to $E \in ad(F)$, some $e \in E$ counterattacks a. Since a occurs in $F^{E'}$ we infer $a \notin (E')^+$ and hence, $a \in (E'')^+$. Hence, E'' counterattacks a. □

Next we show that complete semantics satisfy this property as well.

Proposition 3.4. *Let F be an AF and let $E \in co(F)$ with $E = E' \dot{\cup} E''$ for some $E' \in co(F)$. Then $E'' \in co(F^{E'})$.*

Proof. We already know $E'' \in ad(F^{E'})$. By $E \in co(F)$, there are no unattacked arguments in $F^{E' \cup E''} = (F^{E'})^{E''}$ and therefore E'' is complete in $F^{E'}$ as well. □

As a last step before finally turning to eg itself, we infer an analogous result for ss. This is required since eg builds upon semi-stable extensions.

Proposition 3.5. *Let F be an AF and let $E \in ss(F)$ with $E = E' \dot\cup E''$ for some $E' \in ad(F)$. Then $E'' \in ss(F^{E'})$.*

Proof. We already know $E'' \in ad(F^{E'})$ since $ss \subseteq ad$. Now assume E'' is not semi-stable in $F^{E'}$. Then there is some admissible $S \in ad\left(F^{E'}\right)$ with $(E'')^{\oplus} \subsetneq S^{\oplus}$. Since E'' and S occur in $F^{E'}$, $E^{\oplus} = (E' \cup E'')^{\oplus} \subsetneq (E' \cup S)^{\oplus}$. Since by modularization we have $E' \cup S \in ad(F)$, we infer $E \notin ss(F)$, contradiction. □

Now we are ready to infer the desired result for eg.

Proposition 3.6. *If $E \in eg(F)$, then $eg\left(F^{E}\right) = \{\emptyset\}$.*

Proof. Let $F = (A, R)$ be an AF and let $E \in eg(F)$. Consider the reduct F^{E} and assume $E' \in eg(F^{E})$ is not empty. Let S be a semi-stable extension of F. By definition of eg, $E \subseteq S$. Our goal is to show $E' \subseteq S$ as well, yielding a contradiction since $E \cup E' \in ad(F)$ by modularization of ad; hence the eager extension of F must at least contain $E \cup E'$. To this end note that $S = E \cup S'$ for $E \in ad(F)$ and some S'. By the above proposition, $S' \in ss\left(F^{E}\right)$ and hence $E' \subseteq S' \subseteq S$ and we are done. □

Corollary 3.7. *eg satisfies modularization.*

In order to lift the above proof technique to il as well it suffices to note the following adjustment to Proposition 3.5.

Proposition 3.8. *Let F be an AF and let $E \in pr(F)$ with $E = E' \cup E''$ for some $E' \in ad(F)$. Then $E'' \in pr(F^{E'})$.*

Proof. According to (Baumann, Brewka, and Ulbricht 2020a, Proposition 3.2) $E \in pr$ iff $E \in ad(F)$ and F^{E} does not possess any admissible argument. We already know admissibility of E'' in $F^{E'}$. Moreover, $F^{E} = (F^{E'})^{E''}$ does not contain admissible arguments; thus we are done. □

This yields the same behavior for il as well. First, we again infer that the reduct does not tolerate any non-empty extension.

Proposition 3.9. *If $E \in il(F)$, then $il\left(F^{E}\right) = \{\emptyset\}$.*

And as before, this yields modularization.

Corollary 3.10. *il satisfies modularization.*

4 Incomparability Results

The goal of this section is to investigate whether there is a general relationship between modularization and the other modularity notions we introduced. For example, under mild assumptions the splitting property implies directionality (Baroni, Giacomin, and Liao 2018, Proposition 3.3), i.e. each semantics σ satisfying the splitting property (and $\sigma(F) \neq \emptyset$ for each AF F) also satisfies directionality. We call two properties *incomparable* if there is no such relation in general. For example, directionality and full decomposability are incomparable since

- gr satisfies directionality, but not full decomposability whereas
- stb satisfies full decomposability, but not directionality

as summarized in (Baroni, Giacomin, and Liao 2018, Table 1). Our first main result is that modularization is incomparable to each notion considered in this paper.

Theorem 4.1. *The following properties are incomparable in general:*

- *Modularization and directionality,*
- *Modularization and full deomposability,*
- *Modularization and SCC recursiveness,*

In the following subsections, we will prove this theorem by giving suitable results and counterexamples.

4.1 Modularization vs. Directionality

We define the following auxiliary semantics returning all sects of arguments with no in-going attacks.

Definition 4.2. Let $F = (A, R)$ be an AF. We call an argument $a \in A$ *unquestioned* if there is no $b \in A$ with $b \to a$. We let U_F be the set of all unquestioned arguments of F. An extension $E \subseteq A$ is called an *unquestioned extension* ($E \in un(F)$) if $E \subseteq U_F$.

We want to emphasize that while un is admittedly not a very meaningful semantics in most application scenarios, it follows a natural (yet very cautious) motivation. In this sense, we believe un is simple, but by no means a sophisticated artificial semantics. As the following result shows, it witnesses that directionality does not imply modularization.

Proposition 4.3. *The semantics un satisfies* directionality*, but not* modularization*.*

Proof. Let $F = (A, R)$ be an AF.

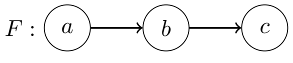
Our first step is to observe the following: Given $a \in U$, a is unquestioned in F iff it is unquestioned in $F{\downarrow_U}$. Here, the "⇒"-direction is immediate since $F{\downarrow_U}$ possesses fewer arguments. For "⇐", we use the fact that U is unattacked in F.

Now let us show that un satisfies directionality, i. e. $\{E \cap U \mid E \in un(F)\} = un(F{\downarrow_U})$.

($\subseteq$) For $E \in un(F)$ we have $E \cap U \subseteq U$ and $E \cap U \subseteq C_F$, i. e. $E \cap U$ is a set of unquestioned arguments occurring in U. By definition, $E \cap U \in un(F{\downarrow_U})$.

($\supseteq$) Given $E \in un(F{\downarrow_U})$, from the above statement we infer that each argument in E is unquestioned in F as well, i. e. $E \in un(F)$. Clearly, $E \cap U = E$ and we are done.

We have left to show that un does not satisfy modularization. As a counterexample, consider the following simple AF F:



It is easy to see that $un(F) = \{\emptyset, \{a\}\}$ and $un\left(F^{\{a\}}\right) = \{\emptyset, \{c\}\}$. In order to satisfy the modularization property, the set $\{a, c\}$ would now also have to be an extension of $un(F)$, but this is not the case. Therefore, the semantics un does not satisfy modularization. □

Thus, for a semantics σ it is not sufficient to satisfy the directionality property in order to also enforce the modularization property. In particular:

Corollary 4.4. *Directionality does not imply modularization.*

Since modularization is a rather general property, it is no surprise that the other implication does not hold, either. To see this, it suffices to consider an arbitrary semantics satisfying modularization, but not directionality. For example, (Dauphin, Rienstra, and van der Torre 2020b, Proposition 10) showed that co^w —the weak counterpart to Dung's classical complete semantics— is not directional. However, modularization is satisfied (Baumann, Brewka, and Ulbricht 2020a, Theorem 4.13).

Corollary 4.5. *Modularization does not imply directionality.*

Hence we established the first item in Theorem 4.1.

4.2 Modularization vs. Full Decomposability

To see that full decomposability alone does not imply modularization, consider again the semantics un, which satisfies full decomposability as formalized in the following proposition.

Proposition 4.6. *The semantics un satisfies* full decomposability*, but not* modularization*.*

Sketch of Proof. For $F = (A, R)$ let us consider

$$F_{un}(F, X, L_X, R_X) =$$
$$\{\big(E,\ E^+ \cup I_1,\ (A \backslash E^+) \backslash I_2\big) \mid E = A \cap E' \in un(F')\}$$

where

$$I_1 = \{x \in A \backslash E^+ \mid \exists x' \in X, (x', x) \in R_X, (x', in) \in L_X\},$$
$$I_2 = \{x \in A \mid \exists x' \in X, (x', x) \in R_X, (x', in) \in L_X\}$$

and F' being the standard AF w. r. t. (F, X, L_X, R_X).

In words, we are doing the following: Let F' be the standard AF w. r. t. (F, X, L_X, R_X). We take some set of unquestioned arguments of F' s. t. all the chosen arguments are also present in F and denote this set by E. All $e \in E$ receive the label in. Every $a \in E^+$, as well as every a which is attacked by some $x \in X$ where x is labeled in via L_X, receives the label out. Lastly, all remaining arguments of A receive the label $undec$.

Now suppose we have some partition $P = \{P_1, ..., P_n\}$ of F, some $E \in un(F)$ and Lab_E its corresponding labelling. If $e \in U_F$, then $e \in U_{F\downarrow_{P_i}}$ also holds. We can choose $E_i \in U_{F\downarrow_{P_i}} \subseteq P_i$ for every P_i s. t. $\bigcup_{i=1,...,n} E_i = E$. With $F_{un}(F\downarrow_{P_i}, P_i^{inp}, (\bigcup_{j=1..n, i \neq j} L_{P_j})\downarrow_{P_i^{inp}}, P_i^R)$, we can label each $e \in E_i$ as in for the subframework $F\downarrow_{P_i}$. With the sets I_1 and I_2 defined above, we make sure that every $a \in P_i$ which is not attacked by some accepted $e \in E \cap P_i$ but attacked by some accepted $e \in E \cap P_j$ $(P_j \neq P_i)$ is labelled out (instead of $undec$, which would be the case if we only considered attacks from within P_i). Thus we can recreate Lab_E via $F_{un}(F\downarrow_{P_i}, P_i^{inp}, (\bigcup_{j=1..n, j \neq i} L_{P_j})\downarrow_{P_i^{inp}}, P_i^R)$ and the partition P.

On the other hand, if we combine some elements of $F_{un}(F \downarrow_{P_i}, P_i^{inp}, (\bigcup_{j=1..n, j \neq i L_{P_j}}) \downarrow_{P_i^{inp}}, P_i^R)$ for P, we know that only arguments are labelled in, which were unquestioned in the corresponding standard AFs, i. e. those which did not receive any input from any partition. Thus we can infer that all arguments which received the label in in the union of all the subframeworks $F\downarrow_{P_i}$, are also unquestioned in F. Therefore these arguments form some $E \in un(F)$. □

Recall from the previous section that un does not satisfy modularization. We hence infer the following result.

Corollary 4.7. *Full decomposability does not imply modularization.*

For the other direction consider pr, Dung's classical preferred extensions. Here modularization is satisfied (Baumann, Brewka, and Ulbricht 2020a, Proposition 3.4), but full decomposability is not (Baroni et al. 2014, Example 5).

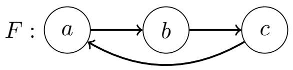
Corollary 4.8. *Modularization does not imply full decomposability.*

This yields the second item in Theorem 4.1.

4.3 Modularization vs. SCC Recursiveness

As before, we start with an example showing that modularization is not implied. Consider the fact that $cf2$ is SCC-recursive (Baroni, Giacomin, and Guida 2005). However, a simple odd cycle shows that $cf2$ does not satisfy modularization.

Example 4.9. Let F be the following AF:



F forms a single SCC, implying that $cf2$ coincides with the base function na, i. e. $cf2(F) = \{\{a\}, \{b\}, \{c\}\}$. The reduct $F^{\{a\}}$ is the single unattacked argument c with $cf2\left(F^{\{a\}}\right) = \{\{c\}\}$, so modularization would imply $\{a, c\} \in cf2(F)$ which is not the case.

Corollary 4.10. *SCC recursiveness does not imply modularization.*

In Proposition 11, (Dauphin, Rienstra, and van der Torre 2020b) showed that co^w is not SCC-recursive. Recall that modularization is satisfied (Baumann, Brewka, and Ulbricht 2020a, Theorem 4.13) and hence:

Corollary 4.11. *Modularization does not imply SCC-recursiveness.*

Thus we are done with the third item in Theorem 4.1.

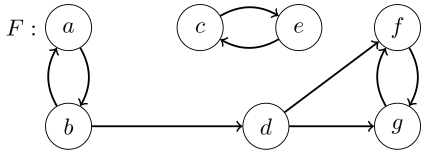
5 Inferring Modularization

Let us now have a closer look at the relation between full decomposability and modularization. As we saw in the previous section there is in general no implication between the two notions. However, both are somewhat similar in their spirit: Intuitively, full decomposability requires that the AF can be partitioned into sub-frameworks and evaluated almost independently, while taking possible inputs into account. Modularization, on the other hand, requires that the reduct F^E can be considered independently of E, so one can think of partitioning F into $E^{\oplus}$ and $A \setminus E^{\oplus}$. In this sense, one could expect full decomposability to be a stronger notion than modularization. As we will see in this section, this is indeed true under mild assumptions for complete-based semantics.

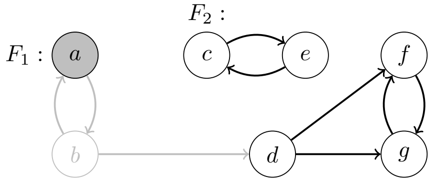
The first assumption we are going to make is based on the notion of a splitting (Baumann 2011) of an AF:

Definition 5.1. Let $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$ be two AFs with $A_1 \cap A_2 = \emptyset$. For $R_3 \subseteq A_1 \times A_2$ we call (F_1, F_2, R_3) a *splitting* of $F = (A_1 \cup A_2, R_1 \cup R_2 \cup R_3)$.

Example 5.2. Consider the following AF F:



We have the splitting (F_1, F_2, R_3) where F_1 is the subframework induced by the arguments $\{a, b\}$, F_2 the subframework consisting of the remaining arguments and $R_3 = \{(b, d)\}$. The property we require can be illustrated as follows: Consider the admissible extension $E_1 = \{a\}$. Now, b is labeled *out*:



The subframework F_2 possesses the admissible extension $E_2 = \{d, e\}$. Let us examine the interaction between E_1 and E_2:

1. We do not expect the attack (b, d) to be problematic for E_2 since b is labeled *out*;
2. we do not expect any attack towards c to be problematic for E_2 since c is already labeled out due to E_2 itself.

Indeed, $E_1 \cup E_2$ is an admissible extension of F.

In the following we formalize this compatibility requirement whenever either the first or the second case holds for each attack in R_3.

Definition 5.3. Let σ be a labeling-based semantics. We say that σ satisfies the *weak splitting* property if for every splitting $F = (F_1, F_2, R_3)$ as well as $Lab_{F_1} \in \mathcal{L}_\sigma(F_1)$ and $Lab_{F_2} \in \mathcal{L}_\sigma(F_2)$, both of the following two statements hold:

1. If $\forall (a_1, a_2) \in R_3$ we have $Lab_{F_1}(a_1) = out$, then we also have $Lab_{F_1} \cup Lab_{F_2} \in \mathcal{L}_\sigma(F)$.
2. If $\forall (a_1, a_2) \in R_3$ we have $Lab_{F_2}(a_2) = out$, then we also have $Lab_{F_1} \cup Lab_{F_2} \in \mathcal{L}_\sigma(F)$.

Since we require the weak splitting property for the main result of this section, we want to emphasize that this premise is rather mild. More specifically, almost all considered semantics adhere to it.

Proposition 5.4. *Consider any semantics σ with $\sigma \in \{ad, co, pr, gr, stb, ad^w, pr^w, il\}$. Then σ satisfies the weak splitting property.*

Second, we introduce a property we call *range compatibility* formalizing that an extension E should be an extension of the subframework $F{\downarrow_{E^{\oplus}}}$ as well.

Definition 5.5. Let σ be a semantics. We say that σ satisfies *range compatibility* if for every AF F and every extension $E \in \sigma(F)$ we have $E \in \sigma(F{\downarrow_{E^{\oplus}}})$.

Again this is requirement is not very restrictive. In fact, again almost all considered semantics in this paper are range compatible. The only exceptions are il and eg.

Proposition 5.6. *Consider any semantics σ with $\sigma \in \{ad, co, pr, gr, stb, ss, ad^w, pr^w\}$. Then σ satisfies range compatibility.*

Now we are ready to perform the preparatory considerations for this sections main theorem. First, we give two auxiliary lemmata showing how arguments of an extension $E \in co(F)$ interact with arguments outside the range $E^{\oplus}$. Note that both of them seem quite technical at first glance, but it is worth noting that both of them formalize a quite intuitive behavior.

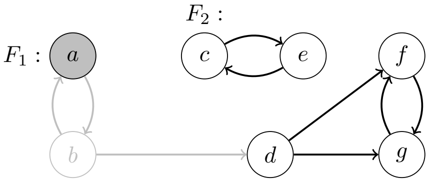
From a technical point of view the following two Lemmata help us to apply the weak splitting property as they show under which conditions certain arguments are labeled out.

Lemma 5.7. *Let $F = (A, R)$ be an AF and let $E \in co(F)$. Furthermore, let Lab be the associated labeling. Then for all arguments $x \in A \setminus E^{\oplus}$ we have: If $(x, e) \in R$ and $e \in E^{\oplus}$, then $Lab(e) = out$.*

Lemma 5.8. *Let $F = (A, R)$ be an AF and let $E \in co(F)$. Let $E' \in co\left(F^E\right)$ and $Lab = (E, E^+, A \setminus E^{\oplus})$. Then for all $e \in E^{\oplus}$ and $e' \in E'$ s. t. $(e, e') \in R$, we have $Lab(e) = out$.*

Now let us turn to the desired relation between modularization and full decomposability. Before giving the formal result and its proof, let us head back to our running example.

Example 5.9. Again consider $E_1 = \{a\} \in \sigma(F)$ for some arbitrary semantics σ.



We observe that F^{E_1} is the subframework induced by the arguments $\{c, d, e, f, g\}$. Now if $E_2 = \{d, e\} \in \sigma\left(F^{E_1}\right)$, then full decomposability applied to $E^{\oplus}$ and $A \setminus E^{\oplus}$ would - assuming σ adheres to our mild assumptions- imply $E_1 \cup E_2$ to be an extension of F. Since this is precisely the requirement of the modularization property, this does the job.

Now, carefully putting together the pieces we collected yields the following result.

Theorem 5.10. *Let $\sigma \subseteq co$ be an extension-based semantics. Let σ_l be the associated labeling-based semantics. If σ_l is fully decomposable and satisfies the weak splitting property as well as range compatibility, then σ satisfies modularization as well.*

Sketch of Proof. Let $F = (A, R)$ be an AF and let $E \in \sigma(F)$ as well as $E' \in \sigma\left(F^E\right)$. Let Lab_E, $Lab_{E'}$ be the labellings corresponding to E and E'. Set $P = \{P_1, P_2\}$ with $P_1 = E^{\oplus}$ and $P_2 = A\left(F^E\right)$ as a partition of A.

Note that $Lab_E{\downarrow}_{P_1} \in \sigma_l(F{\downarrow}_{P_1})$, since σ satisfies range compatibility. As $\sigma \subseteq co$, we can use the canonical local function L_F as the local function for the defintion of decomposability.

Furthermore, since $\sigma \subseteq co$, we can use Lemma 5.7 and Lemma 5.8 to analyze the influence that P_1 and P_2 receive from each other: In P_1, only arguments labelled out are attacked by P_2, while in P_2 any attacker from P_1 is an argument labelled out. Because σ satisfies the weak splitting property, we can therefore infer:

- $Lab_E{\downarrow}_{P_1} \in L_F\left(F{\downarrow}_{P_1}, P_1^{inp}, Lab_{E'}{\downarrow}_{P_1^{inp}}, P_1^R\right)$ and
- $Lab_{E'} \in L_F\left(F{\downarrow}_{P_2}, P_2^{inp}, Lab_E{\downarrow}_{P_2^{inp}}, P_2^R\right)$.

Now we can use the full decomposability of σ_l to infer $(Lab_E{\downarrow}_{P_1} \cup Lab_{E'}) \in U(P, F, L_F) = \mathcal{L}_\sigma(F)$, i.e. we have $E \cup E' \in \sigma(F)$. □

6 SCC-recursiveness

This section is devoted to the notion of SCC-recursiveness. We stick with the pattern of the previous section and develop abstract principles which ensure a relation to modularization. This time, however, the implication will be the other way round, i. e. we give a sufficient criterion for SCC-recursiveness rather than modularization. This is a more challenging (and admittedly more technical) endeavour since modularization is a quite common feature while SCC-recursiveness is rather demanding.

However, before developing this result, we close an open gap from (Dauphin, Rienstra, and van der Torre 2020b) regarding weakly preferred semantics: The paper investigates, among others, abstract property of weak admissibility-based semantics, but leaves open whether or not weakly preferred semantics satisfy SCC-recursiveness. We will answer this question in the affirmative.

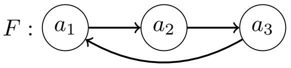
6.1 Weakly Preferred Semantics

Let us first consider an example illustrating why ad^w is not SCC-recursive (as already mentioned in (Dauphin, Rienstra, and van der Torre 2020b)) and to gain some intuition why this problematic mechanism does not apply to pr^w.

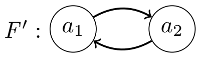
First, let us recall the recursive definition: $E \in ad^w(F)$ iff

1. $E \in cf(F)$ and
2. for any attacker y of E we have $y \notin \bigcup ad^w\left(F^E\right)$.

The problem with SCC-recursiveness is now that it is sometimes impossible to tell which attacks are meaningful (that is, coming from a weakly admissible extension of the reduct F^E) and which not. To illustrate this, let us quickly compare an even and an odd cycle and then move to a simple example consisting of just two SCCs. For F forming an odd cycle



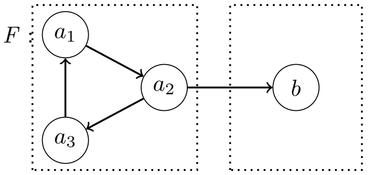
there is no non-empty weakly admissible extension: For example, set $E = \{a_1\}$. Then the reduct F^E consists of the unattacked argument a_3 attacking a_1; thus $E \notin ad^w(F)$. In contrast, the even cycle



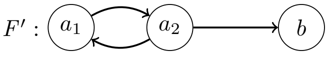
has two non-empty extensions $\{a_1\}$ and $\{a_2\}$; both are even stable and hence clearly weakly admissible as well.

The problem with SCC-recursiveness can now be seen as in the following example.

Example 6.1. Consider the following AF F, consisting of two SCCs:



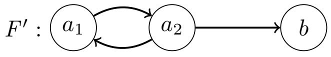
The only weakly admissible extension of the initial SCC is $\emptyset$, as we just saw. Regarding the second SCC, $\{b\}$ is an extension; however, b receives an attack from an undefeated argument a_2. In this case, b is acceptable since a_3 does not occur in any weakly admissible extension of $F^{\{b\}}$. However, if we turn the odd cycle into an even one



then suddenly $\{b\} \notin ad^w(F')$. However from the perspective of the second SCC the situation did not change: There is one argument with an input attack from an undecided argument a_2.

This is why ad^w is not SCC-recursive; but what is the catch for pr^w? To see the difference we characterize pr^w with (Baumann, Brewka, and Ulbricht 2020a, Theorem 4.5): $E \in pr^w(F)$ iff $E \in cf(F)$ and $\bigcup ad^w\left(F^E\right) = \emptyset$; that is, if E is weakly preferred, then *no* argument in the reduct F^E is weakly admissible. This means, the unpredictable behavior as illustrated in the previous example does not occur for pr^w. To see this, let us recall the example from the point of view of weakly preferred semantics.

Example 6.2. Given F as in the previous example, we start by considering the initial SCC: The only weakly preferred extension is $\emptyset$. So, moving to the second SCC we know that no argument in the first one is acceptable and hence we can be sure that $\{b\}$ is; thus $pr^w(F) = \{\{b\}\}$ can be inferred. In case of an even cycle



the two weakly preferred extensions of the initial SCC are $\{a_1\}$ and $\{a_2\}$; hence it is rightfully inferred that $pr^w(F') = \{\{a_1, b\}, \{a_2\}\}$.

Therefore, weakly preferred semantics are SCC-recursive for the same reason as stable semantics are: Arguments which are not in our extension are defeated (stb) resp. no threat (pr^w). The goal of the following considerations is to formalize this intuition.

We start by giving two auxiliary lemmata, the first is to establish the required connection between consideration of the reduct of an SCC and the set $F\downarrow_{UP_F(S,E)}$. Recall $UP_F(S,E) = \{a \in S \mid \nexists b \in E \setminus S : (b,a) \in R\}$.

Lemma 6.3. *Let $F = (A,R)$ be an AF, $E \subseteq A$ with $E \in cf(F)$ and $S \in SCCS_F$. Then*

$$\left(F\downarrow_{UP_F(S,E)}\right)^{E\cap S} = \left(F^E\right)\downarrow_{UP_F(S,E)}$$

Moreover, we need to be able to turn non-empty extensions of SCCs into non-empty extensions of the whole AF and vice versa. To illustrate this, recall the AF from Example 6.1: Here we see that F possesses some non-empty weakly admissible extension since the second SCC does ($\{b\}$). In this case, the initial SCC does not possess one, so $\{b\} \in ad^w(F)$. If there was an SCC possessing some weakly admissible argument attacking b, then we would move to this SCC and continue the argument inductively. For technical reasons, we need to formalize this for the reduct F^E for some extension E instead of F itself. This yields the following:

Lemma 6.4. *Let $F = (A,R)$ and let $E \subseteq A$. There is an SCC $S \in SCCS_F$ with $ad^w\left(\left(F^E\right)\downarrow_{UP_F(S,E)}\right) \neq \{\emptyset\}$ if and only if $ad^w\left(F^E\right) \neq \{\emptyset\}$.*

Now we follow Baroni et al, Section 5.2, where $\sigma = stb$ is considered, with adjustments to make it work for pr^w:

Proposition 6.5. *Let $F = (A,R)$ be an AF. Then $E \in pr^w(F)$ iff for any SCC $S \in SCCS_F$, $E \cap S \in pr^w(F\downarrow_{UP_F(S,E)})$.*

Proof. ($\Rightarrow$) Let $E \in pr^w(F)$. Let $S \in SCCS_F$.

(well-defined) It is clear that $E \cap S \subseteq UP_F(S,E)$ since otherwise, $E \notin cf(F)$. Thus $E \cap S$ in an extension in $F\downarrow_{UP_F(S,E)}$.

(cf) We have $E \cap S \in cf(F\downarrow_{UP_F(S,E)})$ due to $E \in cf(F)$.

(w-pref) Since E is w-preferred, F^E does not contain any w-admissible argument. Now assume $E \cap S$ is not w-preferred in $F\downarrow_{UP_F(S,E)}$. Since $E \cap S$ is conflict-free, this means there must be a non-empty w-admissible extension in $\left(F\downarrow_{UP_F(S,E)}\right)^{E\cap S}$. Hence by Lemma 6.3 there is a non-empty w-admissible extension in

$$\left(F\downarrow_{UP_F(S,E)}\right)^{E\cap S} = \left(F^E\right)\downarrow_{UP_F(S,E)} .$$

Thus by Lemma 6.4 there is a non-empty w-admissible extension in F^E, i. e. $E \notin pr^w(F)$; a contradiction.

($\Leftarrow$) We have to show that $E \in cf(F)$ and $ad^w\left(F^E\right) = \{\emptyset\}$. The former is clear since each argument is chosen among $UP_F(S,E)$. Furthermore, in each SCC F we have

$$\{\emptyset\} = ad^w\left(\left(F\downarrow_{UP_F(S,E)}\right)^{E\cap S}\right) = ad^w\left(\left(F^E\right)\downarrow_{UP_F(S,E)}\right)$$

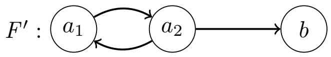
yielding $ad^w\left(F^E\right) = \{\emptyset\}$ by Lemma 6.4. □

6.2 Inferring SCC-recursiveness

Let us now continue with the aforementioned relation between modularization and SCC-recursiveness. The overall idea is that modularization allows to calculate extensions step-by-step. Starting with an initial SCC, say S_1, we consider an extension E_1 of $F\downarrow_{S_1}$, consider the reduct F^{E_1} and proceed analogously with the remaining parts of the AF. While this works from a quite high level point of view, we still have some work to do in order to get the details in place.

The first observation is that modularization as considered so far needs to be adjusted a bit, as the following example shall illustrate.

Example 6.6. Recall our AF



from above. Proceeding as described means we start with the initial SCC consisting of arguments a_1 and a_2. Take the complete extension $E = \{a_1\}$. Now one can already observe the following problem: While E is a complete extension of the initial SCC (which is fine for SCC recursiveness), it is *not* a complete extension of the whole AF since b is defended. However, modularization is only applicable if $E \in \sigma(F)$ is given.

This example is however no counterexample for admissible semantics; of course, $E = \{a_1\}$ is admissible in F since it does not matter that b is defended. Interestingly, in Section 3 we already utilized a property that could benefit from this behavior of admissibility, namely: Many semantics satisfy $E \cup E' \in \sigma(F)$ if $E \in ad(F)$ and $E' \in \sigma\left(F^E\right)$. This motivates the following notion of τ-modularization which will be mainly applied to $\tau = ad$.

Definition 6.7. A semantics σ satisfies *τ-modularization* if for any AF F we have: If $E \cap E' = \emptyset$ and $E \in \tau(F)$, then $E' \in \sigma\left(F^E\right)$ iff $E \cup E' \in \sigma(F)$.

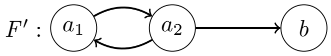
As already mentioned, many semantics possess this property for $\tau = ad$.

Proposition 6.8. *Each $\sigma \in \{ad, co, pr, stb, ss, il, eg\}$ satisfies ad-modularization.*

The case $\tau = ad$ suffices for our purpose since we will consider admissibility-based semantics only.

Next, we need to ensure that ad-modularization gives us extensions which are in a certain sense compatible with the structure at hand. To illustrate this, let us once again head to our running example.

Example 6.9. The initial SCC of the AF



possesses three admissible extensions $\emptyset$, $\{a_1\}$, and $\{a_2\}$. Starting with $\emptyset$, the reduct $F^{\emptyset}$ is F itself. Assuming that we are done with the initial SCC, in this situation b cannot be accepted due to the undecided attacker a_2.

It is clear that in the above situation, our semantics needs to take the input from the initial SCC into consideration and thus restricting the choices for the second one. The following notion formalizes this in a quite general fashion.

Definition 6.10. For some splitting (F_1, F_2, R_3) consider the set $D = \{a_2 \in A_2 \mid A_1 \not\rightarrow a_2\}$ of defendable arguments in F_2. We call σ f-splitting-compatible if there is some mapping $f_\sigma : \mathcal{F} \times 2^A \to 2^{2^A}$ satisfying

$$\{E \in \sigma(F) \mid E \subseteq A_2\} = f_\sigma(F_2, D).$$

Informally speaking, the extensions in F which are included in A_2 can be written as some function in the AF F_2 and the defendable arguments D. The underlying idea was already used in (Baroni, Giacomin, and Guida 2005) to prove SCC-recursiveness of Dung's classical AF semantics. Thus, the following examples of f-splitting-compatibility can already be inferred from (Baroni, Giacomin, and Guida 2005).

Example 6.11.

- For $\sigma = ad$ let $f_{ad}(F, D) = \{E \in \sigma(F) \mid E \subseteq D\}$.
- For $\sigma = co$ consider $\Gamma_F(E, D)$ with $\Gamma_F(E, D) = \{a \in A \cap D \mid E \text{ defends } a\}$. Then let

$$f_{co}(F, D) = \{E \in \sigma(F) \mid E = \Gamma_F(E, D)\}.$$

Analogous mappings can be found for pr and gr.

In the following, we will always assume that the set of all strongly connected components of any AF F, $SCCS_F = \{S_1, ..., S_m\}$, is ordered in such a way that each S_i is covered only after all its predecessors are covered, i. e. for all $S_i \in SCCS_F$, we have that for all $S_j \in S_i^{\prec}$, $j < i$ holds. With this, we make sure that the subframework $F{\downarrow}_{S_1,...,S_i}$ is always unattacked in F. The outline for our proof will be as follows: We will move along the SCCs of F using the order above. Whenever we encounter some S_i which is not initial, we will consider $(F^E){\downarrow}_{S_1,...,S_{i-1}}$ and $(F^{E \cap \{S_1,...,S_{i-1}\}}){\downarrow}_{S_i}$. If the latter subframework consists of only a single SCC, we will apply the base-function σ_b immediately. On the other hand, if we have more than one SCC in this subframework, we will order these in such a way, that we begin with an initial SCC of the subframework $UP_F(S_i, E)$, for which we can again use the base-function σ_b.

Theorem 6.12. *Let $F = (A, R)$ be an AF. Let $\sigma \subseteq ad$ be f-splitting compatible with mapping f_σ. If σ satisfies directionality and ad-modularization then σ is SCC-recursive.*

Proof. We set $\sigma_b(F, C) = f_\sigma(F, C)$. We first show that $E \in \sigma(F)$ implies $E \in \bar{\sigma}\left(F{\downarrow}_{UP_F(S,E)}, U_F(S, E)\right)$ for all $S \in SCCS_F$.

If $|SCCS_F| = 1$, use the splitting $((\emptyset, \emptyset), F, \emptyset)$. Thus we assume $|SCCS_F| > 1$. Let $S_i \in SCCS_F$ s. t. $S_i^{\prec} = \emptyset$. W. l. o. g. we consider $S_i = S_1$. Similar to before, we can now take the splitting $((\emptyset, \emptyset), F{\downarrow}_{S_1}, \emptyset)$. By directionality we get $E \cap S_1 \in \sigma(F{\downarrow}_{S_1})$, so we are done.

Now we assume that we are looking at some $S_i \in SCCS_F$ s. t. $S_i^{\prec} \neq \emptyset$. By the order of SCCs defined above, we have already considered all $S_j \in S_i^{\prec}$. Let $P = \bigcup_{S_j \in S_i^{\prec}} S_j$ be the set consisting of all ancestors of S_i. By directionality of σ, we can again infer $E' = E \cap (P \cup S_i) \in \sigma(F{\downarrow}_{P \cup S_i})$. Since $E \cap P \in \sigma(F{\downarrow}_P) \subseteq ad(F{\downarrow}_P)$ and as P is unattacked in $F{\downarrow}_{P \cup S_i}$, we also have $E \cap P \in ad(F{\downarrow}_{P \cup S_i})$. Now by ad-modularization, $E'' = (E' \cap S_i) \in \sigma\left((F^{E \cap P}){\downarrow}_{P \cup S_i}\right)$ holds.

Let $F' = F{\downarrow}_{UP_F(S_i,E)}$. For $|SCCS_{F'}| = k$, we now distinguish between the cases $k = 1$ and $k > 1$. For $k = 1$, we have to show that $E'' \in \sigma_b\left(F', U_F(S_i, E) \cap A\right)$. Take the splitting $F'' = (F_1, F_2, R_3)$ with $F_1 = (F^{E \cap P}){\downarrow}_P$, $F_2 = F'$ and $R_3 = (A(F_1) \times A(F_2)) \cap R$. Observe that we have $F'' = (F^{E \cap P}){\downarrow}_{P \cup S_i}$. The set of defendable arguments in F', $D = \{a_2 \in A(F') \mid A(F_1) \not\rightarrow a_2\}$, is exactly the set $U_F(S_i, E)$. Now

$$\begin{aligned}\sigma_b(F', U_F(S_i, E) \cap A) &= \sigma_b(F', U_F(S_i, E) \cap A)\\ &= f_\sigma(F', D)\\ &= \{\bar{E} \in \sigma(F'') \mid \bar{E} \subseteq A(F')\},\end{aligned}$$

therefore we have $E'' \in \sigma_b\left(F', U_F(S_i, E)\right)$.

Now suppose $k > 1$. In this case, we have left to show $E'' \cap S_{i_j} \in \bar{\sigma}\left(F'{\downarrow}_{UP_{F'}(S_{i_j},E'')}, U_{F'}(S_{i_j}, E'')\right)$ for all $S_{i_j} \in SCCS_{F'}$. For this, we can simply continue along the strongly connected components in $SCCS_{F'}$ using the order defined at the beginning. One can think of this as replacing S_i in the set $\{S_1, ..., S_i\}$ by $\{S_{i_1}, ..., S_{i_m}\}$, i. e. we are proving that $E \in \bar{\sigma}\left(F{\downarrow}_{UP_F(S,E)}, U_F(S, E)\right)$ for every $S \in \{S_1, ..., S_{i-1}, S_{i_1}, ..., S_{i_m}\}$. This can be achieved by the techniques we utilized above.

Next, we will show $E \in \bar{\sigma}(F, A) \Rightarrow E \in \sigma(F)$. For this, we again move along the strongly connected components of F to show that for each S_i and all its parent components, E is an extension of this particular subframework: The case where $|SCCS_F| = 1$ is trivial. Suppose $|SCCS_F| > 1$ and $S_i \in SCCS_F$ s. t. $S_i^{\prec} = \emptyset$. W. l. o. g. we assume that $S_i = S_1$. With the splitting $((\emptyset, \emptyset), F{\downarrow}_{S_1}, \emptyset)$, we see that $E \cap S_1 \in \sigma(F{\downarrow}_{S_1})$.

Now suppose that $S_i \in SCCS_F$ s. t. $S_i^{\prec} \neq \emptyset$ and let $|SCCS_{F{\downarrow}_{UP_F(S_i,E)}}| = 1$. For $P = \bigcup_{S_j \in S_i^{\prec}} S_j$, we have $E \cap P \in \sigma(F{\downarrow}_P) \subseteq ad(F{\downarrow}_P)$. Since P is unattacked in F (by the order of elements in $\{S_1, ..., S_i\}$), we know that $E \cap P \in ad(F{\downarrow}_{P \cup S_i})$. Let $F_1 = (F^{E \cap P}){\downarrow}_P$ and let $F_2 = (F^{E \cap P}){\downarrow}_{S_i} = F{\downarrow}_{UP_F(S_i,E)}$. For the splitting $F' = (F_1, F_2, (A(F_1) \times A(F_2)) \cap R)$, we can see that the set $D = \{a_2 \in A(F_2) \mid A(F_1) \not\rightarrow a_2\}$ of defendable arguments in F_2 is exactly $U_F(S_i, E)$. Thus, by $E \cap S_i \in \sigma_b(UP_F(S_i, E), U_F(S_i, E))$, we can infer that $E \cap S_i \in \sigma(F')$. Now we can use ad-modularization of σ to find $(E \cap P) \cup (E \cap S_i) = E \cap (P \cup S_i) \in \sigma(F{\downarrow}_{P \cup S_i})$.

For $|SCCS_{F_{UP_F(S_i,E)}}| > 1$ let $F' = F{\downarrow}_{UP_F(S_i,E)}$ and $E_i = E \cap S_i$. Suppose $SCCS_{F'} = \{S_{i_1}, ..., S_{i_m}\}$ We know that for each SCC $S_{i_j} \in \{S_{i_1}, ..., S_{i_m}\}$, it holds that $E_i \cap S_{i_j} \in \bar{\sigma}\left(F'{\downarrow}_{UP_{F'}(S_{i_j},E_i)}, U_{F'}(S_{i_j}, E_i)\right)$, Now we again choose some initial S_{i_j} in F', i. e. we assume $S_{i_j}^{\prec} = \emptyset$.

Since P is unattacked in F and S_{i_j} is unattacked in F', we infer that $P \cup S_{i_j}$ is unattacked in $F{\downarrow}_{P \cup \{S_{i_1} \cup \ldots \cup S_{i_m}\}}$. From this we get $E \cap P \in ad(F{\downarrow}_{P \cup S_{i_j}})$. For $F_1 = (F^{E \cap P}){\downarrow}_P$ and $F_2 = (F^{E \cap P}) {\downarrow}_{S_{i_j}} = F' {\downarrow}_{UP_{F'}(S_{i_j}, E_i)}$ consider the splitting $F'' = (F_1, F_2, (A(F_1) \times A(F_2)) \cap R)$. Again, we have that the set $D = \{a_2 \in A(F_2) \mid A(F_1) \not\rightarrow a_2\}$ of defendable arguments in F_2 is exactly $U_{F'}(S_{i_j}, E_i)$. Thus we can infer that $(E \cap P) \cup (E_i \cap S_{i_j}) = E \cap (P \cup S_{i_j}) \in \sigma(F {\downarrow}_{P \cup S_{i_j}})$ by ad-modularization of σ. As before, this part of the proof amounts to replacing S_i in $\{S_1, ..., S_i\}$ by the set $\{S_{i_1}, ..., S_{i_m}\}$, i.e. we have now shown that $E \cap \{S_1 \cup ... \cup S_{i-1} \cup S_{i_1} \cup ... \cup S_{i_m}\}$ is a σ-extension of $F{\downarrow}_{S_1, \ldots, S_{i-1}, S_{i_1}, \ldots, S_{i_m}}$. Note that this subframework of F is missing $D_F(S_i, E)$. However, since $\bigcup_{j=1}^{i-1} S_j = P$, we now know that $E \cap \{S_{i_1} \cup ... \cup S_{i_m}\}$ is an extension of $\big(F^{E \cap P}\big){\downarrow}_{P \cup S_{i_1} \cup \ldots \cup S_{i_m}} = \big(F^{E \cap P}\big){\downarrow}_{P \cup S_i}$, i.e. we can infer $E \cap \{P \cup S_{i_1} \cup ... \cup S_{i_m}\} = E \cap \{P \cup S_i\} \in \sigma(F{\downarrow}_{P \cup S_i})$ by ad-modularization of σ. $\square$

7 Conclusion and Future Work

In this paper, we compared the recently introduced modularization property to established modularity notions from the literature. Thereby, we showed that the notions are incomparable in general. Continuing existing research on the relationships, we developed abstract criteria to infer implications between i) full decomposability and modularization as well as ii) SCC recursiveness and modularization.

Due to space restrictions, our investigation only covered two semantics based on weak admissibility, whereas also covering weakly grounded and weakly complete as well as the more recently introduced qualitative and semi-qualitative semantics from (Dauphin, Rienstra, and van der Torre 2020a) is a natural future work direction. In addition, including more principles from the literature, e.g. (van der Torre and Vesic 2017) would broaden our investigation.

We would also like to find further relations and milder or more intuitive assumptions for the ones reported so far.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig and by the Austrian Science Fund (FWF) through project Y698.

References

Baroni, P.; Boella, G.; Cerutti, F.; Giacomin, M.; Van Der Torre, L.; and Villata, S. 2014. On the input/output behavior of argumentation frameworks. *Artificial Intelligence* 217:144–197.

Baroni, P.; Caminada, M.; and Giacomin, M. 2011. An introduction to argumentation semantics. *The knowledge engineering review* 26(4):365–410.

Baroni, P.; Caminada, M.; and Giacomin, M. 2018. Abstract argumentation frameworks and their semantics. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 4.

Baroni, P.; Giacomin, M.; and Guida, G. 2005. Scc-recursiveness: a general schema for argumentation semantics. *Artif. Intell.* 168(1-2):162–210.

Baroni, P.; Giacomin, M.; and Liao, B. 2018. Locality and modularity in abstract argumentation. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 19.

Baumann, R.; Brewka, G.; and Ulbricht, M. 2020a. Comparing weak admissibility semantics to their dung-style counterparts - reduct, modularization, and strong equivalence in abstract argumentation. In *Proc. KR*, 79–88.

Baumann, R.; Brewka, G.; and Ulbricht, M. 2020b. Revisiting the foundations of abstract argumentation: Semantics based on weak admissibility and weak defense. In *Proc. AAAI*, 2742–2749. AAAI Press.

Baumann, R.; Linsbichler, T.; and Woltran, S. 2016. Verifiability of argumentation semantics. *CoRR* abs/1603.09502.

Baumann, R. 2011. Splitting an argumentation framework. In *Proc. LPNMR*, 40–53. Springer.

Caminada, M., and Dunne, P. 2019. Strong admissibility revisited: Theory and applications. *Argument and Computation* 1–24.

Dauphin, J.; Rienstra, T.; and van der Torre, L. 2020a. A principle-based analysis of weakly admissible semantics. In Prakken, H.; Bistarelli, S.; Santini, F.; and Taticchi, C., eds., *Computational Models of Argument - Proceedings of COMMA 2020*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, 167–178. IOS Press.

Dauphin, J.; Rienstra, T.; and van der Torre, L. 2020b. A principle-based analysis of weakly admissible semantics. *Computational Models of Argument-Proceedings of COMMA 2020, Perugia Italy, September 4-11, 2020* 167–178.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–357.

Dvořák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. also appears in IfCoLog Journal of Logics and their Applications 4(8):2623–2706.

Lifschitz, V., and Turner, H. 1994. Splitting a logic program. In Hentenryck, P. V., ed., *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, 23–37. MIT Press.

Turner, H. 1996. Splitting a default theory. In Clancey, W. J., and Weld, D. S., eds., *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, 645–651. AAAI Press / The MIT Press.

van der Torre, L., and Vesic, S. 2017. The principle-based approach to abstract argumentation semantics. *FLAP* 4(8).

Limits and Possibilities of Forgetting in Abstract Argumentation

Ringo Baumann and **Matti Berthold**
Leipzig University, Department of Computer Science
{baumann,berthold}@informatik.uni-leipzig.de

Abstract

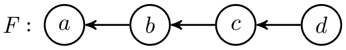
The topic of *forgetting* has been extensively studied in the field of knowledge representation and reasoning for many major formalisms. Quite recently it has been introduced to abstract argumentation. However, many already known as well as essential aspects about forgetting like *strong persistence* or *strong invariance* have been left unconsidered. Moreover, we show that forgetting in abstract argumentation cannot be reduced to forgetting in logic programming. In addition, we deal with the more general problem of forgetting whole sets of arguments and show that iterative application of existing operators for single arguments does not necessarily yield a desirable result as it may not produce an informationally economic argumentation framework. As a consequence we provide a systematic and exhaustive study of forgetting desiderata and associated operations adapted to the intrinsics of abstract argumentation. We show the limits and shed light on the possibilities.

1 Introduction

The notion of *forgetting* has been extensively studied in the field of knowledge representation and reasoning for many major formalisms like classical logic (Lin and Reiter 1994), logic programming (Gonçalves, Knorr, and Leite 2016a; Eiter and Kern-Isberner 2018) and more recently for abstract argumentation (Baumann, Gabbay, and Rodrigues 2020). Roughly speaking, forgetting is about getting rid of some variables, atoms or arguments while keeping as much as possible of the reasoning not concerned with the forgotten. The ability of forgetting is often exploited to make reasoning more efficient. In this paper we want to further elaborate the limits and possibilities of forgetting in abstract argumentation. The latter is a vibrant research area in AI (Simari and Rahwan 2009; Baroni et al. 2018) with Dung-style argumentation frameworks (AFs) and their associated semantics at the heart of this field (Dung 1995) .

In order to obtain reasonable forgetting operators for abstract reasoning we may try to convey ideas from other formalisms. The area of logic programming with its plenty of approaches to forgetting is a good candidate (cf. (Gonçalves, Knorr, and Leite 2016b) for an excellent overview). However, the following two examples show that forgetting in abstract argumentation cannot be reduced to forgetting in logic programming in a straightforward manner.

Example 1 (Limits of the Standard Translation)**.** *Consider the following AF F. We observe $stb(F) = \{\{b,d\}\}$. Assume now that we want to forget the argument b. Note that simply deleting b would yield an AF F_b, s.t. $stb(F_b) = \{\{a,d\}\}$. This means, such a syntactical removal would render the previously unaccepted argument a acceptable.*



Let us consider instead the standard translation from AFs to LPs (Strass 2013). This yields the following equivalent logic program P.

$$P: \quad a \leftarrow not\, b \qquad b \leftarrow not\, c$$
$$c \leftarrow not\, d \qquad d$$

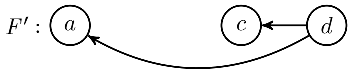
Now we may apply the already defined forgetting operator f_{SP} (Berthold et al. 2019b). More precisely, forgetting b from P results in $\mathsf{f}_{SP}(P,b)$ as given below.

$$\mathsf{f}_{SP}(P,b): \quad a \leftarrow not\, not\, c \qquad c \leftarrow not\, d \qquad d$$

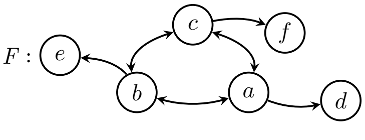
Unfortunalety, $\mathsf{f}_{SP}(P,b)$ is a non-AF-like program. Therefore, it is generally not possible to simply reverse the standard translation. However, in case of $\mathsf{f}_{SP}(P,b)$ we may find an equivalent LP P' which is indeed AF-like.

$$P': \quad a \leftarrow not\, d \qquad c \leftarrow not\, d \qquad d$$

Retranslating P' to the realm of AFs results in F'. Note that $stb(F) = \{\{d\}\}$ as desired.



Example 2 (Representational Limits)**.** *Consider now the slightly more involved AF F. We observe $stb(F) = \{\{a,e,f\},\{b,f,d\},\{c,d,e\}\}$.*

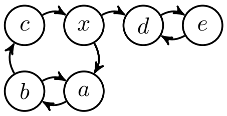


Let us assume again that we want to forget the argument b. The favored forgetting result is thus the extension set $D = \{\{a,e,f\},\{f,d\},\{c,d,e\}\}$. *Since* D *forms a* $\subseteq$*-antichain there is an LP* P *realizing it (Eiter et al. 2013). However, we will never find an equivalent AF-like LP* P' *since* D *does not satisfies so-called tightness (Dunne et al. 2015). In particular,* $\{f,d\}\cup\{e\}\notin D$ *but* $\{e,f\}\subseteq\{a,e,f\}$ *and* $\{d,e\}\subseteq\{c,d,e\}$.

The final example deals with already existing forgetting operators in abstract argumentation. It shows that forgetting multiple arguments cannot be simply reduced to forgetting single arguments as this does not necessarily yield desirable results.

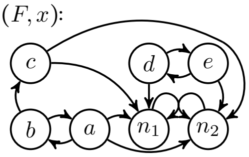
Example 3 (Forgetting Sets vs. Arguments). *Consider the following AF* F. *We have* $stb(F) = \{\{x,b,e\},\{a,c,d\},\{a,c,e\}\}$. *Assume that we want to forget a set of arguments, say* $\{x,b\}$. *One reasonable forgetting result is thus an AF* F', *s.t.* $stb(F') = \{\{a,c,d\},\{a,c,e\}\} = D$.

F:

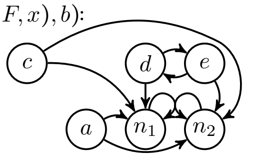


One natural approach for forgetting multiple arguments is to iteratively apply an existing forgetting operator for single arguments. The following frameworks illustrate this procedure for the operator f *firstly presented in (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1, Example 4).*

$\mathsf{f}(F,x)$:

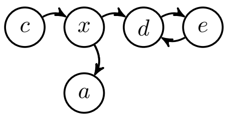


$\mathsf{f}(\mathsf{f}(F,x),b)$:

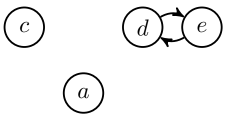


If forgetting b *first and subsequently* x *reveals that this approach is sensitive to the order of forgetting and might not yield an informationally economic result.*

$\mathsf{f}(F,b)$:



$\mathsf{f}(\mathsf{f}(F,b),x)$:



The three examples above show that we need further investigation on how sets of arguments can be forgotten in case of AFs. As a consequence we provide a systematic and comprehensive analysis of forgetting desiderata and associated operations adapted to the intrinsics of abstract argumentation. We hereby draw a lot of inspiration from logic programming. We show the limits and shed light on the possibilities. In particular, we study the relations between desiderata, their individual as well as combined satisfiability and look for promising combinations. Moreover, we consider forgetting under stable semantics as it shows a quite different behaviour regarding the fulfillment of combined desiderata. Finally, we conclude and discuss related work.

2 Background

Logic Programming

Syntax and Semantics We assume a *propositional signature* $\mathcal{U}$. A *logic program* P over $\mathcal{U}$ (Lifschitz, Tang, and Turner 1999) is a finite set of *rules* of the form $a_1 \vee \ldots \vee a_k \leftarrow b_1,...,b_l,\ not\,c_1,...,not\,c_m,\ not\,not\,d_1,...,not\,not\,d_n$. For such a rule r let $H(r) = \{a_1,\ldots a_k\}$, $B^+(r) = \{b_1,\ldots,b_l\}$, $B^-(r) = \{c_1,\ldots,c_m\}$ and $B^{--}(r) = \{d_1,\ldots,d_n\}$. We define $\mathcal{U}(P) = \bigcup_{r\in P} H(r)\cup B^+(r)\cup B^-(r)\cup B^{--}(r)$.

Given a program P over $\mathcal{U}$ and a set of atoms $I\subseteq\mathcal{U}$, a so-called *interpretation*, the *reduct* of P w.r.t. I, is defined as $P^I = \{H(r)\leftarrow B^+(r)\mid r\in P, B^-(r)\cap I=\varnothing, B^{--}(r)\subseteq I\}$. An interpretation I is an *answer set* of P if $I\vDash P$, and for each interpretation I' we have: If $I'\vDash P^I$, then $I'\not\subset I$. The set of all answer sets of P is denoted by $\mathcal{AS}(P)$. We say that two programs P_1,P_2 are *equivalent* if $\mathcal{AS}(P_1)=\mathcal{AS}(P_2)$ and *strongly equivalent*, denoted by $P_1\equiv P_2$, if $\mathcal{AS}(P_1\cup R)=\mathcal{AS}(P_2\cup R)$ for any program R (Lifschitz, Pearce, and Valverde 2001). Given a set $V\subseteq\mathcal{U}$, the V*-exclusion* of a set of answer sets $\mathcal{M}$, denoted $\mathcal{M}_{\|V}$, is $\{X\backslash V\mid X\in\mathcal{M}\}$.

Forgetting: Desiderata and Operators Let $\mathcal{P}$ be the set of all logic programs. A *forgetting operator* is a (partial) function $\mathsf{f}:\mathcal{P}\times 2^{\mathcal{U}}\rightarrow\mathcal{P}$ with $(P,V)\mapsto\mathsf{f}(P,V)$. The program $\mathsf{f}(P,V)$ is interpreted as the *result of forgetting about* V *from* P. Moreover, $\mathcal{U}(\mathsf{f}(P,V))\subseteq\mathcal{U}(P)\smallsetminus V$ is usually required. In the following we introduce some well-known properties for forgetting operators (Gonçalves, Knorr, and Leite 2016a).

Strong persistence is presumbly the best known one (Knorr and Alferes 2014). It requires that the result of forgetting $\mathsf{f}(P,V)$ is strongly equivalent to the original program P, modulo the forgotten atoms.

(SP) f satisfies *strong persistence* if, for each program P and each set of atoms V, we have: $\mathcal{AS}(\mathsf{f}(P,V)\cup R) = \mathcal{AS}(P\cup R)_{\|V}$ for all programs R with $\mathcal{U}(R)\subseteq\mathcal{U}\backslash V$.

Strong invariance requires that rules not mentioning atoms to be forgotten can be added before or after forgetting.

(SI) f satisfies *strong invariance* if, for each program P and each set of atoms V, we have: $\mathsf{f}(P,V)\cup R\equiv\mathsf{f}(P\cup R,V)$ for all programs R with $\mathcal{U}(R)\subseteq\mathcal{U}\backslash V$.

Consequence persistence and its two variations are weaker forms of strong persistence dealing with ordinary equivalence only.

(CP) f satisfies *consequence persistence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V)) = \mathcal{AS}(P)_{\|V}$.

(wC) f satisfies *strengthened consequence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V))\subseteq\mathcal{AS}(P)_{\|V}$.

(sC) f satisfies *weakened consequence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V))\supseteq\mathcal{AS}(P)_{\|V}$.

Note that the presented desiderata are often considered for certain subclasses like *disjunctive*, *normal* or *Horn programs*. Sometimes forgetting properties are also considered relativized to concrete forgetting instances (Berthold et al. 2019b).

Argumentation Theory

Syntax and Semantics Let $\mathcal{U}$ be an infinite background set. An *abstract argumentation framework (AF)* (Dung 1995) is a directed graph $F = (A, R)$ with $A \subseteq \mathcal{U}$ representing arguments and $R \subseteq A \times A$ interpreted as attacks. If $(a, b) \in R$ we say that a *attacks* b or a is *an attacker of* b. Moreover, a set E *defends* an argument a if any attacker of a is attacked by some argument of E. In this paper we consider finite AFs only and use the symbol $\mathcal{F}$ to denote the set of all finite AFs. Moreover, for a set $E \subseteq A$ we use E^+ for $\{b \mid (a, b) \in R, a \in E\}$ and define $E^\oplus = E \cup E^+$. Given an AF $F = (B, S)$, we use $A(F)$ to refer to the set B and $R(F)$ to refer to the relation S. For two AFs F and G, we define the expansion of F by G, in symbols $F \sqcup G$, as expected: $F \sqcup G = (A(F) \cup A(G), R(F) \cup R(G))$. Finally, the restriction of an AF F to a set of arguments $C \subseteq \mathcal{U}$ is defined as $F|_C = (A(F) \cap C, R(F) \cap (C \times C))$.

An *extension-based semantics* $\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}}$ is a function which assigns to any AF F a set of sets of arguments $\sigma(F) \subseteq 2^{A(F)}$. Each set of arguments $E \in \sigma(F)$ is considered to be acceptable with respect to F and is called a *σ-extension*. The most basic requirements of an extension are called *conflict-freeness* (cf) and *admissibility* (ad). Other well-studied semantics include stage (stg), stable (stb), semi-stable (ss), complete (co), preferred (pr), grounded (gr), ideal (il) and eager (eg). The requirements of each semantics are summarised below. A recent overview of argumentation semantics can be found in (Baroni, Caminada, and Giacomin 2018).

Definition 1. *Let $F = (A, R)$ be an AF and $E \subseteq A$.*

1. *$E \in cf(F)$ iff for no $a, b \in E$, $(a, b) \in R$,*
2. *$E \in ad(F)$ iff $E \in cf(F)$ and E defends all its elements,*
3. *$E \in co(F)$ iff $E \in ad(F)$ and for any $a \in A$ defended by E, $a \in E$,*
4. *$E \in stg(F)$ iff $E \in cf(F)$ and for no $\mathcal{I} \in cf(F)$, $E^\oplus \subset \mathcal{I}^\oplus$,*
5. *$E \in stb(F)$ iff $E \in cf(F)$ and $E^\oplus = A$,*
6. *$E \in ss(F)$ iff $E \in ad(F)$ and for no $\mathcal{I} \in ad(F)$, $E^\oplus \subset \mathcal{I}^\oplus$,*
7. *$E \in pr(F)$ iff $E \in co(F)$ and for no $\mathcal{I} \in co(F)$, $E \subset \mathcal{I}$,*
8. *$E \in gr(F)$ iff $E \in co(F)$ and for any $\mathcal{I} \in co(F)$, $E \subseteq \mathcal{I}$,*
9. *$E \in il(F)$ iff $E \in co(F)$, $E \subseteq \bigcap pr(F)$ and there is no $\mathcal{I} \in co(F)$ satisfying $\mathcal{I} \subseteq \bigcap pr(F)$ s.t. $E \subset \mathcal{I}$,*
10. *$E \in eg(F)$ iff $E \in co(F)$, $E \subseteq \bigcap ss(F)$ and there is no $\mathcal{I} \in co(F)$ satisfying $\mathcal{I} \subseteq \bigcap ss(F)$ s.t. $E \subset \mathcal{I}$.*

Existence, Reasoning and Expressibility A semantics σ is *universally defined*, if $\sigma(F) \neq \emptyset$ for any $F \in \mathcal{F}$. If even $|\sigma(F)| = 1$ we say that σ is *uniquely defined*. Apart from stable semantics all considered semantics are universally defined. The grounded, ideal and eager semantics are uniquely defined (cf. (Baumann and Spanring 2015) for an overview).

With respect to the acceptability of arguments, we consider the two main reasoning modes. Given a semantics σ, an AF F, and an argument $a \in A(F)$, we say that a is *credulously accepted w.r.t.* σ if $a \in \bigcup \sigma(F)$ and that a is *skeptically accepted w.r.t.* σ if $\sigma(F) \neq \emptyset$ and $a \in \bigcap \sigma(F)$. In case of stable semantics a collapse is possible, i.e. $stb(F) = \emptyset$ for some F. From basic set theory we know that in this case all arguments $x \in A(F)$ are skeptically accepted. To get around this cornercase we redefine the intersection over the empty extension set as $\bigcap stb(F) = \emptyset$.

We say that a set of sets $\mathcal{E} \subseteq 2^{\mathcal{U}}$ is *realizable w.r.t. a semantics* σ if there is an AF F s.t. $\sigma(F) = \mathcal{E}$. Realizability under stable semantics is given if and only if i) $\mathcal{E}$ forms a $\subseteq$-antichain[1] and ii) $\mathcal{E}$ is *tight* (Dunne et al. 2015). Tightness if fulfilled if for all $E \in \mathcal{E}$ and $a \in \bigcup \mathcal{E}$ we have: if $E \cup \{a\} \notin \mathcal{E}$ then there exists an $e \in E$, s.t. $(a, e) \notin \{(b, c) \mid \exists E' \in \mathcal{E} : \{b, c\} \subseteq E'\}$. See Example 2 for an illustration. Moreover, we will frequently use that stage, semi-stable as well as preferred semantics satisfy I-maximality too (cf. (Baumann 2018) for an overview).

3 Desiderata for Forgetting

Given an AF F and a set of arguments $X \subseteq \mathcal{U}$, we use $\mathsf{f}_\sigma(F, X)$ to denote the *result of forgetting the arguments X in F under semantics σ*. This means, we consider a function $\mathsf{f}_\sigma : \mathcal{F} \times 2^{\mathcal{U}} \to \mathcal{F}$ mapping a pair (F, X) to an AF $\mathsf{f}_\sigma(F, X)$. If clear from context or irrelevant we will omit σ.

In the following we collect and define a large number of desiderata for forgetting in abstract argumentation. Some of them have been already considered in (Baumann, Gabbay, and Rodrigues 2020) for the case of single arguments. We generalize them to sets of arguments as done in the LP case. Moreover we introduce further important conditions firstly considered in the realm of LPs (Gonçalves, Knorr, and Leite 2016a). We will see that there are many dependencies that are not clear at first glance. Note that desiderata e_1 as well as e_2 could be alternatively renamed as $e_{\mathbf{CP}}$ and $e_{\mathbf{SP}}$ (see Section 2 for more details.) However, we decided to keep in line with the notation chosen in (Baumann, Gabbay, and Rodrigues 2020).

Desiderata 1. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator f we define:*

e_1. $\sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$
(X-adjusted extension)

$e_{\mathbf{wC}}$. $\sigma(\mathsf{f}(F, X)) \supseteq \{E \setminus X \mid E \in \sigma(F)\}$
(no such extension is lost)

$e_{\mathbf{sC}}$. $\sigma(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$
(no further extensions are added)

e_2. *$\sigma(\mathsf{f}(F, X) \sqcup H) = \{E \setminus X \mid E \in \sigma(F \sqcup H)\}$ for any H with $A(H) \subseteq \mathcal{U} \setminus X$*
(delete X even from any future extension)

$e_{3_\subseteq}$. *$\sigma(\mathsf{f}(F, X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \setminus X$*
(subsets of X-adjusted extension)

$e_{3_\supseteq}$. *$\sigma(\mathsf{f}(F, X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \supseteq E \setminus X$*
(supersets of X-adjusted extension)

e_4. $\sigma(\mathsf{f}(F, X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \emptyset\}$
(remove X-overlapping extensions)

[1] Within the argumentation community this property is usually referred to as *I-maximality* (Baroni and Giacomin 2007).

The next four desiderata are concerned with skeptical and credulous reasoning.

Desiderata 2. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

r_1. $\bigcap\sigma(\mathsf{f}(F,X)) \cap X = \varnothing$ *(X is not skept. accepted)*

r_2. $\bigcup\sigma(\mathsf{f}(F,X)) \cap X = \varnothing$ *(X is not cred. accepted)*

r_3. $\bigcap\sigma(\mathsf{f}(F,X)) = (\bigcap\sigma(F)) \setminus X$ *(rigid skept. acceptance)*

r_4. $\bigcup\sigma(\mathsf{f}(F,X)) = (\bigcup\sigma(F)) \setminus X$ *(rigid cred. acceptance)*

Arguably the presented reasoning desiderata either describe to strictly or too loosely what is skeptically or credulously accepted. For r_1 and r_2 to be satisfied, it suffices to syntactically remove X. In contrast, to satisfy r_3 or r_4 the resulting AF must entail a precise set of arguments. As a compromise between them, we suggest the following desiderata, that bridge semantic and syntatic requirements.

Desiderata 3. *Given two AFs F and H as well as a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

m_1. $\bigcap\sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$
(skept. acceptance is among unforgotten old arguments)

m_2. $\bigcup\sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$
(cred. acceptance is among unforgotten old arguments)

m_3. $\bigcap\sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$ *for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$*
(forgotten arguments are never skept. accepted)

m_4. $\bigcup\sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$
for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$
(forgotten arguments are never cred. accepted)

Condition m_1 (resp. m_2) requires that, if there are new arguments added while forgetting, they be irrelevant to skeptical (resp. credulous) reasoning. In other words, that these arguments are purely administrative. Then m_3 (resp. m_4) require new arguments to be irrelevant, even under the addition of new information.

The following three conditions are purely syntactical ones. Desideratum s_1 makes explicit what is often implicitly assumed for forgetting operators in other formalisms. Condition s_3 presents the most straightforward way of forgetting a set of arguments. Such an syntactical approach was firstly considered in (Bisquert et al. 2011).

Desiderata 4. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

s_1. $A(\mathsf{f}(F,X)) \cap X = \varnothing$ *(no arguments from X)*

s_2. $A(\mathsf{f}(F,X)) = A(F) \setminus X$ *(precise set of arguments)*

s_3. $\mathsf{f}(F,X) = F|_{A(F)\setminus X}$ *(rigid AF)*

The following vacuity desiderata provide conditions under which a given framework does not require any changes.

Desiderata 5. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

v_1. *If $\bigcap\sigma(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$.* *(skept. vacuity)*

v_2. *If $\bigcup\sigma(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$.* *(cred. vacuity)*

v_3. *If $A(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$.* *(argument vacuity)*

When deriving a forgetting result it would be advantageous to be able to confine the construction in some way. For comparison, some forgetting operators in LP have been shown to be able to disregard rules that do not mention the atoms to be forgotten, i.e. they satisfy the discussed property $(\mathbf{SI})$. Similarly, when forgetting arguments from an AF we could require that arguments that do not stand in (close) contact to the arguments to be forgotten can be left unchanged.

Desiderata 6. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

l_0. $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ *for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$* *($\mathsf{f}$ and $\sqcup$ are compatible)*

l_1. $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ *for all AFs H with $A(H) \subseteq \mathcal{U} \setminus (X \cup \{a \mid \exists x \in X,$ s.t. $(a,x) \in R$ or $(x,a) \in R\})$*
(less tolerant refinement of compatibility)

We proceed with an analysis of their dependencies.

Proposition 1. *For $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and conditions c and c' in the diagram below, a path from c to c' indicates that any function f_σ satisfying c under σ also satisfies c' under σ. Moreover, only these relations hold.*

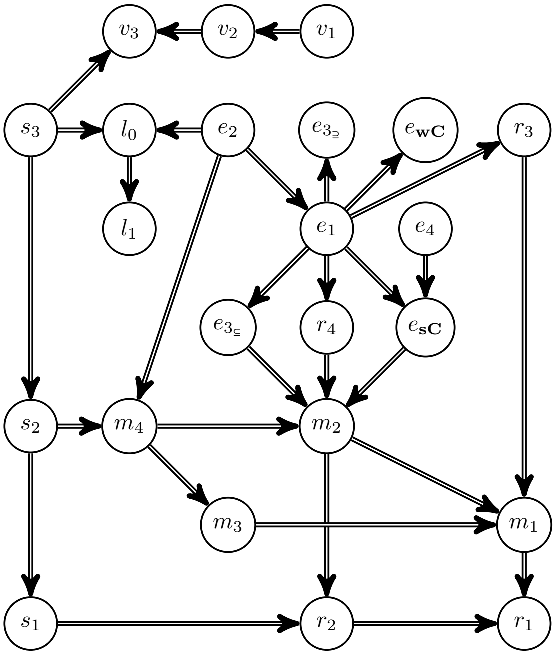


Figure 1: Dependencies

Proof: In the following we show all valid relations.

- $e_1 \Rightarrow r_3$: $\bigcap\sigma(\mathsf{f}(F,X)) = \bigcap\{E \setminus X \mid E \in \sigma(F)\} = (\bigcap\{E \mid E \in \sigma(F)\}) \setminus X = (\bigcap\sigma(F)) \setminus X$
- $e_1 \Rightarrow r_4$: $\bigcup\sigma(\mathsf{f}(F,X)) = \bigcup\{E \setminus X \mid E \in \sigma(F)\} = (\bigcup\{E \mid E \in \sigma(F)\}) \setminus X = (\bigcup\sigma(F)) \setminus X$
- $e_1 \Rightarrow e_{\mathbf{sC}}, e_{\mathbf{wC}}$: Obviously, $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$ implies $\sigma(\mathsf{f}(F,X)) \circ \{E \setminus X \mid E \in \sigma(F)\}$ for each $\circ \in \{\subseteq, \supseteq\}$.

- $e_1 \Rightarrow e_{3_\supseteq}, e_{3_\subseteq}$: $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$ implies $T(E) = E \setminus X$ and thus, $T(E) \circ E \setminus X$ for each $\circ \in \{\subseteq, \supseteq\}$.
- $e_4 \Rightarrow e_{\mathbf{sC}}$: We have $\sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\} \subseteq \{E \setminus X \mid E \in \sigma(F)\}$.
- $e_{\mathbf{sC}} \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \subseteq \bigcup\{E \setminus X \mid E \in \sigma(F)\} \subseteq A(F) \setminus X$ since for each extension $E \in \sigma(F)$, $E \subseteq A(F)$ is implied.
- $e_2 \Rightarrow e_1$: Consider $H_\varnothing = (\varnothing, \varnothing)$.
- $e_2 \Rightarrow m_4$: Let H be an AF with $A(H) \subseteq \mathcal{U} \setminus X$. We have: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) = \bigcup\{E \setminus X \mid E \in \sigma(F \sqcup H)\} \subseteq (A(H) \cup A(F)) \setminus X$.
- $e_2 \Rightarrow l_0$: $\sigma(\mathsf{f}(F,X) \sqcup H) = \{E \setminus X \mid E \in \sigma(F \sqcup H)\} = \{E \setminus X \mid E \in \sigma(F \sqcup H \sqcup \varnothing)\} = \sigma(\mathsf{f}(F \sqcup H, X) \sqcup \varnothing) = \sigma(\mathsf{f}(F \sqcup H, X))$
- $l_0 \Rightarrow l_1$: Obviously, $\mathcal{U} \setminus (X \cup \{a \mid \exists x \in X,$ s.t. $(a,x) \in R$ or $(x,a) \in R\}) \subseteq \mathcal{U} \setminus X$
- $r_3 \Rightarrow m_1$: $\bigcap \sigma(\mathsf{f}(F,X)) = (\bigcap \sigma(F)) \setminus X \subseteq A(F) \setminus X$
- $r_4 \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) = (\bigcup \sigma(F)) \setminus X \subseteq A(F) \setminus X$
- $m_3 \Rightarrow m_1$ and $m_4 \Rightarrow m_2$: Consider $H_\varnothing$.
- $m_4 \Rightarrow m_3$: $\bigcap \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq \bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$
- $m_2 \Rightarrow m_1$: $\bigcap \sigma(\mathsf{f}(F,X)) \subseteq \bigcup \sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$
- $m_1 \Rightarrow r_1$ and $m_2 \Rightarrow r_2$: Obvious since $(A(F) \setminus X) \cap X = \varnothing$.
- $s_3 \Rightarrow s_2$: $A(\mathsf{f}(F,X)) = A(F|_{A(F) \setminus X}) = A(F) \setminus X$.
- $s_3 \Rightarrow v_3$: Let $A(F) \cap X = \varnothing$. Consequently, $\mathsf{f}(F,X) = F|_{A(F) \setminus X} = F$.
- $s_3 \Rightarrow l_0$: We have $\mathsf{f}(F,X) \sqcup H = F|_{A(F) \setminus X} \sqcup H = F|_{A(F) \setminus X} \sqcup H|_{A(H) \setminus X}$ since $A(H) \subseteq \mathcal{U} \setminus X$ is assumed. Consequently, $\mathsf{f}(F,X) \sqcup H = F \sqcup H|_{A(F \sqcup H) \setminus X} = \mathsf{f}(F \sqcup H, X)$ implying $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$.
- $s_2 \Rightarrow s_1$: $A(\mathsf{f}(F,X)) \cap X = (A(F) \setminus X) \cap X = \varnothing$.
- $s_2 \Rightarrow m_4$: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq A(\mathsf{f}(F,X) \sqcup H) = A(\mathsf{f}(F,X)) \cup A(H) = (A(F) \setminus X) \cup A(H) = (A(H) \cup A(F)) \setminus X$ since for the AF H we have $A(H) \subseteq \mathcal{U} \setminus X$.
- $r_2 \Rightarrow r_1$: Obviously, $\bigcup \sigma(\mathsf{f}(F,X)) \cap X = \varnothing \Rightarrow \bigcup \sigma(\mathsf{f}(F,X)) \cap X = \varnothing$.
- $s_1 \Rightarrow r_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \cap X \subseteq A(\mathsf{f}(F,X)) \cap X = \varnothing$.
- $v_1 \Rightarrow v_2$: $\bigcup \sigma(F) \cap X = \varnothing \Rightarrow \bigcap \sigma(F) \cap X = \varnothing \Rightarrow F = \mathsf{f}(F,X)$.
- $v_2 \Rightarrow v_3$: $A(F) \cap X = \varnothing \Rightarrow \bigcup \sigma(F) \cap X = \varnothing \Rightarrow F = \mathsf{f}(F,X)$.

In order to argue that the remaining relations do not hold we have to provide counter examples. Due to the limited space we provide two illustrating examples only. The remaining non-relations can be shown in a similar fashion.

- $e_4 \not\Rightarrow e_1$: Towards a contradiction suppose $e_4 \Rightarrow e_1$. Consider the AF $F = (\{a,b\}, \varnothing)$ and $X = \{b\}$. For any considered semantics σ we have, $\sigma(F) = \{\{a,b\}\}$. Let f be a forgetting operator satisfying e_4. Thus, $\sigma(\mathsf{f}(F,X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\} = \varnothing$. On the other hand, since f satisfies e_1 too we derive, $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\} = \{\{a\}\}$. Contradiction.
- $e_4 \not\Rightarrow s_1$: Consider a forgetting operator f satisfying e_4, i.e. $\sigma(\mathsf{f}(F,X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\}$. Assume that s_1 is satisfied, i.e. $A(\mathsf{f}(F,X)) \cap X = \varnothing$. Pick an argument $x \in X$ and define a new operator g, s.t. $A(\mathsf{g}(F,X)) = A(\mathsf{f}(F,X)) \cup \{x\}$ and $R(\mathsf{g}(F,X)) = R(\mathsf{f}(F,X)) \cup \{(z,x) \mid z \in A(\mathsf{g}(F,X))\}$. Obviously, e_4 is still satisfied by g since $\sigma(\mathsf{f}(F,X)) = \sigma(\mathsf{g}(F,X))$ by construction, but s_1 is not.

■

Apart from the relationships concerning single conditions there are more complex implications. In the realm logic programming it was already shown that $(\mathbf{SP})$ is necessary and sufficient for $(\mathbf{SI})$ and $(\mathbf{CP})$ (Gonçalves, Knorr, and Leite 2016a). Beside other interesting relations we state the analogous result for abstract argumentation in Item 5 of the following proposition. Please note that the proof is astonishingly simple.

Proposition 2. *For any semantics $\sigma \in \{stg, stb\}$:*

1. *s_2, l_0 and $e_{3_\subseteq}$ imply e_2,*
2. *s_2, l_0 and $e_{3_\supseteq}$ imply e_2.*

Moreover, for any $\tau \in \{stg, stb, ss, pr, gr, il, eg\}$ we have:

3. *$e_{3_\subseteq}$ and $e_{3_\supseteq}$ if and only if e_1,*
4. *$e_{\mathbf{sC}}$ and $e_{\mathbf{wC}}$ if and only if e_1,*
5. *e_1 and l_0 if and only if e_2.*

Proof:

1. Let f satisfies s_2, l_0 and $e_{3_\subseteq}$. In order to show desideratum e_2 we will first prove that condition e_1 is implied. Then, applying Statement 5 of this Proposition yields e_2. Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. Desideratum e_1 requires $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$.
   Due to s_2 we have $A(\mathsf{f}(F,X)) = A(F) \setminus X := A$. Define a copy of A, i.e. a set of fresh arguments $A' = \{a' \mid a \in A\}$, s.t. $A' \cap (A(F) \cup X) = \varnothing$. Let us define the AF $H = (A \cup A', \{(a,a') \mid a \in A\})$. By construction any argument $a \in A$ attacks its copy $a' \in A'$. Consequently, for any extension $E \in \sigma(F)$ we have, $E \cup \{a' \in A' \mid a \in A \setminus E\} \in \sigma(F \sqcup H)$. Vice versa, any $E' \in \sigma(F \sqcup H)$ can be uniquely associated with some $E \in \sigma(F)$, s.t. $E' = E \cup \{a' \in A' \mid a \in A \setminus E\}$. Therefore, for both semantics, stable and stage, we deduce for any $a \in A$: either $a \in E'$ or (its copy) $a' \in E'$.
   The same relation between extensions applies to $\mathsf{f}(F,X)$ and $\mathsf{f}(F,X) \sqcup H$. Furthermore, since $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ due to l_0 we may even conclude the same behaviour regarding arguments $a \in A$ for $\mathsf{f}(F \sqcup H, X)$. More precisely, for any extension $E' \in \sigma(\mathsf{f}(F \sqcup H, X))$, either $a \in E'$ or $a' \in E'$.
   - ($\supseteq$) If $E \in \sigma(F)$, then $E \setminus X \in \sigma(\mathsf{f}(F,X))$.
     Since $E \in \sigma(F)$ we deduce $E \cup \{a' \in A' \mid a \in A \setminus E\} \in \sigma(F \sqcup H)$. Due to $e_{3_\subseteq}$ we obtain $T(E \cup \{a' \in A' \mid a \in A \setminus E\}) \subseteq (E \cup \{a' \in A' \mid a \in A \setminus E\}) \setminus X = E \setminus X \cup \{a' \in$

$A' \mid a \in A \setminus E\}$ for some function $T : \sigma(F \sqcup H) \rightarrow 2^{\mathcal{U}}$ and $T(E \cup \{a' \in A' \mid a \in A \setminus E\}) \in \sigma(\mathsf{f}(F \sqcup H, X))$. Assuming $T(E \cup \{a' \in A' \mid a \in A \setminus E\}) \subsetneq E \setminus X \cup \{a' \in A' \mid a \in A \setminus E\}$ yields the existence of an $a \in A$, s.t. neither $a \in T(E \cup \{a' \in A' \mid a \in A \setminus E\})$, nor $a' \in T(E \cup \{a' \in A' \mid a \in A \setminus E\})$. Contradiction. Hence, $T(E \cup \{a' \in A' \mid a \in A \setminus E\}) = E \setminus X \cup \{a' \in A' \mid a \in A \setminus E\}$. Moreover, applying l_0 justifies $E \setminus X \cup \{a' \in A' \mid a \in A \setminus E\} \in \sigma(\mathsf{f}(F, X) \sqcup H)$. In consideration of the one-to-one correspondence we deduce $E \setminus X \in \sigma(\mathsf{f}(F, X))$ as claimed.

- ($\subseteq$) If $E' \in \sigma(\mathsf{f}(F, X))$, then $E' = E \setminus X$ for some $E \in \sigma(F)$.
  Due to condition $e_{3_{\subseteq}}$ we know $\sigma(\mathsf{f}(F, X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \rightarrow 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \setminus X$. This means, there is some $E \in \sigma(F)$, s.t. $E' \subseteq E \setminus X$. Applying the former case ($\supseteq$) yields $E \setminus X \in \sigma(\mathsf{f}(F, X))$. Since stable as well as stage semantics satisfies I-maximality, i.e. $\sigma(\mathsf{f}(F, X))$ has to form a $\subseteq$-antichain we deduce $E' = E \setminus X$ concluding the proof.

2. This proof is analogous to the previous one.
3. ($\Leftarrow$) Confer Proposition 1.
   ($\Rightarrow$) Let f satisfies $e_{3_{\supset}}$ and $e_{3_{\subset}}$. Furthermore, let F be an AF and $X \subseteq \mathcal{U}$ a set of arguments. Consider now a certain extension $E \in \tau(F)$. Due to $e_{3_{\supset}}$ and $e_{3_{\subset}}$ we deduce that there are two functions $T_1, T_2 : \tau(F) \rightarrow 2^{\mathcal{U}}$ s.t. $T_1(E) \supseteq E \setminus X$ and $T_2(E) \subseteq E \setminus X$. Since any considered semantics satifies I-maximality, i.e. $\tau(\mathsf{f}(F, X))$ has to form a $\subseteq$-antichain we deduce $T_1(E) = E \setminus X = T_2(E)$. Hence, $\tau(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \tau(F)\}$ as required.
4. Obvious.
5. ($\Leftarrow$) Confer Proposition 1.
   ($\Rightarrow$) $\tau(\mathsf{f}(F, X) \sqcup H) =^{(l_0)} \tau(\mathsf{f}(F \sqcup H, X)) =^{(e_1)} \{E \setminus X \mid E \in \tau(F \sqcup H)\}$

■

4 Satisfiability and Unsatisfiability

In this section we consider the satisfiability of single conditions as well as whole sets of desiderata. Most of the results underline the intrinsic limits of forgetting in abstract argumentation as they prove unsatisfiability.

Individual Desiderata

We start with a positive result regarding individual satisfiablity. In fact, 19 conditions are satisfiable under any considered semantics if considered in isolation.

Proposition 3. *Desideratum $d \in \{e_{3_{\supseteq}}, e_{3_{\subseteq}}, e_{\mathbf{sC}}, r_1, r_2, r_3, r_4, s_1, s_2, s_3, m_1, m_2, m_3, m_4, v_1, v_2, v_3, l_0, l_1\}$ is satisfiable under any semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$.*

Proof: In the following we will only show that each desideratum $d \in \{e_{3_{\supseteq}}, e_{3_{\subseteq}}, e_{\mathbf{sC}}, r_3, r_4, v_1\}$ is satisfiable. The satisfiability of the remaining desiderata is implied by Proposition 1.
Given an AF F and a set of arguments X.

- $e_{3_{\supseteq}}$: If $\sigma(F) = \emptyset$, then set $\mathsf{f}(F, X) = F$. If not, define $\mathsf{f}(F, X) = (\bigcup_{E \in \sigma(F)} E \setminus X, \emptyset)$. Consequently, $\sigma(\mathsf{f}(F, X)) = \{\bigcup_{E \in \sigma(F)} E \setminus X\}$. Thus, the constant function $T : \sigma(F) \rightarrow 2^{\mathcal{U}}$ with $E \mapsto T(E) = \bigcup_{E \in \sigma(F)} E \setminus X$ satisfies $T(E) \supseteq E \setminus X$ for any $E \in \sigma(F)$ as desired.
- $e_{3_{\subseteq}}$: If $\sigma(F) = \emptyset$, then set $\mathsf{f}(F, X) = F$. If not, define $\mathsf{f}(F, X) = (\emptyset, \emptyset)$. Thus, the constant function $T : \sigma(F) \rightarrow 2^{\mathcal{U}}$ with $E \mapsto T(E) = \emptyset$ satisfies $T(E) \subseteq E \setminus X$ for any $E \in \sigma(F)$ as required.
- $e_{\mathbf{sC}}$: If $\sigma(F) = \emptyset$, then set $\mathsf{f}(F, X) = F$. If not, just pick an arbitrary $E \in \sigma(F)$ and define $\mathsf{f}(F, X) = (E \setminus X, \emptyset)$. Obviously, $\sigma(\mathsf{f}(F, X)) = \{E \setminus X\}$ justifying $\sigma(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$.
- r_3: Consider $\mathsf{f}(F, X) = ((\bigcap \sigma(F)) \setminus X, \emptyset)$.
- r_4: Define $\mathsf{f}(F, X) = ((\bigcup \sigma(F)) \setminus X, \emptyset)$.
- v_1: Set $\mathsf{f}(F, X) = F$.

■

The following proposition shows a dividing line between uniquely and universally defined semantics. The subset antichain property of the latter family prevent the satisfiability of e_1 and $e_{\mathbf{wC}}$.

Proposition 4. *Desiderata e_1 and $e_{\mathbf{wC}}$ are satisfiable under any $\tau \in \{gr, il, eg\}$, but not under $\sigma \in \{stb, stg, ss, pr\}$.*

Proof: Consider the uniquely defined semantics τ. For any AF F we obtain a singleton as extension-set, i.e. $\tau(F) = \{E\}$. Define $\mathsf{f}(F, X) = (E \setminus X, \emptyset)$. Thus, $\tau(\mathsf{f}(F, X)) = \{E \setminus X\}$ proving e_1 and therefore $e_{\mathbf{wC}}$.
Consider now the universally defined semantics σ and let $F = (\{a, x\}, \{(a, x), (x, a)\})$ be an specific AF. We obtain $\sigma(F) = \{\{a\}, \{x\}\}$. According to $e_{\mathbf{wC}}$ it must hold $\sigma(\mathsf{f}(F, \{x\})) \supseteq \{\emptyset, \{a\}\}$. This means that $\sigma(\mathsf{f}(F, \{x\}))$ does not form a $\subseteq$-antichain. Hence, no such $\mathsf{f}(F, \{x\})$ can exist. ■

The following two propositions are mainly due to already shown results in (Baumann, Gabbay, and Rodrigues 2020).

Proposition 5. *Desiderata e_4 is satisfiable under stable semantics, but not under any $\tau \in \{stg, ss, pr, gr, il, eg\}$.*

Proof: Consider stable semantics. In (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1, Example 4) an operator f was introduced able to precisely remove a stable extension, whenever it contains a certain argument x. Since we consider finite AFs and thus, finite forgetting sets X we obtain a new operator satisfying e_4 by simply applying f iteratively. Note that forgetting result is sensitive to the order of forgetting (cf. Example 3 for an illustration). Therefore, a predefined order is essential.
The impossibility of satisfying e_4 under τ was already shown for singletons (Baumann, Gabbay, and Rodrigues 2020, Proposition 5). Thus, it does not work for arbitrary sets either. ■

Proposition 6. *Desiderata e_2 is unsatisfiable under any semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$.*

Proof: The desideratum e_2 was shown to be unsatisfiable when forgetting single arguments only (Baumann, Gabbay, and Rodrigues 2020, Proposition 6). Hence, unsatisfiability for arbitrary sets is implied. ■

Combined Desiderata

In the following we consider the satisfiability of whole sets of conditions. Most of the results underline the intrinsic limits of forgetting in abstract argumentation.

Proposition 7. *We have the following satisfiability results:*

1. $\{s_2, l_0, e_{3_\subseteq}\}$ *as well as* $\{s_2, l_0, e_{3_\supseteq}\}$ *are unsatisfiable for any semantics* $\mu \in \{\mathit{stg}, \mathit{stb}\}$.
2. *Moreover,* $\{e_{3_\subseteq}, e_{3_\supseteq}\}$ *and* $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ *are unsatisfiable for any semantics* $\sigma \in \{\mathit{stg}, \mathit{stb}, \mathit{ss}, \mathit{pr}\}$*, but satisfiable for each* $\tau \in \{\mathit{gr}, \mathit{il}, \mathit{eg}\}$.

Proof:

1. Let $\mu \in \{\mathit{stg}, \mathit{stb}\}$. According to Items 1 and 2 of Proposition 2 we have that $\{s_2, l_0, e_{3_\subseteq}\}$ as well as$\{s_2, l_0, e_{3_\supseteq}\}$ imply e_2. The latter is unsatisfiable due to Proposition 6. Hence, both sets are unsatisfiable under μ.
2. Given $\sigma \in \{\mathit{stb}, \mathit{stg}, \mathit{ss}, \mathit{pr}\}$ and $\tau \in \{\mathit{gr}, \mathit{il}, \mathit{eg}\}$. The sets of desiderata $\{e_{3_\subseteq}, e_{3_\supseteq}\}$ respective $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ imply e_1 (Items 3 and 4 of Proposition 2). Hence, both are unsatisfiable under σ, but satisfiable under τ (cf. Proposition 4).

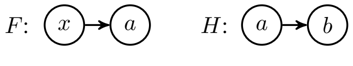
■

The next two results show that stable semantics is somehow exceptional regarding its potential for forgetting.

Proposition 8. *The set* $\{l_0, e_{\mathbf{sC}}\}$ *is satisfiable under stable semantics but not under* $\sigma \in \{\mathit{gr}, \mathit{stg}, \mathit{ss}, \mathit{pr}\}$.

Proof:

- The set of desiderata $\{l_0, e_{\mathbf{sC}}\}$ is satisfied under stable semantics by setting $\mathsf{f}(F,X) = (A(F), R(F) \setminus \{(a,x) \mid x \in X\} \cup \{(x,x) \mid x \in X \cap A(F)\})$.
  Please note that $\mathsf{f}(F,X) \sqcup H = \mathsf{f}(F \sqcup H, X)$ if considering AFs H, s.t. $A(H) \cap X = \emptyset$. Consequently, l_0 is fulfilled since $\sigma(\mathsf{f}(F,X) \sqcup H) = \sigma(\mathsf{f}(F \sqcup H, X))$ is implied for any semantics σ.
  Moreover, if $A(F) \cap X = \emptyset$, then $\mathsf{f}(F,X) = F$ and hence, $\sigma(\mathsf{f}(F,X)) = \sigma(F)$ for any semantics σ. If not, i.e. $A(F) \cap X \neq \emptyset$ we observe that $\mathsf{f}(F,X)$ collapses for stable semantics, i.e. $\mathit{stb}(\mathsf{f}(F,X)) = \emptyset$. In both cases, $\mathit{stb}(\mathsf{f}(F,X)) \subseteq \{E \setminus X \mid E \in \mathit{stb}(F)\}$ yielding $e_{\mathbf{sC}}$.
- Towards a contradiction suppose that there is a forgetting operator f satisfying l_0 and $e_{\mathbf{sC}}$. Consider the following AFs F and H.

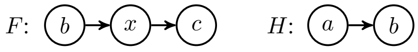


For any semantics $\sigma \in \{\mathit{gr}, \mathit{ss}, \mathit{pr}\}$ we have $\sigma(F) = \{\{x\}\}$ as well as $\sigma(F \sqcup H) = \{\{x,b\}\}$. Since all considered semantics are universally defined we deduce $\sigma(\mathsf{f}(F,X)) \neq \emptyset$ for any set of arguments X. Let $X = \{x\}$. Applying condition $e_{\mathbf{sC}}$, i.e. $\sigma(\mathsf{f}(F,X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$ yields $\sigma(\mathsf{f}(F,X)) = \{\emptyset\}$ and $\sigma(\mathsf{f}(F \sqcup H, x)) = \{\{b\}\}$, respectively. Due to l_0 we further infer $\sigma(\mathsf{f}(F,X) \sqcup H)) = \{\{b\}\}$.

This already yields a contradiction in case of grounded semantics since $\mathit{gr}(\mathsf{f}(F,X) \sqcup H)) = \{\{b\}\}$ implies b is unattacked in $\mathsf{f}(F,X) \sqcup H$ which is obviously not true.

For preferred and semi-stable semantics we deduce that $\{b\}$ is admissible in $\mathsf{f}(F,X) \sqcup H$. Hence, the AF $H' = (\{a,b\}, \{(b,a)\})$ must be a subframework of $\mathsf{f}(F,X)$. Moreover, whenever b is attacked by some $c \neq a$ in $\mathsf{f}(F,X)$, it has to be counterattacked by b in $\mathsf{f}(F,X)$ because admissibility of $\{b\}$ in $\mathsf{f}(F,X) \sqcup H$ has to be guarenteed. Consequently, $\{b\} \in \mathit{ad}(\mathsf{f}(F,X))$ is implied too. In case of preferred semantics we infer the existence of a set E, s.t. $\{b\} \subseteq E \in \mathit{pr}(\mathsf{f}(F,X))$. For semi-stable we deduce that either $\{b\}$ is already semi-stable in $\mathsf{f}(F,X)$, or there is an admissible set E, s.t. $\{b\}^\oplus \subset E^\oplus$ with $E \in \mathit{ss}(\mathsf{f}(F,X))$. Note that $E \neq \emptyset$ is implied. For both semantics, $\sigma(\mathsf{f}(F,X)) \neq \{\emptyset\}$. Contradiction!

Let us turn now to stage semantics. Consider therefore the following two AFs



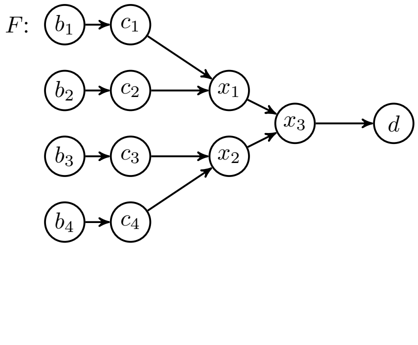
where $a \notin A(\mathsf{f}(F,X))$, i.e. a is an argument that does not appear in the forgetting result $\mathsf{f}(F,X)$. We have $\mathit{stg}(F) = \{\{b,c\}\}$ and $\mathit{stg}(F \sqcup H) = \{\{a,x\}\}$. Since $\mathit{stg}(F)$ is non-empty for any AF, through use of $e_{\mathbf{sC}}$ we can deduce that $\mathit{stg}(\mathsf{f}(F,x)) = \{\{b,c\}\}$, and respectively $\mathit{stg}(\mathsf{f}(F \sqcup H, x)) = \{\{a\}\}$. Since c appears in an extension of $\mathsf{f}(F,x)$, we have $(c,c) \notin R(\mathsf{f}(F,x))$, hence $(c,c) \notin R(\mathsf{f}(F,x) \sqcup H)$. Also $(a,c), (c,a) \notin R(\mathsf{f}(F,x) \sqcup H)$. Then $\{a,c\}$ is conflict-free and hence, $\{a\}^\oplus \subset \{a,c\}^\oplus$, resulting in $\{a\} \notin \mathit{stg}(\mathsf{f}(F,x) \sqcup H)$. Then $\mathit{stg}(\mathsf{f}(F,x) \sqcup H) \neq \mathit{stg}(\mathsf{f}(F \sqcup H, x)) = \{\{a\}\}$ contradicting l_0.

■

Proposition 9. *The set* $\{l_1, e_{\mathbf{sC}}\}$ *is satisfiable under stable semantics but not under* $\sigma \in \{\mathit{gr}, \mathit{stg}, \mathit{ss}, \mathit{pr}\}$.

Proof:

- The set of desiderata $\{l_1, e_{\mathbf{sC}}\}$ is satisfiable under stable semantics as it is implied by the satisfiable set of conditions $\{l_0, e_{\mathbf{sC}}\}$ (cf. Figure 1 and Proposition 8).
- Let $\sigma \in \{\mathit{gr}, \mathit{stg}, \mathit{ss}, \mathit{pr}\}$. Towards a contradiction suppose that there is a forgetting operator f satisfying l_1 and $e_{\mathbf{sC}}$ under σ. Consider the following AF F and let $X = \{x_1, x_2, x_3\}$.



Obviously, $\sigma(F) = \{\{b_1, b_2, b_3, b_4, x_1, x_2, d\}\}$. Let $\mathsf{f}(F, X)$ be the forgetting result. Let further a_1, a_2, a_3 and a_4 be arguments not contained in $A(\mathsf{f}(F, X))$. For each $1 \leq i \leq 4$ we define $H_i = (\{a_i, b_i\}, \{(a_i, b_i)\})$. Hence,
$\sigma(F \sqcup H_1 \sqcup H_3) = \{\{a_1, b_2, a_3, b_4, c_1, c_3, x_3\}\}$,
$\sigma(F \sqcup H_2 \sqcup H_4) = \{\{b_1, a_2, b_3, a_4, c_2, c_4, x_3\}\}$,
$\sigma(F \sqcup H_1 \sqcup H_2) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, x_2, d\}\}$, and
$\sigma(F \sqcup H_3 \sqcup H_4) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, x_1, d\}\}$.

Using the universal definedness of any considered semantics σ together with condition $e_{\mathbf{sC}}$ yields:

$\sigma(\mathsf{f}(F \sqcup H_1 \sqcup H_3, X)) = \{\{a_1, b_2, a_3, b_4, c_1, c_3\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_2 \sqcup H_4, X)) = \{\{b_1, a_2, b_3, a_4, c_2, c_4\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_1 \sqcup H_2, X)) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, d\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_3 \sqcup H_4, X)) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, d\}\}$.

Applying l_1 justifies:

$\sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3) = \{\{a_1, b_2, a_3, b_4, c_1, c_3\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_2 \sqcup H_4) = \{\{b_1, a_2, b_3, a_4, c_2, c_4\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, d\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, d\}\}$.

The last two lines show that any a_i, b_i as well as c_i appears together with d in at least one extension. Consequently, the forgetting result $\mathsf{f}(F, X)$ neither contains attacks between a_i and d, nor b_i and d, nor c_i and d.
Hence, $\{a_1, b_2, a_3, b_4, c_1, c_3\} \in cf(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$, implies $\{a_1, b_2, a_3, b_4, c_1, c_3, d\} \in cf(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$. Moreover, $\{a_1, b_2, a_3, b_4, c_1, c_3\}^{\oplus} \subset \{a_1, b_2, a_3, b_4, c_1, c_3, d\}^{\oplus}$ which contradicts $\{a_1, b_2, a_3, b_4, c_1, c_3\} \in stg(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$.
Let us consider the remaining semantics, i.e. $\sigma \in \{gr, pr, ss\}$. Since $d \notin \{a_1, b_2, a_3, b_4, c_1, c_3\} \in \sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ as well as $d \notin \{b_1, a_2, b_3, a_4, c_2, c_4\} \in \sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$ we conclude that d must be attacked by an argument $e \notin \{a_1, \dots, a_4, b_1, \dots b_4, c_1, \dots, c_4\} = A$ not being counterattacked by any $a \in A$. If so, we deduce $\{a_1, a_2, b_3, b_4, c_1, c_2, d\} \notin ad(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ as well as $\{b_1, b_2, a_3, a_4, c_2, c_4, d\} \notin ad(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$. Thus, $\{a_1, a_2, b_3, b_4, c_1, c_2, d\} \notin \sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ and $\{b_1, b_2, a_3, a_4, c_2, c_4, d\} \notin \sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$. Contradiction!

■

Testing the Limits: Promising Combinations

Proposition 10 shows the compatibility of promising combinations of a semantical and syntactical condition. The strongest syntactical desideratum s_3 is incompatible with all considered semantical conditions and can only be trivially combined with r_1 and r_2. In this context, *trivial* means, that already one condition implies the other as shown in Proposition 1. Stable semantics is the only considered semantics able to collapse for certain AFs. This unique property is reflected in its different behaviour regarding the fulfillment of combined desiderata (see Figure 2).

Proposition 10. *Figure 2 summarizes the compatibility under semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. A "✓"/"×" in cell (l,c) indicates whether or not the conditions in line l and column c are simultaneously satisfiable under σ. The symbol "τ" restricts the satisfiability to the semantics gr, il and eg, the symbol "stb" to stable semantics and the symbol "¬stb" to all semantics but stable respectively. The combinations in a dark background are trivial .*

	s_1	s_2	s_3	m_1	m_2	m_3	m_4
r_1	✓	✓	✓	✓	✓	✓	✓
r_2	✓	✓	✓	✓	✓	✓	✓
r_3	✓	✓	×	✓	✓	✓	✓
r_4	✓	✓	×	✓	✓	✓	✓
e_1	τ	τ	×	τ	τ	τ	τ
e_2	×	×	×	×	×	×	×
$e_{3_\subseteq}$	✓	$\neg stb$	×	✓	✓	✓	✓
$e_{3_\supseteq}$	✓	$\neg stb$	×	✓	✓	✓	✓
e_4	stb	×	×	stb	stb	stb	stb
$e_{\mathbf{sC}}$	✓	$\neg stb$	×	✓	✓	✓	✓
$e_{\mathbf{wC}}$	τ	τ	×	τ	τ	τ	τ

Figure 2: Compatibility of syntactical/semantical conditions

Proof: The proof involves 77 combinations of desiderata which has to be checked with respect to 7 semantics. It takes more than 3 pages and is omitted here due to the limited space. ■

5 Forgetting under Stable Semantics

Let us reflect on the proposed extension-based conditions as listed in Desidirata 1. At first we observe that e_1 and e_4 represent two opposing philosophies about the concept of forgetting. Desideratum e_1 requires that any former extension has to survive in an adjusted fashion, namely new extensions are obtained from initial ones via deleting the arguments which has to be forgotten ($\sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$). In contrast, Condition e_4 requires to delete any extension carrying arguments which has to be forgotten ($\sigma(\mathsf{f}(F, X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \emptyset\}$). Both interpretations of forgetting are independent as shown in Proposition 1. Any other considered extension-based condition is either a relaxation of e_1, or a lifting of this interpretation to the level of strong equivalence. In the following we will consider these two main desiderata in more detail. We restrict ourselves to stable semantics and left the consideration of other semantics for future work.

Forgetting via e_4 Quite recently, an e_4-operator f for forgetting single arguments was presented (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1). In Example 3 of the introductory part we have seen that applying this operator f iteratively does not necessarily produce a desirable outcome.

Moreover, this procedure is sensitive to the order of forgetting. How to adapt the existing procedures for multiple arguments?

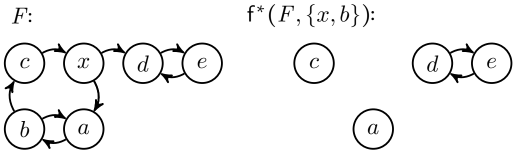
The former construction consists of two steps. Given an AF F and an argument x. First, remove x and its related attacks, i.e. consider $F_x := F|_{A(F)\setminus\{x\}}$. Any stable extension E of F not containing x remains stable in F_x. However, new extensions may arise. Now, in a second step, each new (undesired) extension E' is removed via adding a self-defeating argument not attacked by E'.

This procedure can be more or less directly applied to forget a whole set of arguments X. In the first step we simply restrict the initial framework to $A(F) \setminus X$, i.e. we consider $F|_{A(F)\setminus X}$. Thus, any former extension containing arguments from X is not stable anymore but any other survives. And secondly, we eliminate any unwanted extensions via the addition of self-attacking arguments. This yields the following algorithm.

Algorithm 1: Construct $G = \mathsf{f}^*(F, X)$

```
Input  : AF $F$; arguments $X \subseteq \mathcal{U}$
Output : AF $G$ satisfying $\{s_1, e_4, v_3\}$
Function compute_G($F, X$)
    if $X \cap A(F) = \emptyset$ then $G \leftarrow F$;
    else
        $G_0 \leftarrow F|_{A(F)\setminus X}$;
        $A = A(G_0)$;  $R \leftarrow R(G_0)$;
        foreach $E_i \in stb(G_0) \setminus stb(F)$ do
            Let $a_i$ be a fresh argument s.t. $a_i \notin A(F) \cup A$;
            $A \leftarrow A \cup \{a_i\}$; $R \leftarrow R \cup \{(a_i, a_i)\}$
            foreach $y \in \bigcup stb(G_0) \setminus E_i$ do
                $R \leftarrow R \cup \{(y, a_i)\}$;
        $G \leftarrow (A, R)$;
    return $G$;
```

Example 4 (Example 3 cont.)**.** *Consider again AF F. We have $stb(F) = \{\{x, b, e\}, \{a, c, d\}, \{a, c, e\}\}$. Let $X = \{x, b\}$. Applying Algorithm 1 immediately yields $\mathsf{f}^*(F, X) = F|_{A(F)\setminus X}$ as $stb(F|_{A(F)\setminus X}) = \{\{a, c, d\}, \{a, c, e\}\}$.*



The attentive reader may have already noticed that $\mathsf{f}^*(F, \{x, b\}) = \mathsf{f}(\mathsf{f}(F, b), x)$. This means, forgetting x and b simultaneously yields the more compact outcome if applying the former operator f iteratively ($\mathsf{f}(\mathsf{f}(F, b), x)$ vs. $\mathsf{f}(\mathsf{f}(F, x), b)$). In fact, it can be shown that forgetting through f^* necessarily yields a smaller AF, than iterating f no matter which order is chosen.

Forgetting via e_1 The main reason for the impossibility to find an operator satisfying e_1 under stable semantics is an intrinsic one, namely realizability. More precisely, certain instances, i.e. an initial framework F and a set X of arguments would enforce a framework F' with a set of stable extensions violating the $\subseteq$-antichain property or tightness. Consequently, one reasonable strategy is to look for forgetting operators satisfying e_1 whenever possible, and trying to satisfy a certain relaxation if not. Natural candidates would be $e_{3_\subseteq}$, $e_{3_\supseteq}$ or $e_{\mathbf{sC}}$. A similar procedure was suggested and also implemented for *strong persistence* in the realm of logic programming (Gonçalves et al. 2017).

The question which relaxation to choose has no clear answer. First of all, according to Figure 1 each proposed desideratum is independent of the other. Moreover, each relaxation has its particular advantages and drawbacks. It does not make sense to express general preferences among the desiderata as the specific application will determine which criterion is most suitable. Furthermore, even for a particular chosen relaxation, the precise result of forgetting might not be clear. This is demonstrated by the following example.

Example 5. *Consider again AF F presented in Example 2. Let $X = \{a, b, c\}$. As $stb(F) = \{\{a, e, f\}, \{b, f, d\}, \{c, d, e\}\}$ we obtain $\{E \setminus X \mid E \in stb(F)\} = \{\{d, e\}, \{d, f\}, \{e, f\}\}$. This set is not tight implying that e_1 is impossible. The relaxation $e_{\mathbf{sC}}$ is satisfied by any AF F' with $stb(F') \subseteq \{\{d, e\}, \{d, f\}, \{e, f\}\}$. Giving up one of these three extensions would result in a realizable set. However, without further information there is no reason to prefer one set over the other.*

In summary, that means both the choice of how to relax e_1 as well as its particular implementation depends on the application in mind. A more thorough study on this issue is left for future work.

6 Discussion and Conclusion

The paper shed more light on forgetting in abstract argumentation. One central motivation was to convey ideas and desiderata from recent studies of forgetting in the realm of logic programming (Knorr and Alferes 2014; Gonçalves, Knorr, and Leite 2016a; Berthold et al. 2019a; 2019b). We redefined several principles and provided a comprehensive study regarding satisfiability and relations. Moreover, we demonstrated that already existing forgetting operators from logic programming cannot be unconditionally applied to abstract argumentation. The two main reasons are: First, the use of such an operator does not guarantee to stay within the AF-fragment and secondly (as well as more importantly), there are essential differences in the expressibility between both formalisms. Finally, we presented a specific forgetting operator for a particular combination of conditions inspired by an algorithm introduced in (Baumann, Gabbay, and Rodrigues 2020).

One future line of research is the study of forgetting regarding labelling-based semantics (Baroni, Caminada, and Giacomin 2018). These kind of semantics provide some more information than their extension-based counterparts. In contrast to the latter they allow to distinguish explicitly two different kinds of not being accepted, namely *out* (attacked by an accepted argument) and *undec* (not attacked by an accepted argument). Consequently, more differentiated desiderata can be formalized regarding the acceptance status of the arguments to be forgotten. A further related work in this context is (Rienstra et al. 2020) dealing with so-called *robustness* principles. The paper studies the question to which extent old labellings persist/new labellings arise if a certain change of the initial AF is performed. Such results are highly relevant for the theory of forgetting as they can be used to show the satisfiability/unsatisfiability of desidered properties.

Finally, the paper can be seen as one central part of a much broader investigation on how properties of forgetting on the abstract and structured level are related. One interesting agenda might be to consider *rationality postulates* for forgetting (Caminada 2017).

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF, 1IS18026A) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig.

References

Baroni, P., and Giacomin, M. 2007. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.* 171:675–700.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L. 2018. *Handbook of Formal Argumentation*. College Publications.

Baroni, P.; Caminada, M.; and Giacomin, M. 2018. Abstract argumentation frameworks and their semantics. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 4.

Baumann, R., and Spanring, C. 2015. Infinite argumentation frameworks - On the existence and uniqueness of extensions. In *Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060, 281–295. Springer.

Baumann, R.; Gabbay, D. M.; and Rodrigues, O. 2020. Forgetting an argument. In *The Thirty-Fourth AAAI Conference on Artif. Intell., AAAI 2020, The Thirty-Second Innovative Applications of Artif. Intell. Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artif. Intell., EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2750–2757. AAAI Press.

Baumann, R. 2018. On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 18. also appears in IfCoLog Journal of Logics and their Applications 4(8):2779-2886.

Berthold, M.; Gonçalves, R.; Knorr, M.; and Leite, J. 2019a. Forgetting in answer set programming with anonymous cycles. In Moura Oliveira, P.; Novais, P.; and Reis, L. P., eds., *Progress in Artif. Intell.*, 552–565. Cham: Springer International Publishing.

Berthold, M.; Gonçalves, R.; Knorr, M.; and Leite, J. 2019b. A syntactic operator for forgetting that satisfies strong persistence. *Theory and Practice of Logic Programming* 19(5-6):1038–1055.

Bisquert, P.; Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M. 2011. Change in argumentation systems: Exploring the interest of removing an argument. In *Scalable Uncertainty Management - 5th International Conference, SUM 2011*, 275–288.

Caminada, M. 2017. Rationality postulates: Applying argumentation theory for non-monotonic reasoning. *FLAP* 4(8).

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–357.

Dunne, P. E.; Dvořák, W.; Linsbichler, T.; and Woltran, S. 2015. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.* 228:153–178.

Eiter, T., and Kern-Isberner, G. 2018. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI - Künstliche Intelligenz*.

Eiter, T.; Fink, M.; Pührer, J.; Tompits, H.; and Woltran, S. 2013. Model-based recasting in answer-set programming. *Journal of Applied Non-Classical Logics* 23(1-2):75–104.

Gonçalves, R.; Knorr, M.; Leite, J.; and Woltran, S. 2017. When you must forget: Beyond strong persistence when forgetting in answer set programming. *Theory and Practice of Logic Programming* 17(5-6):837–854.

Gonçalves, R.; Knorr, M.; and Leite, J. 2016a. The ultimate guide to forgetting in answer set programming. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016*, 135–144.

Gonçalves, R.; Knorr, M.; and Leite, J. 2016b. You can't always forget what you want: On the limits of forgetting in answer set programming. In *ECAI 2016 - 22nd European Conference on Artif. Intell.*, 957–965.

Knorr, M., and Alferes, J. J. 2014. Preserving strong equivalence while forgetting. In *Proceedings of the 14th European Conference on Logics in Artif. Intell. (JELIA-14)*, 412–425.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2(4):526–541.

Lifschitz, V.; Tang, L. R.; and Turner, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artif. Intell.* 25(3-4):369–389.

Lin, F., and Reiter, R. 1994. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, 154–159.

Rienstra, T.; Sakama, C.; van der Torre, L.; and Liao, B. 2020. A principle-based robustness analysis of admissibility-based argumentation semantics. *Argument & Computation* 11(3):305–339.

Simari, G. R., and Rahwan, I., eds. 2009. *Argumentation in Artif. Intell.* Springer.

Strass, H. 2013. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.* 205:39–70.

Equivalence in Argumentation Frameworks with a Claim-centric View – Classical Results With Novel Ingredients

Ringo Baumann[1] and **Anna Rapberger**[2] and **Markus Ulbricht**[1,2]
[1]Leipzig University, Department of Computer Science
[2]TU Wien, Institute of Logic and Computation
[1]{baumann,mulbricht}@informatik.uni-leipzig.de
[2]arapberg@dbai.tuwien.ac.at

Abstract

A common feature of non-monotonic logics is that the classical notion of equivalence does not preserve the intended meaning in light of additional information. Consequently, the term strong equivalence was coined in the literature and thoroughly investigated. In the present paper, the knowledge representation formalism under consideration are claim-augmented argumentation frameworks (CAFs) which provide a formal basis to analyze conclusion-oriented problems in argumentation by adapting a claim-focused perspective. CAFs extend Dung AFs by associating a claim to each argument representing its conclusion. In this paper, we investigate both ordinary and strong equivalence in CAFs. Thereby, we take the fact into account that one might either be interested in the actual arguments or their claims only. The former point of view naturally yields an extension of strong equivalence for AFs to the claim-based setting while the latter gives rise to a novel equivalence notion which is genuine for CAFs. We tailor, examine and compare these notions and obtain a comprehensive study of this matter for CAFs. We conclude by investigating the computational complexity of naturally arising decision problems.

1 Introduction

Equivalence is an important subject of research in knowledge representation and reasoning. Given a knowledge base $\mathcal{K}$, finding an equivalent one, say $\mathcal{K}'$, helps to obtain a better understanding or more concise representation of $\mathcal{K}$. From a computational point of view, equivalence is particularly interesting whenever a certain subset of a collection of information can be replaced without changing the intended meaning. In propositional logics, for example, replacing a subformula ϕ of Φ with an equivalent one, say ϕ', yields a formula $\Phi[\phi/\phi']$ equivalent to Φ. That is, we may view ϕ as an independent module of Φ. Within the KR community it is folklore that this is usually not the case for non-monotonic logics (apart from folklore, we refer the reader to (Baumann and Strass 2016) for a rigorous study of this matter).

Motivated by this observation, the notion of strong equivalence was introduced in the literature. In a nutshell, strong equivalence requires the aforementioned property by design: $\mathcal{K}$ and $\mathcal{K}'$ are strongly equivalent if for any $\mathcal{H}$, the knowledge bases $\mathcal{K} \cup \mathcal{H}$ and $\mathcal{K}' \cup \mathcal{H}$ are equivalent. Although a naive implementation would require to iterate over an infinite number of possible $\mathcal{H}$, researchers discovered techniques to decide strong equivalence of two knowledge bases efficiently, most notably for logic programming (Lifschitz, Pearce, and Valverde 2001) and argumentation frameworks (AFs) (Oikarinen and Woltran 2011). In this paper, we extend this line of research to a recent extension of AFs, called Claim-augmented argumentation frameworks (CAFs).

Abstract argumentation frameworks as proposed by Dung (Dung 1995) in his seminal 1995 paper are by now a major research area in knowledge representation and reasoning. They have been thoroughly investigated since then and various extensions have been proposed in order to extend their expressive power. For example, researchers considered the addition of supports (Cayrol and Lagasquie-Schiex 2005), recursive (Baroni et al. 2011) and collective (Nielsen and Parsons 2006) attacks, or probabilities (Thimm 2012) to mention a few. CAFs as introduced by (Dvorák and Woltran 2020) provide means for conclusion-oriented reasoning in argumentation. While traditional argumentation formalisms focus on the identification of acceptable arguments, the emphasis in claim-augmented argumentation lies instead on the argument's conclusions (*claims*). Building on the basic observation that a claim can be supported by different arguments, it becomes evident that the traditional argument-focused perspective is often insufficient to capture claim-based reasoning. CAFs address this issue by extending AFs with a function which assigns a claim to each argument. They are in particular well-suited to analyze instantiation-based approaches, e.g., instantiations of logic programs (Caminada et al. 2015b), rule-based formalisms like ABA+ (Bondarenko, Toni, and Kowalski 1993; Caminada et al. 2015a), or logic-based instantiations (Besnard and Hunter 2001; Gorogiannis and Hunter 2011), where the focus lies on the claims of the arguments which have been constructed during the process.

The goal of this paper is to investigate equivalence notions for reasoning with a claim-centered point of view. Due to their generality, CAFs form an ideal basis to obtain a comprehensive study of this matter. Our main contributions are:

- We provide characterization results of strong equivalence between CAFs via semantics-dependent kernels for each CAF semantics which has been considered in the literature so far. Moreover, we discuss ordinary equivalence for CAFs and present dependencies between semantics for this weaker equivalence notion.

- We introduce novel equivalence concepts based on argument renaming which are genuine for CAFs. We show that ordinary equivalence up to renaming coincides with ordinary equivalence while strong equivalence up to renaming can be characterized via kernel isomorphism.
- We present a rigorous complexity analysis of deciding equivalence between two CAFs for all of the aforementioned equivalence notions. We show that deciding ordinary equivalence can be computationally hard, up to the third level of the polynomial hierarchy while strong equivalence is computationally tractable. Moreover, we show that strong equivalence up to renaming has the same complexity as the graph isomorphism problem.

Full proofs are available under https://www.dbai.tuwien.ac.at/research/report/dbai-tr-2021-122.pdf.

2 Background

Abstract Argumentation. We fix a non-finite background set $\mathcal{U}$. An argumentation framework (AF) (Dung 1995) is a directed graph $F = (A, R)$ where $A \subseteq \mathcal{U}$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. In this paper we consider finite AFs only.

For two arguments $a, b \in A$, if $(a, b) \in R$ we say that a *attacks* b as well as a *attacks* (the set) E given that $b \in E \subseteq A$. We frequently use the so-called *range* of a set E defined as $E^{\oplus}_F = E \cup E^{+}_F$ where $E^{+}_F = \{a \in A \mid E \text{ attacks } a\}$.

A set $E \subseteq A$ is conflict-free in F (for short, $E \in cf(F)$) iff for no $a, b \in E$, $(a, b) \in R$. A set E *defends* an argument a if any attacker of a is attacked by some argument of E. A *semantics* is a function $\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}}$ with $F \mapsto \sigma(F) \subseteq 2^A$. This means, given an AF $F = (A, R)$ a semantics returns a set of subsets of A. These subsets are called *σ-extensions*.

In this paper we consider so-called *naive*, *admissible*, *complete*, *grounded*, *preferred*, *stable*, *semi-stable* and *stage* semantics (abbr. na, ad, co, gr, pr, stb, ss, stg). Apart from naive, semi-stable and stage semantics (Verheij 1996; Caminada 2006), all mentioned semantics were already introduced in (Dung 1995).

Definition 2.1. Let $F = (A, R)$ be an AF and $E \in cf(F)$.

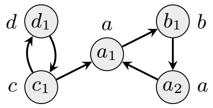
1. $E \in na(F)$ iff E is $\subseteq$-maximal in $cf(F)$,
2. $E \in ad(F)$ iff E defends all its elements,
3. $E \in co(F)$ iff $E \in ad(F)$ and for any a defended by E we have, $a \in E$,
4. $E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$, and
5. $E \in pr(F)$ iff E is $\subseteq$-maximal in $ad(F)$,
6. $E \in stb(F)$ iff E attacks any $a \in A \setminus E$,
7. $E \in ss(F)$ iff $E \in ad(F)$ and there is no $D \in ad(F)$ with $E^{\oplus}_F \subsetneq D^{\oplus}_F$,
8. $E \in stg(F)$ iff there is no $D \in cf(F)$ with $E^{\oplus}_F \subsetneq D^{\oplus}_F$.

Claim-based Argumentation. A *claim-augmented argumentation framework (CAF)* (Dvorák and Woltran 2020) is a triple $\mathcal{F} = (A, R, cl)$ where $F = (A, R)$ is an AF and $cl : A \to \mathcal{C}$ is a function which assigns a claim to each argument in A; $\mathcal{C}$ is a set of (countable infinite) possible claims. The claim-function is extended to sets in the natural way, i.e. for a set $E \subseteq A$, we let $cl(E) = \{cl(a) \mid a \in E\}$.

There are several ways in which semantics for AFs extend to CAFs. The most basic one is to choose an appropriate AF semantics and consider the claims of the induced extensions.

Definition 2.2. For a CAF $\mathcal{F} = (A, R, cl)$, $F = (A, R)$, and a semantics σ, we define the inherited variant of σ *(i-σ)* as $\sigma_c(\mathcal{F}) = \{cl(E) \mid E \in \sigma(F)\}$. We call $E \in \sigma(F)$ with $cl(E) = S$ a σ_c-realization of S in $\mathcal{F}$.

Example 2.3. Consider the following CAF $\mathcal{F}$:



Let us focus on stable semantics. For the underlying AF F we have the unique stable extension $E = \{c_1, b_1\}$. It is thus easy to see that $stb_c(\mathcal{F}) = \{\{c, b\}\}$. Moreover, $\{c_1, b_1\}$ is a stb_c-realization of E.

Let us now turn to the semantics which actually operate on the level of the claims instead of focusing on the underlying arguments. For this, we need to generalize the notion of defeat to claims. A set of arguments $E \subseteq A$ *defeats* a claim $c \in cl(A)$ in $\mathcal{F}$ if E attacks every $a \in A$ with $cl(a) = c$ (in F); we write $E^{\boldsymbol{+}}_{\mathcal{F}} = \{c \in cl(A) \mid E \text{ defeats } c \text{ in } \mathcal{F}\}$ to denote the set of all claims which are defeated by E in $\mathcal{F}$. The claim-range of a set of claims $S = cl(E)$ is denoted by $E^{\boldsymbol{\oplus}}_{\mathcal{F}} = cl(E) \cup E^{\boldsymbol{+}}_{\mathcal{F}}$.

Example 2.4. Consider again the CAF $\mathcal{F}$ from the previous example. Although c_1 defeats a_1, it does not defeat the claim a. However, $E = \{c_1, b_1\}$ defeats a, i.e. $a \in E^{\boldsymbol{+}}_{\mathcal{F}}$. The claim-range of E is thus $E^{\boldsymbol{\oplus}}_{\mathcal{F}} = \{a, b, c, d\}$.

Observe that the range of a set of claims is not a well-defined concept: In our example CAF $\mathcal{F}$, the claim-range of $\{a\}$ could either be $\{a, b\}$ induced by the realization $\{a_1\}$ or it could be $\{a\}$, which is induced by the realization $\{a_2\}$. Nonetheless, we can define semantics based on the claim-range by focusing on the underlying set E of arguments. We consider *cl-preferred*, *cl-naive*, *cl-cf-stable*, *cl-ad-stable*, *cl-semi-stable* and *cl-stage* semantics (abbr. *cl-pr*, *cl-na*, *cl-stb*$_{cf}$, *cl-stb*$_{ad}$, *cl-ss*, *cl-stg*) as introduced in (Rapberger 2020; Dvorák, Rapberger, and Woltran 2020a).

Definition 2.5. Let $\mathcal{F} = (A, R, cl)$ be a CAF with underlying AF $F = (A, R)$. For a set of claims $S \subseteq cl(A)$,

- $S \in$ *cl-pr*$(\mathcal{F})$ if S is $\subseteq$-maximal in $ad_c(\mathcal{F})$;
- $S \in$ *cl-na*$(\mathcal{F})$ if S is $\subseteq$-maximal in $cf_c(\mathcal{F})$;
- $S \in$ *cl-stb*$_\tau(\mathcal{F})$, $\tau \in \{cf, ad\}$, if there is a τ_c-realization E of S which defeats any $c \in cl(A) \setminus S$ (i.e., $E^{\boldsymbol{\oplus}}_{\mathcal{F}} = cl(A)$);
- $S \in$ *cl-ss*$(\mathcal{F})$ if there is an ad_c-realization E of S in $\mathcal{F}$ such that there is no $D \in ad(F)$ with $E^{\boldsymbol{\oplus}}_{\mathcal{F}} \subsetneq D^{\boldsymbol{\oplus}}_{\mathcal{F}}$;
- $S \in$ *cl-stg*$(\mathcal{F})$ if there is an cf_c-realization E of S in $\mathcal{F}$ such that there is no $D \in cf(F)$ with $E^{\boldsymbol{\oplus}}_{\mathcal{F}} \subsetneq D^{\boldsymbol{\oplus}}_{\mathcal{F}}$.

A set $E \subseteq A$ *cl-σ-realizes* the claim-set S in $\mathcal{F}$ if $cl(E) = S$ and E satisfies the respective requirements; e.g., $E \in cf(F)$ and $E^{\boldsymbol{\oplus}}_{\mathcal{F}} = cl(A)$ for cl-cf-stable semantics. We call E a *cl-σ*-realization of S in $\mathcal{F}$.

Example 2.6. Consider the semantics $\mathit{cl\text{-}stb}_{cf}$. We have that $S = \{c, b\} \in \mathit{cl\text{-}stb}_{cf}(\mathcal{F})$ since the realization $E = \{c_1, b_1\}$ for S has full claim-range as we already observed before. Moreover, $S' = \{d, a\} \in \mathit{cl\text{-}stb}_{cf}(\mathcal{F})$ as well: We consider the realization $E' = \{d_1, a_1\}$. The claims c and b are defeated by E' and hence, $E'^{\oplus}_{\mathcal{F}} = \{a, b, c, d\}$. Note that E' is not a stable extension of the underlying AF.

Basic relations between i-semantics carry over from AF semantics, e.g., $stb_c(\mathcal{F}) \subseteq ss_c(\mathcal{F}) \subseteq pr_c(CF) \subseteq co_c(CF) \subseteq ad_c(\mathcal{F}) \subseteq cf_c(\mathcal{F})$ and $stb_c(\mathcal{F}) \subseteq stg_c(\mathcal{F}) \subseteq na_c(\mathcal{F}) \subseteq cf_c(\mathcal{F})$. As shown in (Dvorák, Rapberger, and Woltran 2020a), we have $stb_c(\mathcal{F}) \subseteq \mathit{cl\text{-}stb}_{ad}(\mathcal{F}) \subseteq \mathit{cl\text{-}stb}_{cf}(\mathcal{F}) \subseteq \mathit{cl\text{-}stg}(\mathcal{F}) \subseteq na_c(\mathcal{F})$ and $\mathit{cl\text{-}stb}_{ad}(\mathcal{F}) \subseteq \mathit{cl\text{-}ss}(\mathcal{F}) \subseteq pr_c(\mathcal{F})$. Moreover, each cl-σ-claim-set of $\mathcal{F}$ is $\subseteq$-maximal in $\sigma_c(\mathcal{F})$ for $\sigma \in \{pr, na\}$.

Notation. We write $\mathcal{F} = (F, cl)$ as an abbreviation for $\mathcal{F} = (A, R, cl)$ with AF $F = (A, R)$ (similar for CAFs $\mathcal{G}$ or $\mathcal{H}$ for which we denote the corresponding AFs by G and H, respectively). Also, we use the subscript-notation $A_{\mathcal{F}}$, $R_{\mathcal{F}}$, $cl_{\mathcal{F}}$, and $F_{\mathcal{F}}$ to indicate the affiliations.

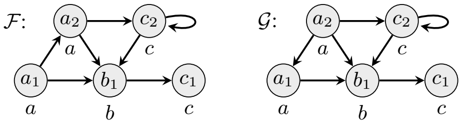
3 Equivalence in CAFs

In this section, we discuss ordinary and strong equivalence for CAFs. We introduce a novel kernel which characterizes strong equivalence for cl-cf-stable and cl-stage semantics; moreover, we show that the remaining semantics can be characterized via known kernels for AFs.

Let us start with ordinary equivalence of CAFs.

Definition 3.1. Two CAFs $\mathcal{F}$ and $\mathcal{G}$ are *ordinary equivalent* to each other w.r.t. a semantics ρ, in symbols $\mathcal{F} \equiv_o^\rho \mathcal{G}$, if $\rho(\mathcal{F}) = \rho(\mathcal{G})$.

Example 3.2. Consider the following CAFs $\mathcal{F}$ and $\mathcal{G}$:



Although $\mathcal{F}$ and $\mathcal{G}$ disagree only on the direction of the attack between the arguments a_1 and a_2, we observe that $\mathcal{F}$ and $\mathcal{G}$ are not ordinary equivalent under i-stable semantics: $stb_c(\mathcal{F}) = \emptyset$ while $\mathcal{G}$ has the unique i-stable claim-set $\{a, c\}$ witnessed by the stable extension $\{a_2, c_1\}$ of G.

If we consider instead cl-stable semantics, we observe that the two CAFs agree on their outcome: First notice that $\{a, c\}$ is also cl-ad-stable (cl-cf-stable) in $\mathcal{G}$ (every stb_c-realization is admissible and has full claim-range). Moreover, we have that $\{a, c\}$ is also cl-ad-stable (cl-cf-stable) in $\mathcal{F}$ since the set $\{a_1, c_1\}$ is admissible and defeats every remaining claim. As a side remark, we mention that the claim-set $\{a, c\}$ has two realizations in $\mathcal{F}$ and $\mathcal{G}$ since both of the sets $\{a_1, c_1\}$, $\{a_2, c_1\}$ are conflict-free and have full claim-range. We obtain that the CAFs $\mathcal{F}$ and $\mathcal{G}$ are ordinary equivalent with respect to $\mathit{cl\text{-}stb}_{ad}$ and $\mathit{cl\text{-}stb}_{cf}$ semantics.

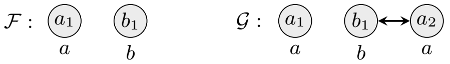
There are only few relations between the semantics for ordinary equivalence. We summarize them as follows:

Proposition 3.3. *For any two CAFs $\mathcal{F}$ and $\mathcal{G}$,*

- $\mathcal{F} \equiv_o^\rho \mathcal{G} \Rightarrow \mathcal{F} \equiv_o^{cl\text{-}pr} \mathcal{G}$, $\rho \in \{ad_c, pr_c\}$;
- $\mathcal{F} \equiv_o^{co_c} \mathcal{G} \Rightarrow \mathcal{F} \equiv_o^\rho \mathcal{G}$, $\rho \in \{gr_c, cl\text{-}pr\}$;
- $\mathcal{F} \equiv_o^{cf_c} \mathcal{G} \Leftrightarrow \mathcal{F} \equiv_o^{cl\text{-}na} \mathcal{G}$;
- $\mathcal{F} \equiv_o^{na_c} \mathcal{G} \Rightarrow \mathcal{F} \equiv_o^\rho \mathcal{G}$, $\rho \in \{cf_c, cl\text{-}na\}$.

Interestingly, we observe that the relations for AF semantics presented in (Oikarinen and Woltran 2011) do not carry over to inherited semantics. This is due to the fact that i-preferred (i-naive) semantics are not necessarily $\subseteq$-maximal i-admissible (i-conflict-free) claim-sets; for CAFs, this role is instead taken over by cl-preferred (cl-naive) semantics.

Example 3.4. Assume we are given two CAFs as follows:

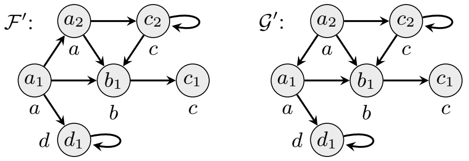


Clearly, $ad_c(\mathcal{F}) = ad_c(\mathcal{G}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. On the other hand, $\{a, b\}$ is the unique i-preferred claim-set of $\mathcal{F}$ while $pr_c(\mathcal{G}) = \{\{a\}, \{a, b\}\}$ witnessed by the extensions $\{a_1, a_2\}$ and $\{a_1, b_1\}$. Thus $\mathcal{F} \equiv_o^{ad_c} \mathcal{G} \not\Rightarrow \mathcal{F} \equiv_o^{pr_c} \mathcal{G}$. The example furthermore shows $\mathcal{F} \equiv_o^{cf_c} \mathcal{G} \not\Rightarrow \mathcal{F} \equiv_o^{na_c} \mathcal{G}$ since cf_c and ad_c as well as the respective variants of naive and preferred semantics coincide in $\mathcal{F}$ and $\mathcal{G}$.

The relations presented in Proposition 3.3 follow since cl-preferred claim-sets are $\subseteq$-maximal in $ad_c(\mathcal{F})$, $co_c(\mathcal{F})$ and $pr_c(\mathcal{F})$ for any CAF $\mathcal{F}$; moreover, the i-grounded claim-set is the $\subseteq$-minimal i-complete extension. Similar observations hold for conflict-free and naive semantics; additionally, we observe that $\mathcal{F} \equiv_o^\rho \mathcal{G}$, $\rho \in \{cl\text{-}na, na_c\}$, implies $\mathcal{F} \equiv_o^{cf_c} \mathcal{G}$ since cf_c semantics satisfies downward closure (every subset of a conflict-free set is conflict-free). We can construct counter-examples for the remaining cases.

A crucial observation is that ordinary equivalence is not robust when it comes to expansion of the frameworks, e.g., if an update in the knowledge base induces new arguments or attacks. Let us illustrate this at the following example:

Example 3.5. Assume we are given an updated version of $\mathcal{F}$ and $\mathcal{G}$ from Example 3.2 where an additional argument has been introduced. Let $\mathcal{F}'$ and $\mathcal{G}'$ be given as follows:



$\mathcal{F}'$ and $\mathcal{G}'$ no longer agree on their cl-ad-stable claim-sets: In $\mathcal{G}'$, the set $\{a_2, c_1\}$ does not defeat claim d, thus $\mathit{cl\text{-}stb}_{ad}(\mathcal{G}') = \emptyset$ while $\{a, c\}$ remains cl-ad-stable in $\mathcal{F}'$.

Let us introduce a stronger notion of equivalence which addresses such situations. We say that two CAFs are *strongly equivalent* to each other if they possess the same extensions independently of any such (simultaneous) expansions of the frameworks. Before we can define this notion formally, we require an additional concept which ensures that the expansion of the frameworks is well-defined.

Definition 3.6. Two CAFs $\mathcal{F}$ and $\mathcal{G}$ are *compatible* to each other if $cl_{\mathcal{F}}(a) = cl_{\mathcal{G}}(a)$ for all $a \in A_{\mathcal{F}} \cap A_{\mathcal{G}}$. The union $\mathcal{F} \cup \mathcal{G}$ of two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$ is defined componentwise, i.e., $\mathcal{F} \cup \mathcal{G} = (A_{\mathcal{F}} \cup A_{\mathcal{G}}, R_{\mathcal{F}} \cup R_{\mathcal{G}}, cl_{\mathcal{F}} \cup cl_{\mathcal{G}})$.

We are ready to introduce strong equivalence for CAFs.

Definition 3.7. Two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$ are *strongly equivalent* to each other w.r.t. a semantics ρ, in symbols $\mathcal{F} \equiv_s^\rho \mathcal{G}$, iff $\rho(\mathcal{F} \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$ for each CAF $\mathcal{H}$ which is compatible with $\mathcal{F}$ and $\mathcal{G}$.

The definition extends strong equivalence for AFs. We write $F \equiv_s^\sigma G$ to denote strong equivalence of two AFs F and G w.r.t. the semantics σ.

Strong equivalence for AFs has been characterized via syntactic equivalence of so-called (semantics-dependent) kernels. Let us recall the definitions of the stable and the naive kernel (Oikarinen and Woltran 2011; Baumann, Linsbichler, and Woltran 2016) as they exhibit interesting overlaps with our novel kernel for cl-cf-stable semantics.

Definition 3.8. For an AF $F = (A, R)$, we define the *stable kernel* $F^{sk} = (A, R^{sk})$ with

$$R^{sk} = R \setminus \{(a,b) \mid a \neq b, (a,a) \in R\};$$

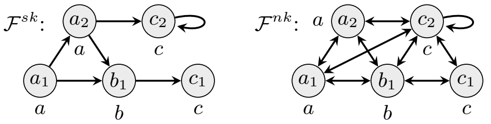
and the *naive kernel* $F^{nk} = (A, R^{nk})$ with

$$R^{nk} = R \cup \{(a,b) \mid a \neq b, \{(a,a),(b,b),(b,a)\} \cap R \neq \emptyset\}.$$

For a CAF $\mathcal{F} = (F, cl)$, we write $\mathcal{F}^{sk}$ ($\mathcal{F}^{nk}$) to denote (F^{sk}, cl) ((F^{nk}, cl), respectively).

The stable kernel characterizes strong equivalence for stable and stage semantics, i.e., $F \equiv_s^\sigma G$ iff $F^{sk} = G^{sk}$ for $\sigma \in \{stb, stg\}$ (Oikarinen and Woltran 2011); similarly, $F \equiv_s^\sigma G$ iff $F^{nk} = G^{nk}$ for $\sigma \in \{cf, na\}$ (Baumann, Linsbichler, and Woltran 2016).

Example 3.9. For the CAF $\mathcal{F}$ from Example 3.2, the stable kernel $\mathcal{F}^{sk}$ and the naive kernel $\mathcal{F}^{nk}$ are given as follows:



In the remaining part of this section, we characterize strong equivalence for all semantics under consideration by identifying appropriate kernels. Let us start with cl-cf-stable semantics. An interesting observation is that the CAFs $\mathcal{F}'$ and $\mathcal{G}'$ from Example 3.5 yield the same cl-cf-stable claim-sets even after the argument d_1 has been added. In fact, it can be shown that $\mathcal{F}$ and $\mathcal{G}$ yield the same cl-cf-stable claim-sets under any possible expansion. The reason is that the direction of the attack between a_1 and a_2 is irrelevant since both arguments possess the same claim a. Thus it suffices to include one of them in a cl-cf-stable claim-set in case not both of them are attacked.

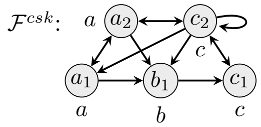
Let us now introduce the cf-stable kernel for CAFs.

Definition 3.10. For a CAF $\mathcal{F} = (A, R, cl)$, we define the cf-stable kernel as $\mathcal{F}^{csk} = (A, R^{csk}, cl)$ with

$$\begin{aligned} R^{csk} = R \cup \{(a,b) \mid a \neq b,\\ (a,a) \in R \vee (cl(a) = cl(b) \wedge \{(b,a),(b,b)\} \cap R \neq \emptyset)\}. \end{aligned}$$

We denote the underlying AF (A, R^{csk}) by F^{csk}.

Example 3.11. Consider again our previous CAF $\mathcal{F}$. We construct the cf-stable kernel $\mathcal{F}^{csk}$ of $\mathcal{F}$ as follows:



Remark 3.12. The cf-stable kernel consists of a combination of the stable and the naive kernel for AFs, where the claim-independent part stems from the stable kernel while the case where two arguments have the same claim relates to the naive kernel. In a nutshell, it is safe to introduce attacks (a, b), $a \neq b$ where a is self-attacking without changing stable semantics because attacks of this form neither interfere with the conflict-free extensions of an AF nor change the range of a conflict-free set. In case two arguments have the same claim, it is irrelevant which of these arguments is included in an extension. It is thus safe to introduce attacks between two arguments in case their union is conflicting.

In what follows, we will prove that the cf-kernel characterizes strong equivalence for claim-level cf-stable and stage semantics. To this end we will first discuss some general observations. The following lemma states that two CAFs having different arguments are not strongly equivalent.

Lemma 3.13. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $A_{\mathcal{F}} \neq A_{\mathcal{G}}$ implies $\mathcal{F} \not\equiv_s^\rho \mathcal{G}$ for any considered semantics ρ.*

Proof. W.l.o.g., we may assume that there is $a \in A_{\mathcal{F}}$ with $a \notin A_{\mathcal{G}}$. To prove the statement, we distinguish the following cases: (a) $(a,a) \notin R_{\mathcal{F}}$ and (b) $(a,a) \in R_{\mathcal{F}}$. We present the construction for case (a): For a fresh argument x and a fresh claim c, let $\mathcal{H} = (A_{\mathcal{H}}, R_{\mathcal{H}}, cl_{\mathcal{H}})$ with

$$\begin{aligned} A_{\mathcal{H}} &= (A_{\mathcal{F}} \cup A_{\mathcal{G}} \cup \{x\}) \setminus \{a\};\\ R_{\mathcal{H}} &= \{(x,b) \mid b \in (A_{\mathcal{F}} \cup A_{\mathcal{G}}) \setminus \{a\}\}; \end{aligned}$$

and $cl_{\mathcal{H}}(b) = cl_{\mathcal{F}}(b)$ for $b \in A_{\mathcal{F}} \cup A_{\mathcal{G}}$ and $cl_{\mathcal{H}}(x) = c$; that is, we introduce a new argument having a fresh claim c which attacks every argument except a. It can be checked that $\{cl_{\mathcal{H}}(a), c\} \in \rho(\mathcal{F} \cup \mathcal{H})$ for every semantics under consideration. Observe that $\{cl_{\mathcal{H}}(a), c\}$ is not a claim-extension under any semantics in $\mathcal{G} \cup \mathcal{H}$ since a is not present in $\mathcal{G} \cup \mathcal{H}$ and x does attack every remaining argument. □

The following lemma implies that two strongly equivalent CAFs $\mathcal{F}$ and $\mathcal{G}$ possess the same self-attacking arguments.

Lemma 3.14. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $(a,a) \in R_{\mathcal{F}} \Delta R_{\mathcal{G}}$ implies $\mathcal{F} \not\equiv_s^\rho \mathcal{G}$ for any semantics ρ under consideration.*

The following lemma states that a CAF admits the same cl-cf-stable (cl-stage) claim-sets as its cf-stable kernel.

Lemma 3.15. *For any CAF $\mathcal{F}$, $\rho(\mathcal{F}) = \rho(\mathcal{F}^{csk})$ for the semantics $\rho \in \{cl\text{-}stb_{cf}, cl\text{-}stg\}$.*

Moreover, it can be shown that syntactic equivalence of cf-stable kernels of two CAFs $\mathcal{F}$ and $\mathcal{G}$ implies that the kernels coincide under any possible expansion.

Lemma 3.16. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F}^{csk} = \mathcal{G}^{csk}$ implies $(\mathcal{F} \cup \mathcal{H})^{csk} = (\mathcal{G} \cup \mathcal{H})^{csk}$ for any CAF $\mathcal{H}$ compatible with $\mathcal{F}$ and $\mathcal{G}$.*

We are now ready to prove our first main result stating that two CAFs $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent to each other w.r.t. cl-cf-stable and cl-stage semantics if and only if their cl-stable kernels coincide.

Theorem 3.17. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F}^{csk} = \mathcal{G}^{csk}$ iff $\mathcal{F} \equiv^{\rho}_{s} \mathcal{G}$ for $\rho \in \{cl\text{-}stb_{cf}, cl\text{-}stg\}$.*

Proof. First suppose we have $\mathcal{F}^{csk} = \mathcal{G}^{csk}$. In this case, $(\mathcal{F} \cup \mathcal{H})^{csk} = (\mathcal{G} \cup \mathcal{H})^{csk}$ for any compatible CAF $\mathcal{H}$ by Lemma 3.16. We infer $\rho(\mathcal{F} \cup \mathcal{H}) = \rho((\mathcal{F} \cup \mathcal{H})^{csk})$ as well as $\rho((\mathcal{G} \cup \mathcal{H})^{csk}) = \rho(\mathcal{G} \cup \mathcal{H})$ from Lemma 3.15. Hence $\mathcal{F} \equiv^{\rho}_{s} \mathcal{G}$ follows.

Now suppose $\mathcal{F}^{csk} \neq \mathcal{G}^{csk}$. Due to Lemma 3.15 we may assume $\rho(\mathcal{F}^{csk}) = \rho(\mathcal{G}^{csk})$; moreover, $A_{\mathcal{F}} = A_{\mathcal{G}} (= A)$ by Lemma 3.13. We thus have that $R_{\mathcal{F}^{csk}} \neq R_{\mathcal{G}^{csk}}$. W.l.o.g., let $(a,b) \in R_{\mathcal{F}^{csk}} \setminus R_{\mathcal{G}^{csk}}$; we apply Lemma 3.14 to assume $a \neq b$. Moreover, observe that $(a,a) \notin R^{csk}_{\mathcal{G}}$ (and thus, $(a,a) \notin R^{csk}_{\mathcal{F}}$) since otherwise $(a,b) \in R_{\mathcal{G}^{csk}}$ by definition of the cf-stable kernel. We distinguish the following cases: (a) $cl(a) \neq cl(b)$, and (b) $cl(a) = cl(b)$.

(a) In case $cl(a) \neq cl(b)$, consider two newly introduced arguments x, y and fresh claims c, d. We consider the AF $\mathcal{H}_1 = (A \cup \{x,y\}, R_1, cl_1)$ where
$$R_1 = \{(x,y)\} \cup \{(y,h) \mid h \in A \cup \{x\}\} \cup \{(x,h) \mid h \in A \setminus \{a,b\}\},$$
and the function cl_1 is given as follows: $cl_1(x) = c$, $cl_1(y) = d$, and the other claims coincide with the given ones, i.e. $cl_1(h) = cl_{\mathcal{F}}(h)$ if $h \in A$. First observe that $\{d\}$ is i-stable in both $\mathcal{F}^{csk} \cup \mathcal{H}_1$ and $\mathcal{G}^{csk} \cup \mathcal{H}_1$ and thus guarantees that $\rho(\mathcal{F}^{csk} \cup \mathcal{H}_1)$ and $\rho(\mathcal{G}^{csk} \cup \mathcal{H}_1)$ are non-empty. It can be checked that $S = \{cl(a), c\}$ is cl-cf-stable and cl-stage in $\mathcal{F}^{csk} \cup \mathcal{H}_1$ (since $\{a,x\}$ is stable); on the other hand, $S \notin \rho(\mathcal{G}^{csk} \cup \mathcal{H}_1)$ since b is not defeated by $\{a,x\}$. However, this is our only candidate since S has no other cf-realization in $\mathcal{G}^{csk} \cup \mathcal{H}_1$.

(b) Now consider the case $cl(a) = cl(b)$ and observe that $(a,a), (b,b), (b,a) \notin R^{csk}_{\mathcal{G}}$ (otherwise $(a,b) \in R_{\mathcal{G}^{csk}}$). Since $\mathcal{F}$ and $\mathcal{G}$ contain the same self-attacks, we furthermore have $(a,a), (b,b) \notin R_{\mathcal{F}^{csk}}$. Having established this situation let us construct $\mathcal{H}_2$ as follows: For fresh arguments x, y, z and fresh claims c, d, e, we consider $\mathcal{H}_2 = (A \cup \{x,y,z\}, R_2, cl_2)$ where
$$R_2 = \{(a,h) \mid h \in (A \cup \{x\}) \setminus \{a,b\}\} \cup \{(a,x),(x,x),(b,y),(y,y),(z,b),(b,z),(z,y)\}$$
and as before we let $cl_2(h) = cl_{\mathcal{F}}(h)$ for $h \in A$; for the fresh arguments let $cl_2(x) = c$, $cl_2(y) = d$, as well as $cl_2(z) = e$. It can be checked that each CAF admits a stable extension; thus it suffices to show that the cl-cf-stable claim-sets disagree. First observe that we now have $\{cl_2(a)\} \in \rho(\mathcal{G}^{csk} \cup \mathcal{H}_2)$ since $\{a,b\}$ is a stable extension in $G^{csk} \cup H_2$. On the other hand, we have that $\{cl_2(a)\}$ is neither $cl\text{-}stb_{cf}$-realizable nor $cl\text{-}stg$-realizable in $\mathcal{F}^{csk} \cup \mathcal{H}_2$.

In every case, we have found some $\mathcal{H}$ enforcing inequality, i.e. $\rho(\mathcal{F}^{csk} \cup \mathcal{H}) \neq \rho(\mathcal{G}^{csk} \cup \mathcal{H})$. By Lemma 3.15, we get $\rho(\mathcal{F} \cup \mathcal{H}) = \rho((\mathcal{F} \cup \mathcal{H})^{csk}) = \rho(\mathcal{F}^{csk} \cup \mathcal{H}) \neq \rho(\mathcal{G}^{csk} \cup \mathcal{H}) = \rho((\mathcal{G} \cup \mathcal{H})^{csk}) = \rho(\mathcal{G} \cup \mathcal{H})$. It follows that $\mathcal{F} \not\equiv^{s}_{\rho} \mathcal{G}$. □

The remaining semantics under consideration can be characterized via known AF kernels. We recall the AF kernels from the literature (Oikarinen and Woltran 2011).

Definition 3.18. For an AF $F = (A,R)$, we define the *admissible kernel* $F^{ak} = (A, R^{ak})$ with
$$R^{ak} = R \setminus \{(a,b) \mid a \neq b, (a,a) \in R, \{(b,a),(b,b)\} \cap R \neq \emptyset\};$$
the *complete kernel* $F^{gk} = (A, R^{gk})$ with
$$R^{ck} = R \setminus \{(a,b) \mid a \neq b, (a,a),(b,b) \in R\};$$
and the *grounded kernel* $F^{gk} = (A, R^{gk})$ with
$$R^{gk} = R \setminus \{(a,b) \mid a \neq b, (b,b) \in R, \{(b,a),(a,a)\} \cap R \neq \emptyset\}.$$

It has been shown that the grounded (complete) kernel characterizes strong equivalence for grounded (complete) semantics; moreover, for any two AFs F and G we have $F \equiv^{\sigma}_{s} G$ iff $F^{ak} = G^{ak}$ for $\sigma \in \{ad, pr, ss\}$ (Oikarinen and Woltran 2011). We write $F^{k(\rho)}$ to denote the kernel which characterizes strong equivalence for the semantics ρ.

To prove that strong equivalence for the remaining semantics can be characterized using known AF kernels, we make use of the following lemma which states that each CAF $\mathcal{F}$ has the same σ_c-claim-sets as its kernel $\mathcal{F}^{k(\sigma)}$ for any AF semantics σ under consideration; moreover, the cl-ad-stable and cl-semi-stable claim-sets of $\mathcal{F}$ and $\mathcal{F}^{ak}$ coincide.

Lemma 3.19. *For any CAF $\mathcal{F}$, (a) $\sigma_c(\mathcal{F}^{k(\sigma)}) = \sigma_c(\mathcal{F})$ for any considered AF semantics σ; and (b) $\rho(\mathcal{F}) = \rho(\mathcal{F}^{ak})$ for $\rho \in \{cl\text{-}stb_{ad}, cl\text{-}ss\}$.*

For inherited semantics, the result is immediate by known results for AFs; for cl-ad-stable and cl-semi-stable semantics, the statement follows by the additional observation that the range of every admissible set of F remains unchanged.

It can be shown that two CAFs are strongly equivalent under cl-ad-stable and cl-semi-stable semantics iff their admissible kernels coincide.

Theorem 3.20. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv^{\rho}_{s} \mathcal{G}$ iff $F^{ak} = G^{ak}$ for $\rho \in \{cl\text{-}stb_{ad}, cl\text{-}ss\}$.*

Moreover, each inherited semantics σ_c can be characterized by the respective kernel for σ.

Theorem 3.21. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv^{\sigma_c}_{s} \mathcal{G}$ iff $F \equiv^{\sigma}_{s} G$ for any considered AF semantics σ.*

Due to space limits, we shall omit the proofs of the above theorems. The proofs proceed in the same way as the proof of Theorem 3.17; first, we use Lemma 3.19 to show $F^{k(\rho)} = G^{k(\rho)}$ implies strong equivalence of two CAFs $\mathcal{F}$ and $\mathcal{G}$ w.r.t. ρ for the respective kernels $F^{k(\rho)}$ and $G^{k(\rho)}$. For the other direction, we assume that the kernels of $\mathcal{F}$ and $\mathcal{G}$ differ. Depending on the semantics, we consider different cases for which we construct a CAF $\mathcal{H}$ which serves as a witness to show $\mathcal{F} \not\equiv^{\rho}_{s} \mathcal{G}$.

For cl-naive and cl-preferred semantics, it can be shown that strong equivalence w.r.t. cl-naive and cl-preferred semantics coincides with strong equivalence w.r.t. their inherited counterparts. This implies that two CAFs are strongly equivalent w.r.t. cl-preferred semantics iff their admissible kernels coincide; likewise, two CAFs are strongly equivalent w.r.t. cl-naive semantics iff their naive kernels coincide.

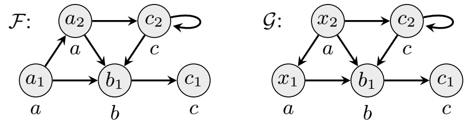
Theorem 3.22. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{cl\text{-}\sigma} \mathcal{G}$ iff $\mathcal{F} \equiv_s^{\sigma_c} \mathcal{G}$ for $\sigma \in \{na, pr\}$.*

The proof proceeds in a slightly different way: To show $\mathcal{F} \not\equiv_s^{\sigma_c} \mathcal{G}$ implies $\mathcal{F} \not\equiv_s^{cl\text{-}\sigma} \mathcal{G}$, it can be assumed that $\mathcal{F}$ and $\mathcal{G}$ disagree on their σ_c claim-sets. We construct counterexamples $\mathcal{H}$ satisfying $cl\text{-}\sigma(\mathcal{F} \cup \mathcal{H}) \neq cl\text{-}\sigma(\mathcal{G} \cup \mathcal{H})$ in such a way that the claim-set which does not appear in either one of the frameworks becomes a $\subseteq$-maximal σ_c-claim-extension.

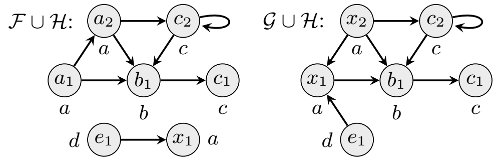
4 Renaming and Equivalence

In the previous section we were assuming that we are interested in the actual arguments and not just the claims and their interactions. In this section, we will also provide another point of view which entirely abstracts from the underlying arguments and thus viewing a CAF as a collection of claims and their relationships. To illustrate this, let us consider the following example.

Example 4.1. Assume we are given again our CAF $\mathcal{F}$ from Example 3.2 together with a CAF $\mathcal{G}$ as follows:



We observe that both CAFs are equivalent w.r.t. cl-cf-stable semantics although the arguments a_1 and a_2 are not even present in $\mathcal{G}$ while the same is true for x_1 and x_2 in $\mathcal{F}$. Moreover, recalling the kernel for $cl\text{-}stb_{cf}$ from Theorem 3.17 we observe that $\mathcal{F}$ and $\mathcal{G}$ would be even strongly equivalent if this mismatch in argument names were not present. This suggests that the usual notion of strong equivalence does not handle situations where we are interested in claims only very well. To illustrate this with a hands-on situation let us suppose we are given $\mathcal{H}$ in a way that a novel argument e_1 with claim e is given which attacks x_1:



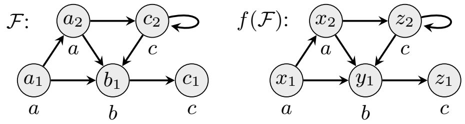
This is fine when insisting on the arguments, but on a claim-level one could of course argue that $\mathcal{H}$ did not yield the same modification on both sides and thus disrupts the similarity between $\mathcal{F}$ and $\mathcal{G}$ in an unintended way.

Our goal is hence to develop notions of equivalence which handle situations like the aforementioned one in a more intuitive way. The first step to formalize the underlying idea is the following notion of a renaming.

Definition 4.2. For a CAF $\mathcal{F}$ and an arbitrary set A' of arguments we call a bijection $f : A_\mathcal{F} \to A'$ s.t. for each $a \in A_\mathcal{F}$ we have $cl_\mathcal{F}(a) = cl_\mathcal{F}(f(a))$ a *renaming for* $\mathcal{F}$.

We abuse notation and write $f(\mathcal{F})$ for the CAF obtained from renaming the arguments, i.e. $f(\mathcal{F})$ is the CAF $(f(F), cl_f) := (f(A), R_f, cl_f)$ where $(a, b) \in R_f$ iff $(f^{-1}(a), f^{-1}(b)) \in R_\mathcal{F}$ and $cl_f(f(a)) = cl_\mathcal{F}(a)$.

Example 4.3. Consider again our previous CAF $\mathcal{F}$. Let us assume we are given $A' = \{x_1, x_2, y_1, z_1, z_2\}$. The renaming f with $a_i \mapsto x_i$, $b_1 \mapsto y_1$ and $c_i \mapsto z_i$ induces the following CAF $f(\mathcal{F})$:



We observe that f does not change the structure of $\mathcal{F}$ on claim-level. In particular, $cl\text{-}stb_{cf}(\mathcal{F}) = cl\text{-}stb_{cf}(f(\mathcal{F}))$.

The last observation we made was no coincidence in the specific situation. More precisely, for the semantics considered in this paper, renaming does not change the meaning of our CAF.

Proposition 4.4. *For a CAF $\mathcal{F}$, an arbitrary set A' of arguments and a renaming f we have $\rho(\mathcal{F}) = \rho(f(\mathcal{F}))$ for any semantics ρ considered in this paper.*

Proof. We have $E \in \sigma(F)$ iff $f(E) \in \sigma(f(F))$ for the underlying AF and since all semantics are defined by selecting (subsets of) $\{cl(E) \mid E \in \sigma(F)\}$, the claim follows since $cl_\mathcal{F}(a) = cl_\mathcal{F}(f(a))$ for each argument a. □

Having formally established that names of arguments do not change the given semantics, let us proceed with defining notions of equivalence that build upon this insight.

Definition 4.5. Two CAFs $\mathcal{F}$ and $\mathcal{G}$ are *ordinary equivalent up to renaming* to each other w.r.t. a semantics ρ, in symbols $\mathcal{F} \equiv_{or}^{\rho} \mathcal{G}$, if there is some set A of arguments and some renaming $f : A_\mathcal{F} \to A$ for $\mathcal{F}$ s.t. $\rho(f(\mathcal{F})) = \rho(\mathcal{G})$.

So, informally speaking, Definition 4.5 requires that $\mathcal{F}$ and $\mathcal{G}$ are equivalent, at least after the underlying arguments are relabeled in a suitable way. However, in Proposition 4.4 we have actually already established that this adjustment is superfluous for our semantics. More formally, we infer the following result.

Proposition 4.6. *For any two CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_{or}^{\rho} \mathcal{G}$ iff $\mathcal{F} \equiv_{o}^{\rho} \mathcal{G}$ for any semantics ρ under consideration.*

Considering this result, it becomes apparent that we could also require that $\rho(f(\mathcal{F})) = \rho(\mathcal{G})$ holds for any renaming, not just for one in particular.

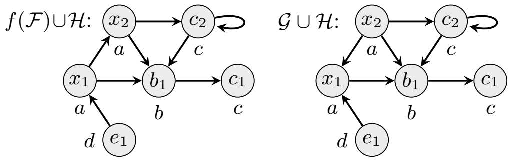
Proposition 4.7. *For two CAFs $\mathcal{F}$ and $\mathcal{G}$ we have that for all semantics considered in this paper $\mathcal{F} \equiv_{or}^{\rho} \mathcal{G}$ implies $\rho(f(\mathcal{F})) = \rho(\mathcal{G})$ for any renaming f for $\mathcal{F}$.*

Now we utilize the notion of a renaming in order to define a strong equivalence-like relation which is more suitable than strong equivalence for situations like the one described in Example 4.1.

Definition 4.8. Two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$ are *strongly equivalent up to renaming* to each other w.r.t. a semantics ρ, in symbols $\mathcal{F} \equiv^{\rho}_{sr} \mathcal{G}$, if there is a renaming $f : A_{\mathcal{F}} \to A_{\mathcal{F}}$ for $\mathcal{F}$ s.t. $\rho(f(\mathcal{F}) \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$ for each CAF $\mathcal{H}$ which is compatible with $\mathcal{F}$ and $\mathcal{G}$.

Let us reconsider our motivating Example 4.1.

Example 4.9. Recall the CAFs $\mathcal{F}$ and $\mathcal{G}$ from before and consider a renaming f which maps a_i to x_i and leaves the remaining arguments unchanged. Augmenting both $f(\mathcal{F})$ and $\mathcal{G}$ with $\mathcal{H}$, we obtain the following desired situation:



Notice that Proposition 4.4 ensures that our renaming for $\mathcal{F}$ only prevents $\mathcal{H}$ from introducing a novel argument, while preserving the semantics of $\mathcal{F}$.

Strong equivalence up to renaming implies the usual strong equivalence. This can be obtained by setting $f = id$.

Proposition 4.10. *For any two CAFs $\mathcal{F}$ and $\mathcal{G}$, if $\mathcal{F} \equiv^{\rho}_{s} \mathcal{G}$, then $\mathcal{F} \equiv^{\rho}_{sr} \mathcal{G}$.*

Even without using Proposition 4.4 explicitly we can infer that strong equivalence survives moving to a renamed version of f as well.

Proposition 4.11. *For any two compatible CAFs $\mathcal{F}$ and $\mathcal{G}$, if $\mathcal{F} \equiv^{\rho}_{sr} \mathcal{G}$, then $f(\mathcal{F}) \equiv^{\rho}_{sr} \mathcal{G}$ for any renaming f for $\mathcal{F}$.*

Proof. We have $\rho(g(\mathcal{F}) \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$ for each $\mathcal{H}$ for some renaming g because we assume $\mathcal{F} \equiv^{\rho}_{sr} \mathcal{G}$. Since f is a bijection we find $\rho(g(f^{-1}(f(\mathcal{F}))) \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$, thus $g \circ f^{-1}$ is our witnessing renaming for $f(\mathcal{F}) \equiv^{\rho}_{sr} \mathcal{G}$. □

Let us now come to the kernels. Since our notion of strong equivalence up to renaming allows for changing the names of the arguments, we expect our kernels to behave similarly. More specifically, we also need to consider renamed versions of the CAFs before evaluating the kernels. However, checking strong equivalence up to renaming will surely require to take the structure of the CAFs into consideration. We thus define what we mean by a CAF isomorphism.

Definition 4.12. Two CAFs $\mathcal{F}$ and $\mathcal{G}$ are *isomorphic* to each other iff there is a mapping $f : A_{\mathcal{F}} \to A_{\mathcal{G}}$ s.t. (1) f is a renaming for $\mathcal{F}$ and (2) for all $a, b \in A_{\mathcal{F}}$, $(a, b) \in R_{\mathcal{F}}$ iff $(f(a), f(b)) \in R_{\mathcal{G}}$. f is called *isomorphism* between $\mathcal{F}$, $\mathcal{G}$.

CAFs $\mathcal{F}$ and $f(\mathcal{F})$ from Example 4.3 are isomorphic. The given renaming f naturally is a CAF-isomorphism between $\mathcal{F}$ and $f(\mathcal{F})$. The following proposition collects basic properties of CAF isomorphisms.

Proposition 4.13. *For any two CAFs $\mathcal{F}$ and $\mathcal{G}$, (a) if $\mathcal{F}$ and $\mathcal{G}$ are isomorphic, then $\rho(\mathcal{F}) = \rho(\mathcal{G})$ for any considered semantics ρ; and (b) if f is a renaming for $\mathcal{F}$, then $\mathcal{F}$ and $f(\mathcal{F})$ are isomorphic.*

As it turns out, we obtain *exactly* the result we desire to: We check strong equivalence up to renaming by choosing the appropriate kernel for ρ, computing the kernels of $\mathcal{F}$ and $\mathcal{G}$ and then checking whether those are isomorphic to each other. Informally speaking, our tailored notion of equivalence which does not take the names of arguments into account yields the exact same kernels after relabeling the arguments in a suitable way.

Theorem 4.14. *For any two CAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv^{\rho}_{sr} \mathcal{G}$ iff $\mathcal{F}^{k(\rho)}$ and $\mathcal{G}^{k(\rho)}$ are isomorphic.*

Proof. ($\Leftarrow$) Let $\mathcal{F}^{k(\rho)}$ and $\mathcal{G}^{k(\rho)}$ be isomorphic, witnessed by the isomorphism f. We have $f(\mathcal{F}^{k(\rho)}) = \mathcal{G}^{k(\rho)}$; moreover, $\mathcal{F}^{k(\rho)} = \mathcal{G}^{k(\rho)}$ implies $(\mathcal{F} \cup \mathcal{H})^{k(\rho)} = (\mathcal{G} \cup \mathcal{H})^{k(\rho)}$ for any compatible CAF $\mathcal{H}$; extending f to $\mathcal{H}$ in a straightforward way yields $f((\mathcal{F} \cup \mathcal{H})^{k(\rho)}) = (\mathcal{G} \cup \mathcal{H})^{k(\rho)}$. Since $(\mathcal{F} \cup \mathcal{H})^{k(\rho)} = (\mathcal{G} \cup \mathcal{H})^{k(\rho)}$ implies $\rho(\mathcal{F} \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$ our isomorphism ensures $\rho(\mathcal{F} \cup \mathcal{H}) = \rho(\mathcal{G} \cup \mathcal{H})$.

($\Rightarrow$) Now assume the kernels $\mathcal{F}^{k(\rho)}$ and $\mathcal{G}^{k(\rho)}$ are not isomorphic, i.e. for any renaming f, $f(\mathcal{F}^{k(\rho)}) \neq \mathcal{G}^{k(\rho)}$. Due to the properties of our kernel, there is some $\mathcal{H}$ s.t. $\rho(f(\mathcal{F}) \cup \mathcal{H}) \neq \rho(\mathcal{G} \cup \mathcal{H})$. □

Example 4.15. For our CAFs $\mathcal{F}$ and $\mathcal{G}$ from Example 4.1 we see that —given $\rho = \textit{cl-stb}_{cf}$— the kernels are isomorphic. Hence $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent up to renaming.

5 Computational Complexity

In this section we examine the computational complexity of deciding equivalence between two CAFs $\mathcal{F}$ and $\mathcal{G}$ for every equivalence notion which has been established in this paper. We assume the reader to be familiar with the polynomial hierarchy. Moreover, by $\mathrm{QSAT}^{\exists}_{n}$ ($\mathrm{QSAT}^{\forall}_{n}$) we denote the generic Σ^{P}_{n}-complete (Π^{P}_{n}-complete) problem, i.e. checking validity of a corresponding QBF. Our results reveal that ordinary equivalence can be computationally hard, up to the third level of the polynomial hierarchy for both variants of semi-stable and stage semantics as well as for i-preferred semantics. For the remaining semantics under consideration, the problem is Π^{P}_{2}-complete; the only exception is i-grounded semantics for which deciding ordinary equivalence is P-complete. Moreover, we show that deciding strong equivalence up to renaming extends the list of problems which lie in NP but are not known to be NP-complete.

First we present our complexity results for ordinary equivalence. We formulate the following decision problem:

VER-OE$_{\rho}$
Input: Two CAFs $\mathcal{F}$, $\mathcal{G}$.
Output: TRUE iff $\mathcal{F}$, $\mathcal{G}$ are ordinary equivalent w.r.t. ρ.

We obtain the following computational complexity results for deciding ordinary equivalence:

Theorem 5.1. VER-OE$_{\rho}$ *is*

- P*-complete for* $\rho = gr_c$*;*
- Π^{P}_{2}*-complete for* $\rho \in \{cf_c, ad_c, co_c, na_c, \textit{cl-pr}, \textit{cl-na}, stb_c, \textit{cl-stb}_{cf}, \textit{cl-stb}_{ad}, \}$*; and*
- Π^{P}_{3}*-complete for* $\rho \in \{pr_c, ss_c, stg_c, \textit{cl-stg}, \textit{cl-ss}\}$.

In the following we will provide proofs for the results from Theorem 5.1. To begin with, we show that verifying ordinary equivalence for i-grounded semantics is P-complete.

Proposition 5.2. *Deciding* $\textsc{Ver-OE}_{gr_c}$ *is* P*-complete.*

Proof. $\textsc{Ver-OE}_{gr_c}$ is in P since computing the grounded extensions of F and G and comparing the claims can be done in polynomial time. Hardness is by a reduction from the verification problem $Ver_{gr_c}^{CAF}$ for i-grounded semantics (which is P-complete by (Dvorák and Woltran 2020)) by setting $\mathcal{F} = \mathcal{F}$ and $\mathcal{G} = (S, \emptyset, id)$ for an instance $(\mathcal{F}, S)$ of $Ver_{gr_c}^{CAF}$. We obtain $gr_c(\mathcal{F}) = gr_c(\mathcal{G})$ iff $S = gr_c(\mathcal{F})$. □

Membership proofs for $\textsc{Ver-OE}_\rho$, $\rho \neq gr_c$ are by standard guess-and-check procedures for the complementary problems: Guess a set of claims S and check whether it holds that $S \in \mathcal{F}$ as well as $S \notin \mathcal{G}$. For the semantics $\rho \in \{cf_c, ad_c, co_c, na_c, stb_c, cl\text{-}stb_{cf}, cl\text{-}stb_{ad}\}$, the latter requires two NP-oracle calls; for $\rho \in \{cl\text{-}pr, cl\text{-}na\}$ we require four NP-oracle calls (recall that verification for cl-preferred and cl-naive semantics is in $\mathsf{D}_1^{\mathsf{P}}$ (Dvořák et al. 2021)), which shows that $\textsc{Ver-OE}_\rho$ is in Π_2^{P}. For the semantics $\rho \in \{pr_c, ss_c, stg_c, cl\text{-}ss, cl\text{-}stg\}$, we require two Σ_2^{P}-oracle calls to check $S \in \mathcal{F}$ and $S \notin \mathcal{G}$; yielding Π_3^{P}-procedures for the decision problem $\textsc{Ver-OE}_\rho$.

To show hardness of $\textsc{Ver-OE}_\rho$ for $\rho \neq gr_c$, we present reductions from $\text{QSAT}_2^\forall$ or $\text{QSAT}_2^\exists$, respectively. The overall idea is to construct two CAFs $\mathcal{F}$, $\mathcal{G}$ where $\rho(\mathcal{F})$ depends on the particular instance of the source problem while $\mathcal{G}$ serves as controlling entity. For a given instance $\Psi = Q_1 X_1 \dots Q_n X_n \varphi$ of $\text{QSAT}_2^\forall$ or $\text{QSAT}_2^\exists$, respectively, we design the CAF $\mathcal{F}$ in a way such that $\rho(\mathcal{F})$ depends on the models of φ while $\mathcal{G}$ possesses every possible ρ-claim-set which can be obtained in $\mathcal{F}$ by varying φ, i.e., $\rho(\mathcal{G})$ is independent of the validity of Ψ. $\mathcal{F}$ is then constructed such that Ψ is valid iff $\rho(\mathcal{F}) = \rho(\mathcal{G})$ (if we reduce $\text{QSAT}_2^\forall$) or $\rho(\mathcal{F}) \neq \rho(\mathcal{G})$ (in case we reduce $\text{QSAT}_2^\exists$).

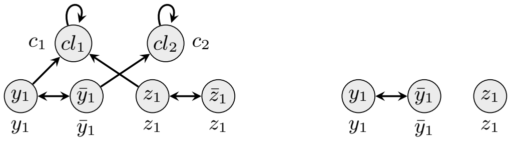
We will first discuss the hardness proofs for those semantics for which $\textsc{Ver-OE}_\rho$ is Π_2^{P}-complete. We outline the underlying aforementioned idea for i-stable semantics.

Proposition 5.3. *Deciding* $\textsc{Ver-OE}_\rho$ *is* Π_2^{P}*-hard for* $\rho \in \{stb_c, cl\text{-}stb_{cf}, cl\text{-}stb_{ad}\}$.

Proof. Let $\Psi = \forall Y \exists Z \varphi(Y, Z)$ be an instance of $\text{QSAT}_2^\forall$ where φ is identified with a set of clauses C over atoms in $V = Y \cup Z$. We construct two CAFs $\mathcal{F} = (A_\mathcal{F}, R_\mathcal{F}, cl_\mathcal{F})$ and $\mathcal{G} = (A_\mathcal{G}, R_\mathcal{G}, id)$. The CAF $\mathcal{F}$ is given by

$$\begin{aligned} A_\mathcal{F} &= V \cup \bar{V} \cup C \text{ with } \bar{V} = \{\bar{v} \mid v \in V\}; \\ R_\mathcal{F} &= \{(v, cl) \mid cl \in C, v \in cl\} \cup \{(cl, cl) \mid cl \in C\} \cup \\ &\quad \{(\bar{v}, cl) \mid cl \in C, \neg v \in cl\} \cup \{(v, \bar{v}), (\bar{v}, v) \mid v \in V\} \end{aligned}$$

and $cl_\mathcal{F}(z) = cl_\mathcal{F}(\bar{z}) = z$ for $z \in Z$ and $cl_\mathcal{F}(a) = a$ else; that is, we introduce arguments for every clause and every literal; a literal argument attacks a clause argument if the corresponding literal is contained in the respective clause; moreover, the clauses are self-attacking and every literal and its negation attack each other. We assign every atom $z \in Z$ the same claim as its negation $\bar{z}$; the remaining arguments have their unique argument name as claim. The CAF $\mathcal{G}$ is given by $A_\mathcal{G} = V \cup \bar{Y}$; $R_\mathcal{G} = \{(y, \bar{y}), (\bar{y}, y) \mid y \in Y\}$. An example of the reduction is given in Figure 1. Observe that the i-stable (cl-stable) claim-sets of $\mathcal{G}$ are given by sets of the form $Y' \cup \{\bar{y} \mid y \notin Y'\} \cup Z$ for $Y' \subseteq Y$.



(a) CAF $\mathcal{F}$ (b) CAF $\mathcal{G}$

Figure 1: Reduction from the proof of Proposition 5.3 for a formula $\forall Y \exists Z \varphi(Y, Z)$ where φ is given by the clauses $\{\{y_1, z_1\}, \{\bar{y}_2\}\}$.

It can be shown that $Y' \cup \{\bar{y} \mid y \notin Y'\} \cup Z$ is i-stable (cl-stable) in $\mathcal{F}$ for every $Y' \subseteq Y$ iff Ψ is valid. By design of $\mathcal{G}$, the latter is satisfied iff the i-stable (cl-stable) extensions of $\mathcal{F}$ and $\mathcal{G}$ coincide. That is, Ψ is valid iff $\rho(\mathcal{F}) = \rho(\mathcal{G})$ for $\rho \in \{stb_c, cl\text{-}stb_{cf}, cl\text{-}stb_{ad}\}$. □

By modifying the constructions from the proof of Proposition 5.3 we obtain Π_2^{P}-hardness of $\textsc{Ver-OE}_{na_c}$. For the construction of $\mathcal{F}$ in the Π_2^{P}-hardness proof of $\textsc{Ver-OE}_\rho$, $\rho = \{cf_c, ad_c, cl\text{-}na, cl\text{-}pr\}$, we choose a slightly different approach: For an instance $\Psi = \forall Y \exists Z \varphi(Y, Z)$ of $\text{QSAT}_2^\forall$, we construct $\mathcal{F}$ such that each literal in a clause cl is represented by an argument having claim cl; we furthermore introduce arguments for each atom $y \in Y$ and its negation; finally, every two arguments representing negated literals attack each other. We construct $\mathcal{G}$ in a way such that $\rho(\mathcal{G})$ contains precisely the claim-sets $Y' \cup \{\bar{y} \mid y \notin Y'\} \cup C$. Similar as above, it can be shown that Ψ is valid iff $\rho(\mathcal{F}) = \rho(\mathcal{G})$. An appropriate adaptation and claim-assignment of the standard construction as presented in (Dvorák and Dunne 2018, Reduction 3.6) yields Π_2^{P}-hardness for i-complete semantics.

Turning now to the Π_3^{P}-hardness results, we adjust our general reduction scheme by targeting inequality of $\rho(\mathcal{F})$ and $\rho(\mathcal{G})$ in case the given instance Ψ of $\text{QSAT}_3^\exists$ is valid. As an example, we present the construction from the Π_3^{P}-hardness proof for cl-semi-stable and cl-stage semantics.

Proposition 5.4. *Deciding* $\textsc{Ver-OE}_\rho$ *is* Π_3^{P}*-hard for* $\rho \in \{cl\text{-}ss, cl\text{-}stg\}$.

Proof. Consider an instance $\Psi = \exists X \forall Y \exists Z \varphi(X, Y, Z)$ of $\text{QSAT}_3^\exists$, where φ is given by a set of clauses C over atoms in $V = X \cup Y \cup Z$. We can assume that there is $y_0 \in Y$ with $y_0 \in cl$ for all $cl \in C$ (otherwise we can add such a y_0 without changing the validity of Ψ). We write $\bar{v}$ to denote $\neg v$ for an atom $v \in V$, moreover, let $V' = X \cup Y$. We construct CAFs $\mathcal{F} = (A_\mathcal{F}, R_\mathcal{F}, cl_\mathcal{F})$ and $\mathcal{G} = (A_\mathcal{G}, R_\mathcal{G}, id)$ as follows: The CAF $\mathcal{F}$ is given by

$$\begin{aligned} A_\mathcal{F} &= V \cup \bar{V} \cup \mathcal{C} \cup \{\varphi_1, \varphi_2\} \cup \{d_v, d_{\bar{v}} \mid v \in V' \cup \bar{V}'\}; \\ R_\mathcal{F} &= \{(a, cl) \mid cl \in C, a \in cl, a \in V \cup \bar{V}\} \cup \{(cl, \varphi) \mid cl \in C\} \cup \\ &\quad \{(a, d_a), (d_a, d_a) \mid a \in V' \cup \bar{V}'\} \cup \{(\varphi_1, \varphi_2), (\varphi_2, \varphi_2)\} \\ &\quad \cup \{(v, \bar{v}), (\bar{v}, v) \mid v \in V\}; \end{aligned}$$

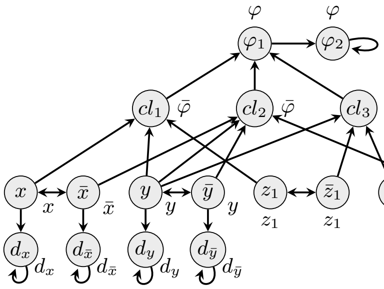


Figure 2: Construction of the CAF $\mathcal{F}$ from the proof from Proposition 5.4 for the formula $\exists X \forall Y \exists Z \varphi(X, Y, Z)$ with clauses $\{\{z_1, x, y\}, \{\neg x, \neg y, \neg z_2, y\}, \{\neg z_1, z_2, y\}\}$.

$cl_{\mathcal{F}}(v) = cl_{\mathcal{F}}(\bar{v}) = v$ for $v \in Y \cup Z$; $cl_{\mathcal{F}}(cl) = \bar{\varphi}$ for $cl \in C$; $cl_{\mathcal{F}}(\varphi_1) = cl_{\mathcal{F}}(\varphi_2) = \varphi$; and $cl_{\mathcal{F}}(a) = a$ otherwise. An example of this construction is given in Figure 2. We observe that each set $X' \cup \{\bar{x} \mid x \notin X'\} \cup Y \cup Z \cup \{\varphi\}$ is cl-semi-stable (cl-stage) in $\mathcal{F}$ for every $X' \subseteq X$ (remember that there is $y_0 \in Y$ which attacks every clause $cl \in C$).

We define $\mathcal{G} = (A_{\mathcal{G}}, R_{\mathcal{G}}, id)$ such that it has the cl-semi-stable (cl-stage) claim-sets $X' \cup \{\bar{x} \mid x \notin X'\} \cup Y \cup Z \cup \{e\}$ for every $X' \subseteq X$, $e \in \{\varphi, \bar{\varphi}\}$ with $A_{\mathcal{G}} = V \cup \bar{X} \cup \{\varphi, \bar{\varphi}\}$, and $R_{\mathcal{G}} = \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \{(\varphi, \bar{\varphi}), (\bar{\varphi}, \varphi)\}$. It is easy to see that $\mathcal{G}$ possesses exactly the desired cl-semi-stable (cl-stage) claim-sets.

It can be checked that $X' \cup \{\bar{x} \mid x \notin X'\} \cup Y \cup Z \cup \{\bar{\varphi}\}$ is cl-semi-stable (cl-stage) in $\mathcal{F}$ for every $X' \subseteq X$ iff Ψ is not valid. The former is satisfied iff $\mathcal{F}$ and $\mathcal{G}$ possess the cl-semi-stable (cl-stage) claim-sets. Thus Ψ is valid iff $\rho(\mathcal{F}) \neq \rho(\mathcal{G})$ for $\rho \in \{cl\text{-}ss, cl\text{-}stg\}$. □

Π_3^{P}-hardness of ordinary equivalence for i-semi-stable and i-stage semantics is by adapting the Π_3^{P}-hardness proof of the concurrence problem for semi-stable semantics, i.e., deciding whether $ss_c(\mathcal{F}) = cl\text{-}ss(\mathcal{F})$ for a CAF $\mathcal{F}$ (Dvořák et al. 2021, Proposition 6). For i-preferred semantics, we modify the standard reduction for preferred semantics (cf. (Dvorák and Dunne 2018, Reduction 3.7)) via an appropriate claim-assignment. This concludes the proof of Theorem 5.1.

Remark 5.5. The computational complexity results from Theorem 5.1 extend to ordinary equivalence up to renaming by Proposition 4.6 for any semantics under consideration.

Having established complexity results for ordinary equivalence it remains to discuss the computational complexity of strong equivalence and strong equivalence up to renaming.

VER-SE$_\rho$
Input: Two CAFs $\mathcal{F}$, $\mathcal{G}$.
Output: TRUE iff $\mathcal{F}$, $\mathcal{G}$ are strongly equivalent w.r.t. ρ.

Recall that in Section 3, we have shown that strong equivalence of two CAFs $\mathcal{F}$ and $\mathcal{G}$ can be characterized via syntactic equivalence of their kernels. Since the computation and comparison of the kernels of $\mathcal{F}$ and $\mathcal{G}$ can be done in polynomial time, we obtain tractability of strong equivalence for every semantics under consideration.

Theorem 5.6. *The problem* VER-SE$_\rho$ *can be solved in polynomial time for any semantics* ρ *considered in this paper.*

Finally, we consider strong equivalence up to renaming. An analogous decision problem be formulated as follows:

VER-SER$_\rho$
Input: Two CAFs $\mathcal{F}$, $\mathcal{G}$.
Output: TRUE iff $\mathcal{F}$, $\mathcal{G}$ are strongly equivalent up to renaming w.r.t. ρ.

As outlined above, the computation of the kernels lies in P and is therefore negligible; the complexity of verifying strong equivalence up to renaming thus stems entirely from deciding whether two labelled graphs (i.e., the kernels of the given CAFs) are isomorphic. As a consequence we obtain that the complexity of VER-SER$_\rho$ coincides with the complexity of the famous graph isomorphism problem.

Theorem 5.7. *The problem* VER-SER$_\rho$ *is exactly as hard as the graph isomorphism problem for any semantics* ρ *considered in this paper.*

It is well-known that the graph isomorphism problem lies in NP but is not known to be NP-complete (although the latter is considered unlikely (Schöning 1988)).

6 Conclusion and Future Work

In this paper, we considered ordinary and strong equivalence as well as novel equivalence notions based on argument renaming for CAFs w.r.t. all semantics for CAFs which have been considered in the literature so far and provided a complexity analysis of all considered equivalence notions.

Our characterization results for strong equivalence are in line with existing studies for related argumentation formalisms (Oikarinen and Woltran 2011; Dvorák, Rapberger, and Woltran 2020b); in addition, we adapt an argument-independent view by considering equivalence under renaming. Equivalence of logic-based argumentation has been studied in (Amgoud, Besnard, and Vesic 2014); they show that under certain conditions on the underlying logic, unnecessary arguments can be removed while retaining (strong) equivalence. In contrast to their work, our studies are independent of the underlying formalism of the instantiated argumentation system as we do not impose any further constraints on the arguments or their claims; in this way, it is even possible to test equivalence between argumentation systems stemming from entirely different base formalisms.

For future work, we want to extend our strong equivalence studies by considering certain constraints of the framework modifications. What has been commonly investigated in the literature are *normal expansions* where attacks can only be introduced if they involve newly added arguments (observe that in the proof of Theorem 3.17, the expansion in case (a) satisfy this criteria while $\mathcal{H}$ in case (b) introduces also new attacks between existing arguments). We moreover want to adapt our strong equivalence notion to arbitrary CAFs, not only compatible ones, by relaxing the notion of framework expansions. Another point on our agenda is to consider certain sub-classes of CAFs, which have been introduced in the literature, e.g., well-formed CAFs which impose restrictions on the attack relation.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig and by the Austrian Science Fund (FWF) through projects W1255-N23 and Y698.

References

Amgoud, L.; Besnard, P.; and Vesic, S. 2014. Equivalence in logic-based argumentation. *J. Appl. Non Class. Logics* 24(3):181–208.

Baroni, P.; Cerutti, F.; Giacomin, M.; and Guida, G. 2011. Afra: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning* 52(1):19–37.

Baumann, R., and Strass, H. 2016. An abstract logical approach to characterizing strong equivalence in logic-based knowledge representation formalisms. *KR* 16:525–528.

Baumann, R.; Linsbichler, T.; and Woltran, S. 2016. Verifiability of argumentation semantics. In *Proceedings of the 6th International Conference of Computational Models of Argument, COMMA*, 83–94.

Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artif. Intell.* 128(1-2):203–235.

Bondarenko, A.; Toni, F.; and Kowalski, R. A. 1993. An assumption-based framework for non-monotonic reasoning. In Pereira, L. M., and Nerode, A., eds., *Logic Programming and Non-monotonic Reasoning, Proceedings of the Second International Workshop, Lisbon, Portugal, June 1993*, 171–189. MIT Press.

Caminada, M.; Sá, S.; Alcântara, J.; and Dvořák, W. 2015a. On the difference between assumption-based argumentation and abstract argumentation. *IfCoLog Journal of Logic and its Applications* 2(1):15–34.

Caminada, M.; Sá, S.; Alcântara, J. F. L.; and Dvorák, W. 2015b. On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reason.* 58:87–111.

Caminada, M. 2006. Semi-stable semantics. In Dunne, P. E., and Bench-Capon, T. J. M., eds., *Computational Models of Argument: Proceedings of COMMA 2006, September 11-12, 2006, Liverpool, UK*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, 121–130. IOS Press.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 378–389. Springer.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–357.

Dvorák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications.

Dvorák, W., and Woltran, S. 2020. Complexity of abstract argumentation under a claim-centric view. *Artif. Intell.* 285:103290.

Dvorák, W.; Rapberger, A.; and Woltran, S. 2020a. Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to setafs. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, 341–350.

Dvorák, W.; Rapberger, A.; and Woltran, S. 2020b. On the different types of collective attacks in abstract argumentation: equivalence results for setafs. *J. Log. Comput.* 30(5):1063–1107.

Dvořák, W.; Greßler, A.; Rapberger, A.; and Woltran, S. 2021. The complexity landscape of claim-augmented argumentation frameworks. In *Proc. AAAI*. To appear - available at https://www.dbai.tuwien.ac.at/research/report/dbai-tr-2021-121.pdf.

Gorogiannis, N., and Hunter, A. 2011. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artif. Intell.* 175(9-10):1479–1497.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)* 2(4):526–541.

Nielsen, S. H., and Parsons, S. 2006. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *International Workshop on Argumentation in Multi-Agent Systems*, 54–73. Springer.

Oikarinen, E., and Woltran, S. 2011. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.* 175(14-15):1985–2009.

Rapberger, A. 2020. Defining argumentation semantics under a claim-centric view. In Rudolph, S., and Marreiros, G., eds., *Proceedings of the 9th European Starting AI Researchers' Symposium 2020 co-located with 24th European Conference on Artificial Intelligence (ECAI 2020), Santiago Compostela, Spain, August, 2020*, volume 2655 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Schöning, U. 1988. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences* 37(3):312–323.

Thimm, M. 2012. A probabilistic semantics for abstract argumentation. In *ECAI*, volume 12, 750–755.

Verheij, B. 1996. Two approaches to dialectical argumentation: admissible sets and argumentation stages. *Proc. NAIC* 96:357–368.

Towards a Temporal Account of Contrary-to-Duty Constraints over Complex Actions in the Situation Calculus

Jens Claßen , **James P. Delgrande**
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
jens_classen@sfu.ca, jim@cs.sfu.ca

Abstract

With the advent of artificial agents in everyday life, it is important that these agents are guided by social norms and moral guidelines. Notions of obligation, permission, and the like have traditionally been studied in the field of Deontic Logic, where deontic assertions generally refer to what an agent should or should not do; that is they refer to *actions*. In Artificial Intelligence, the Situation Calculus is (arguably) the best known and most studied formalism for reasoning about action and change. In this paper, we further investigate the integration of these two areas, particularly addressing so-called contrary-to-duty (CTD) scenarios. For this purpose, we present a new logic based on Lakemeyer and Levesque's modal Situation Calculus variant $\mathcal{ES}$ that we modify to express properties about programs from the action language GOLOG, extended by new constructs for negated programs and their joint execution. We use this formalism to discuss three different approaches to CTD scenarios. First, we show it to be expressive enough to fully capture Meyer's dynamic deontic logic $\mathrm{PD_eL}$, and hence corresponding solutions for CTDs. Second, we demonstrate how our previous approach to tackle CTDs in terms of defeasible conditionals over a restricted set of GOLOG programs can be represented as well, along with a method to compile them directly into the Situation Calculus action theory. Finally, we extend the language of these conditionals to include a simple notion of intention, which allows to describe CTDs not only in terms of actions that will follow immediately, but that the agent has committed to execute at some time in the foreseeable future. All in all, the contribution of the paper is thus an approach that is substantially more general than previous approaches, and is able to handle CTDs in a flexible manner.

1 Introduction

With artificial agents playing an ever-greater role in our daily lives, there has been increasing interest in researching ways to ensure that such agents act ethically and subject their actions to social norms, in particular where they interact with humans or operate in shared environments. One possible approach is to formalize relevant notions such as obligation, permission and prohibition in a logical language, which traditionally has been the subject of study in the field of Deontic Logic (von Wright 1951; Gabbay et al. 2013).

Probably the best researched system of deontic logic is *Standard Deontic Logic* (SDL), a variant of the modal logic KD (Chellas 1980), where a modal operator $\mathbf{O}\phi$ expresses that "ϕ is obligatory" or "it ought to be that ϕ", permission is defined as its dual ($\mathbf{P}\phi = \neg\mathbf{O}\neg\phi$), and prohibition as the negation of permission ($\mathbf{F}\phi = \neg\mathbf{P}\phi$). Semantically, accessible worlds correspond to worlds that are in a certain sense *ideal*, and obligatory/permitted/forbidden is whatever is true in all/some/no accessible worlds.

While simple and elegant, SDL is also somewhat weak, and yields some unintuitive consequences, which have been traditionally referred to as "paradoxes" in the literature. One particular class of such paradoxes is concerned with so-called *contrary-to-duty* (CTD) obligations, usually given in the form of conditional exhortations that state what ought to be (done) if a certain other obligation is neglected. A well-known example scenario is due to Chisholm (1963), and can be phrased as follows:

1. You ought to help your neighbour.
2. If you help your neighbour you should tell them.
3. If you don't help your neighbour, you shouldn't tell them.
4. You don't help your neighbour.

Intuitively, these statements are consistent, independent from another, and lead to the conclusion that one shouldn't tell the neighbour one will come to help. However, different possible encodings in SDL all either lead to an inconsistency, or that one of the statements can be derived from the others. It was later recognized (Hansson 1969) that the problem lies in representing these statements through monadic deontic modalities and material implications, and that it rather requires *dyadic* obligations such as $\mathbf{O}(\mathit{tell}/\mathit{help})$ to express systems of *defeasible* conditionals. Semantically, the latter do not merely distinguish ideal from non-ideal worlds, but rank possible worlds according to some preference relation, allowing for differing "degrees of ideality". For example, worlds in which we don't go to help the neighbour but tell them we are coming are ranked worse than those where we don't go, but at least don't tell them we intend to come, even though both cases are not ideal.

Another observation about the Chisholm scenario is that there is a temporal aspect to it: If we are *going to* help, then we ought to tell them *beforehand*. Furthermore, here deontic modalities apply to actions ("ought-to-do") rather than propositions ("ought-to-be"). While some authors simply used propositions to represent actions, in his seminal article, von Wright (1951) originally introduced deontic modalities as applying to action types. He argued that a suitable deontic logic needs to be built upon the foundation of a more general theory of action (von Wright 1963). Essentially, when reasoning about obligations and permissions applying to actions, we have to take into consideration that actions have preconditions and effects that result in various forms of interaction and interdependency between them, and so it makes sense to formalize these notions. For example, helping the neighbour may require having the necessary supplies to do so, which may necessitate other actions, such as buying supplies at the hardware store.

Deontic action logic is an active area of research, and notable approaches to use such formalisms for tackling CTDs include (Bartha 1999), which uses *stit* ("see to it that") semantics (Horty 2001), and (Meyer, Dignum, and Wieringa 1994), which is based on Meyer's (1988) deontic dynamic logic $\mathrm{PD_eL}$. While (Bartha 1999) extends the aforementioned idea to assign degrees of ideality to possible histories (rather than worlds), a problem with stit is that actions do not have proper names or types, but are described purely through their effects, making it difficult to deal with deontic constraints over complex actions. This is not an issue in dynamic logic, but the approach to CTDs suggested in (Meyer, Dignum, and Wieringa 1994) is somewhat rudimentary in that rankings among alternatives are not inferred "automatically" by means of some non-monotonic mechanism, but need to be encoded "manually" by the domain designer.

In a recent paper (Claßen and Delgrande 2020), we proposed to tackle CTDs over actions by integrating deontic notions into what is (arguably) the best known and most studied formalism for reasoning about action and change, namely the Situation Calculus (McCarthy and Hayes 1969; Reiter 2001), together with the agent programming language GOLOG (Levesque et al. 1997) that is defined on top of it. Among other things, we proposed to express dyadic obligations as defeasible conditionals over complex actions (i.e., programs of GOLOG), and understand them as *deontic constraints* that the agent has to consider when planning its actions. These conditionals would then again induce a ranking of differing "degrees of ideality", but over situations (i.e., action sequences) instead of possible worlds. Moreover, we showed that these constraints can then be "compiled away" into the action theory, so that after a preprocessing step, no additional reasoning machinery is required for planning under such deontic constraints. A limitation was that for conditionals we considered a very restricted fragment of GOLOG programs that only admit single actions, one of the reasons being that the approach requires a notion of *negated* actions and programs, e.g. to express "not helping the neighbour", which is not trivial in the general case. Another limiting assumption we made is that the action the agent is "going to do" (e.g., helping) will follow immediately after the one it is currently deliberating about (e.g., telling).

In this paper, we address some of these issues and explore a more unified view on contrary-to-duty constraints over actions. For this purpose, in Section 2, we propose a new logic called $\mathcal{ESGL}$ that is based on an extension (Claßen and Lakemeyer 2008) of Lakemeyer and Levesque's (2010) modal Situation Calculus variant $\mathcal{ES}$, which we modify to express properties about a fragment of GOLOG programs, now including a more sophisticated notion of action negation as proposed by Meyer (1988). While the classical Situation Calculus is defined axiomatically over Tarskian structures, the modal variant we employ here uses a special semantics that renders many formal definitions and proofs easier, while retaining all benefits such as Reiter's (1991) solution to the frame problem. In particular, this is helpful for defining the new negation operator, where we shift from a macro-based definition of GOLOG (Levesque et al. 1997) to a transition-based semantics (De Giacomo, Lespérance, and Levesque 2000). Moreover, different from previous definitions, our semantics uses linear-time traces rather than branching-time tree models, which further simplifies the treatment. We use the new formalism to discuss three different approaches to CTDs. First, in Section 3 we show it to be expressive enough to fully capture Meyer's dynamic deontic logic $\mathrm{PD_eL}$, and hence corresponding solutions for CTDs. Second, in Section 4 we demonstrate how our previous approach to tackle CTDs in terms of defeasible conditionals over a restricted set of GOLOG programs can equally be represented. Finally, in Section 5 we extend the language of conditionals to include a simple notion of intention, which allows to describe CTDs not only in terms of actions that will follow immediately, but that the agent has committed to execute at some time in the foreseeable future. The overall contribution of this paper is hence an approach that is substantially more general than previous works, allowing for a flexible modelling of, among other things, CTDs in the style of Chisholm's paradox.

2 The Logic $\mathcal{ESGL}$

In this section we present the formal definition of the logic $\mathcal{ESGL}$. It is based on Lakemeyer and Levesque's (2010) logic $\mathcal{ES}$, a modal variant of the (epistemic) situation calculus, where instead of situation terms, modal operators $[t]\phi$ ("ϕ is true after action t") and $\Box\phi$ ("ϕ holds after any sequence of actions") are used to talk about future states of affairs. Our new logic is a variant of Claßen and Lakemeyer's (2008; 2013) extension $\mathcal{ESG}$, which, among other things, extends the $[\cdot]$ operator to take a program (or complex action) δ as argument, where δ is from a subset of the agent programming language GOLOG (Levesque et al. 1997). The latter includes both deterministic programming constructs such as while loops and if conditionals, and non-deterministic ones such as non-deterministic branching and iteration.

While the main purpose of $\mathcal{ESG}$ was the verification of GOLOG programs, our focus of interest in this paper is representing and reasoning about deontic properties. The new logic $\mathcal{ESGL}$ we propose differs from $\mathcal{ESG}$ in two aspects: For one, we extend the set of GOLOG programming constructs by negation ($\overline{\delta}$) and joint execution ($\delta_1 \times \delta_2$) of programs, which will allow to express deontic constraints as conditionals over GOLOG programs. For another, instead of interpreting formulas over branching-time, tree-shaped models as is done in the situation calculus and $\mathcal{ES}$, we will use linear-time models called traces. The latter not only is somewhat simpler, but, as we will see, helps in interpreting the new constructs in a similar fashion as in Meyer's (1988) dynamic deontic logic, thus inheriting many of its desirable features. Note though that this section solely deals with interpretating programs and their properties, and that deontic notions will only be introduced and discussed in the subsequent sections.

2.1 Syntax

The language is a first-order modal dialect with equality and sorts of type *object*, *action* and *number*. It includes countably infinitely many standard names for each of the sorts, denoted by $\mathcal{N}_O$, $\mathcal{N}_A$, and $\mathcal{N}_N$ respectively, allowing for a substitutional interpretation of quantification. Also included are both fluent and rigid predicate and function symbols. Fluents vary as the result of actions, but rigids do not. The logical connectives are $\wedge$, $\neg$, $\forall$, together with the modal operator $\langle\delta\rangle$, where δ may be any program expression, as defined below. Other connectives like $\vee$, $\supset$, $\subset$, $\equiv$, and $\exists$ are used as the usual abbreviations, and terms and formulas are built from these primitives in the usual way, including basic arithmetic operations and relations for numbers.

We read $\langle\delta\rangle\phi$ as "ϕ holds after some execution of program δ" and define its dual $[\delta]\phi$ as abbreviation for $\neg\langle\delta\rangle\neg\phi$, where $\Box\phi$ (read: "ϕ holds after any sequence of actions") in turn stands for $[\top]\phi$. The set of *programs* Δ is given by:

$$\delta ::= \ t \ \mid \ \phi? \ \mid \ \delta_1;\delta_2 \ \mid \ \overline{\delta} \ \mid \ \delta_1+\delta_2 \ \mid \ \pi x.\delta \ \mid \ \delta^* \qquad (1)$$

in which t can be any action term (including a variable), ϕ a formula, and x a variable. We thus consider a set of programs given by primitive actions t, test conditions $\phi?$, sequence $\delta_1;\delta_2$, action negation $\overline{\delta}$, nondeterministic branching $\delta_1+\delta_2$, nondeterministic choice of argument ("pick") $\pi x.\delta$, and nondeterministic iteration δ^*. In addition, we define joint execution $\delta_1 \times \delta_2$ as abbreviation for $\overline{\overline{\delta_1}+\overline{\delta_2}}$, the empty program *nil* as TRUE?, the universal action $\top$ as $\pi a.\ a^*$, and failure $\bot$ as FALSE?. For any expression (formula, term, program,...) β, we use β^x_t to denote the result of simultaneously replacing all free occurrences of variable x by term t. We call a formula without $\Box$ and $[\cdot]$ a *fluent formula*, and one without free variables a *sentence*.

2.2 Semantics

Intuitively, a *trace* τ will be used to determine, at any point in time $k \in \mathbb{N}$, (a) what values the (fluent and rigid) predicates and functions take and (b) what action will be executed next. For the latter, we simply assume that there is a distinguished functional fluent $\aleph$ of sort action with this special meaning, but that we otherwise treat like any other function symbol. More precisely, let

- $\mathcal{N}$ denote the set of all standard names,
- $\mathcal{P}_F$ the set of all primitive sentences $R(n_1,\ldots,n_m)$, where R is a (fluent or rigid) predicate symbol and all the n_i are standard names, and
- $\mathcal{P}_T$ the set of all primitive terms $g(n_1,\ldots,n_m)$, where g is a (fluent or rigid) function symbol and all the n_i are standard names.

Then a trace $\tau \in \mathcal{T}$ is any mapping

$$\tau : \mathbb{N}\times\mathcal{P}_F \to \{0,1\} \qquad\qquad \tau : \mathbb{N}\times\mathcal{P}_T \to \mathcal{N}$$

that preserves sorts, interprets arithmetic operations and relations in the usual way, and satisfies the rigidity constraint: if g is a rigid function or predicate symbol, then for all k and k', $\tau[k, g(n_1,\ldots,n_k)] = \tau[k', g(n_1,\ldots,n_k)]$. The *progression* of a trace τ by k time points is the trace $\tau^{(k)}$ where for all $l \in \mathbb{N}$ and all $\beta \in \mathcal{P}_F \cup \mathcal{P}_T$,

$$\tau^{(k)}[l,\beta] = \tau[k+l,\beta].$$

We extend the idea of co-referring standard names to arbitrary ground terms as follows. Given a variable-free term t and a trace τ, we define $|t|_\tau$ (read: the co-referring standard name for t given τ) by:

1. If $t \in \mathcal{N}$, then $|t|_\tau = t$;
2. $|h(t_1,\ldots,t_k)|_\tau = \tau[0, h(n_1,\ldots,n_k)]$, if $n_i = |t_i|_\tau$.

Truth of a sentence ϕ wrt. a trace τ is then given by:

1. $\tau \models F(t_1,\ldots,t_k)$ iff $\tau[0, F(|t_1|_\tau,\ldots,|t_k|_\tau)] = 1$;
2. $\tau \models (t_1 = t_2)$ iff $|t_1|_\tau$ and $|t_2|_\tau$ are identical;
3. $\tau \models \phi\wedge\psi$ iff $\tau \models \phi$ and $\tau \models \psi$;
4. $\tau \models \neg\phi$ iff $\tau \not\models \phi$;
5. $\tau \models \forall x.\phi$ iff $\tau \models \phi^x_n$ for all $n \in \mathcal{N}_x$;
6. $\tau \models \langle\delta\rangle\phi$ iff $\tau \in \|\delta;\phi?\|$.

Above, $\mathcal{N}_x$ refers to the set of standard names of the same sort as x. A sentence is *satisfiable* if some τ exists with $\tau \models \phi$. When Σ is a set of sentences and ϕ a sentence, we write $\Sigma \models \phi$ (read: "Σ logically entails ϕ") to mean that for every τ, if $\tau \models \beta$ for every $\beta \in \Sigma$, then also $\tau \models \phi$. Finally, we write $\models \phi$ (read: "ϕ is valid") to mean $\{\} \models \phi$.

The interpretation of programs as required in rule 6 is defined by mutual induction. Let a *configuration* $\mathsf{c} = \langle\tau,\delta\rangle$ consist of a trace $\tau \in \mathcal{T}$ (intuitively describing the current and future states of the world) and a program $\delta \in \Delta$ (intuitively what remains to be executed). Then the *final configurations* $\mathcal{F}$ are the least set given by the rules shown in Fig. 2, and for every action name n, the transition relation $\xrightarrow{n}$ among configurations is the least set satisfying the rules shown in Fig. 1. For arbitrary action sequences z, we define the reflexive and transitive closure of $\xrightarrow{n}$ inductively as:

- $\mathsf{c} \xrightarrow{\langle\rangle} \mathsf{c}'$ iff $\mathsf{c} = \mathsf{c}'$;
- $\mathsf{c} \xrightarrow{nz} \mathsf{c}'$ iff there is some c'' such that $\mathsf{c} \xrightarrow{n} \mathsf{c}''$ and $\mathsf{c}'' \xrightarrow{z} \mathsf{c}'$.

The *traces admitted by program* δ are then given by

$$\|\delta\| \ \doteq \ \{\tau \mid \langle\tau,\delta\rangle \xrightarrow{z} \langle\tau',\delta'\rangle,\ \langle\tau',\delta'\rangle \in \mathcal{F}\} \qquad (2)$$

The interpretation of formulas is standard in the sense that atomic formulas that are not in the scope of some $[\cdot]$ or $\langle\cdot\rangle$ operator are evaluated at the first time point of the trace (rule 1), and the Boolean connectives are defined as usual. For quantification (rule 5), we follow the substitutional interpretation of (Lakemeyer and Levesque 2010) in which a formula $\forall x P(x)$ holds just in case $P(x)$ is true for every instantiation of x by a standard name of the same sort.

Probably the most noteworthy difference to previous logics is in rule 6: A trace satisfies $\langle\delta\rangle\phi$ just in case it is one of the traces admitted by the program that executes δ and

(T1) $\langle \tau, n \rangle \xrightarrow{n} \langle \tau', \mathit{nil} \rangle$, if $\tau' = \tau^{(1)}$ and $\tau[0, \aleph] = n$;

(T2) $\langle \tau, \delta_1; \delta_2 \rangle \xrightarrow{n} \langle \tau', \gamma; \delta_2 \rangle$, if $\langle \tau, \delta_1 \rangle \xrightarrow{n} \langle \tau', \gamma \rangle$;

(T3) $\langle \tau, \delta_1; \delta_2 \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \delta_1 \rangle \in \mathcal{F}$ and $\langle \tau, \delta_2 \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T4) $\langle \tau, \delta_1 + \delta_2 \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \delta_1 \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$ or $\langle \tau, \delta_2 \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T5) $\langle \tau, \pi x.\delta \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \delta^x_m \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$ for some $m \in \mathcal{N}_x$;

(T6) $\langle \tau, \delta^* \rangle \xrightarrow{n} \langle \tau', \gamma; \delta^* \rangle$, if $\langle \tau, \delta \rangle \xrightarrow{n} \langle \tau', \gamma \rangle$;

(T7) $\langle \tau, \overline{m} \rangle \xrightarrow{n} \langle \tau', \mathit{nil} \rangle$,
if $\langle \tau, n \rangle \xrightarrow{n} \langle \tau', \mathit{nil} \rangle$ and $n \neq m \in \mathcal{N}_A$;

(T8) $\langle \tau, \overline{\delta_1; \delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$, if $\langle \tau, \overline{\delta_1} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T9) $\langle \tau, \overline{\delta_1; \delta_2} \rangle \xrightarrow{n} \langle \tau', \gamma; \overline{\delta_2} \rangle$, if $\langle \tau, \delta_1 \rangle \xrightarrow{n} \langle \tau', \gamma \rangle$;

(T10) $\langle \tau, \overline{\delta_1; \delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \delta_1 \rangle \in \mathcal{F}$ and $\langle \tau, \overline{\delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T11) $\langle \tau, \overline{\delta_1 + \delta_2} \rangle \xrightarrow{n} \langle \tau', \delta'_1 \times \delta'_2 \rangle$,
if $\langle \tau, \overline{\delta_1} \rangle \xrightarrow{n} \langle \tau', \delta'_1 \rangle$ and $\langle \tau, \overline{\delta_2} \rangle \xrightarrow{n} \langle \tau', \delta'_2 \rangle$;

(T12) $\langle \tau, \overline{\delta_1 + \delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \overline{\delta_1} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$ and $\langle \tau, \overline{\delta_2} \rangle \in \mathcal{F}$;

(T13) $\langle \tau, \overline{\delta_1 + \delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if $\langle \tau, \overline{\delta_1} \rangle \in \mathcal{F}$ and $\langle \tau, \overline{\delta_2} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T14) $\langle \tau, \overline{\overline{\delta}} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$, if $\langle \tau, \delta \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$;

(T15) $\langle \tau, \overline{\pi x.\, \delta} \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$,
if for all $n \in \mathcal{N}_x$, $\langle \tau, \delta^x_n \rangle \in \mathcal{F}$ or $\langle \tau, \delta^x_n \rangle \xrightarrow{n} \langle \tau', \delta' \rangle$.

Figure 1: Transition rules for programs

(F1) $\langle \tau, \phi? \rangle \in \mathcal{F}$ if $\tau \models \phi$;

(F2) $\langle \tau, \delta_1; \delta_2 \rangle \in \mathcal{F}$ if $\langle \tau, \delta_1 \rangle \in \mathcal{F}$ and $\langle \tau, \delta_2 \rangle \in \mathcal{F}$;

(F3) $\langle \tau, \delta_1 + \delta_2 \rangle \in \mathcal{F}$ if $\langle \tau, \delta_1 \rangle \in \mathcal{F}$ or $\langle \tau, \delta_2 \rangle \in \mathcal{F}$;

(F4) $\langle \tau, \pi x.\delta \rangle \in \mathcal{F}$ if $\langle \tau, \delta^x_n \rangle \in \mathcal{F}$ for some $n \in \mathcal{N}_x$;

(F5) $\langle \tau, \delta^* \rangle \in \mathcal{F}$;

(F6) $\langle \tau, \overline{\phi?} \rangle \in \mathcal{F}$ if $\tau \not\models \phi$;

(F7) $\langle \tau, \overline{\delta_1; \delta_2} \rangle \in \mathcal{F}$ if $\langle \tau, \overline{\delta_1} \rangle \in \mathcal{F}$ or $\langle \tau, \delta_1; \overline{\delta_2} \rangle \in \mathcal{F}$;

(F8) $\langle \tau, \overline{\delta_1 + \delta_2} \rangle \in \mathcal{F}$ if $\langle \tau, \overline{\delta_1} \rangle \in \mathcal{F}$ and $\langle \tau, \overline{\delta_2} \rangle \in \mathcal{F}$;

(F9) $\langle \tau, \overline{\overline{\delta}} \rangle \in \mathcal{F}$ if $\langle \tau, \delta \rangle \in \mathcal{F}$;

(F10) $\langle \tau, \overline{\pi x.\delta} \rangle \in \mathcal{F}$ if for all $n \in \mathcal{N}_x$, $\langle \tau, \overline{\delta^x_n} \rangle \in \mathcal{F}$.

Figure 2: Finality rules for programs

afterwards tests for ϕ. Note that while a trace is infinite, we require that a successful execution of a program consists of finitely many transition steps leading to a final configuration (2). Our semantics hence follows a similar intuition as the one presented by Meyer (1988), where a terminating program δ corresponds to all traces that start with a sequence of actions compatible with δ, and that afterwards continue indefinitely with the execution of arbitrary actions. A program such as $\mathit{knock}; \mathit{open}(\mathit{door})$ can be viewed as a constraint on traces τ to satisfy $\tau[0, \aleph] = \mathit{knock}$ and $\tau[1, \aleph] = \mathit{open}(\mathit{door})$, without saying anything about how to proceed afterwards (e.g., entering the door or not).

The transition semantics shown above is very similar to the one presented in (Claßen and Lakemeyer 2008; Claßen 2013), which in turn is based on the one for CONGOLOG (De Giacomo, Lespérance, and Levesque 2000), but with the modification that tests are interpreted as *conditions* (rule (F1)) rather than *transitions*. Due to the use of linear-time traces instead of branching-time worlds, transition rule (T1) for primitive actions differs in that it is required that the executed action n is actually the one scheduled to be executed next according to trace τ; in a tree-shaped world w this additional requirement would not be necessary since there is a successor situation for every possible action.

The most obvious change is that Figures 1 and 2 contain additional rules for negated programs, and hence provide a semantics for both negation and joint action. In the next section, we explore some properties and show in what sense defining the new constructs in this fashion is reasonable.

2.3 Basic Action Theories

We can formulate action theories in a similar fashion as in the classical Situation Calculus for encoding dynamic domains. Formally:

Definition 1 (Basic Action Theory)**.** *A* basic action theory *(BAT)* $\Sigma = \Sigma_0 \cup \Sigma_{post}$ *is a set of formulas consisting of:*

1. Σ_0*,* the initial theory*, a finite set of fluent sentences describing the initial state of the world;*

2. Σ_{post}*, a finite set of* successor state axioms *(SSAs) incorporating Reiter's (1991) solution to the frame problem for encoding action effects:*[1]

$$\Box[a]F(\vec{x}) \equiv \gamma_F \tag{3}$$

$$\Box[a]f(\vec{x}) = y \equiv \gamma_f \tag{4}$$

Here it is assumed that one axiom of the form (3) is included for each relational fluent F relevant to the application domain, and one of the form (4) for each functional fluent f relevant to the application domain, and where γ_F is a fluent formula with free variables a and $\vec{x}$, and γ_f a fluent formula with free variables among a, $\vec{x}$, and y.

For simplicity, we don't include a precondition axiom into the BAT. Note that this is without loss of generality when dealing with programs due to the fact that a formula ϕ being

[1] Free variables are understood as universally quantified from the outside, $[t]$ has higher precedence than the logical connectives, and $\Box$ has lower precedence. So $\Box[a]F(\vec{x}) \equiv \gamma_F$ abbreviates $\forall a, \vec{x}.\Box(([a]F(\vec{x})) \equiv \gamma_F)$.

a precondition of action t can be represented by using the program expression $\phi?; t$ in its place.

2.4 Properties

We first note some properties related to joint action and sequence. In what follows, for any two programs δ_1 and δ_2, let $\delta_1 \equiv \delta_2$ stand for $\|\delta_1\| = \|\delta_2\|$.

Proposition 1.

(P1) $\delta_1 \times (\delta_1; \delta_2) \equiv \delta_1; \delta_2$

(P2) $\bot; \delta \equiv \bot$

Property (P1) exemplifies the aforementioned understanding of programs as constraints on the possible future courses of action, where after completing a program δ, infinitely many arbitrary actions will follow. The program $\delta_1; \delta_2$ hence constitutes an additional constraint on the set of traces admitted by δ_1 only. (Meyer 1988) gives the example that "opening the door" together with "opening the door and then leaving" is the same as "opening the door and then leaving". Property (P2) is due to the fact that the set of traces admitted by program $\bot$ is already empty.

It should also be noted that the shift from tree-shaped worlds to linear-time traces comes at a loss of expressiveness, and has the effect that $\mathcal{ESGL}$ is not (directly) comparable to $\mathcal{ES}$ or $\mathcal{ESG}$. A deeper analysis is beyond the scope of this paper, but to illustrate the point, consider the question whether a BAT Σ entails the formula $\neg\Box\phi$. Here, it means that along *every* trace consistent with Σ, there will sooner or later come a point where ϕ does not hold. In $\mathcal{ES}$ and $\mathcal{ESG}$, it means that in every world consistent with Σ there is *some* path where ϕ will be false at some point, which is a much weaker condition. However, it can be argued that for the purpose of planning, it is sufficient to look at the projection problem, which is to decide whether $\Sigma \models [\delta]\psi$ for some fluent formula ψ and some program δ that only mentions fluent formulas as tests, and it can be shown that for this class of reasoning tasks, the logics coincide.

Moreover, it is true that branching-time structures have traditionally been favoured in the deontic logic literature, the main reason being that there must be the possibility to violate a norm, as otherwise, if the future were not open, there would be nothing to reason about in terms of deontic properties. However, as we will see, our linear-time semantics equally allows for this possibility due to the fact that when reasoning about deontic constraints, we will consider *sets* of linear traces, some of which may violate certain deontic constraints, and others don't. Intuitively, this is therefore no real restriction as any tree-shaped branching-time model can be understood as a representation of a set of paths (i.e., traces).

3 Relation to $\mathrm{PD_eL}$

In this section we argue that the definitions and extensions presented in the previous section are reasonable in the sense that among other things, they capture Meyer's (1988) dynamic deontic logic $\mathrm{PD_eL}$. It is based on Anderson's (1958) proposal of reducing deontic logic to alethic modal logic by using a distinguished propositional variable V that intuitively represents a "bad state" or the violation of a norm. For dynamic logic, one defines

$$\mathbf{F}\delta \doteq [\delta]V \tag{5}$$

saying that an action δ is forbidden if its execution leads to a violation. Permission and obligation can then be defined in terms of prohibition as usual:

$$\mathbf{P}\delta \doteq \neg\mathbf{F}\delta \quad \text{and} \quad \mathbf{O}\delta \doteq \mathbf{F}\overline{\delta} \tag{6}$$

Obviously, this requires the action algebra to include an operator for negating actions in order to represent "ought-to-do" obligations $\mathbf{O}\delta$. As it turns out, the question of how to define the negation of a complex action is far from trivial[2]. For $\mathrm{PD_eL}$, Meyer presents the following five axioms as desiderata that he argues must "reasonably hold" for $\overline{\delta}$:

(N1) $\overline{\overline{\delta_1}} \equiv \delta_1$

(N2) $\overline{\delta_1; \delta_2} \equiv \overline{\delta_1} + \delta_1; \overline{\delta_2}$

(N3) $\overline{\delta_1 + \delta_2} \equiv \overline{\delta_1} \times \overline{\delta_2}$

(N4) $\overline{\delta_1 \times \delta_2} \equiv \overline{\delta_1} + \overline{\delta_2}$

(N5) $\overline{\phi \to \delta_1/\delta_2} \equiv \phi \to \overline{\delta_1}/\overline{\delta_2}$

The most interesting of these properties is probably (N2). It says that there are exactly two possible ways of executing the negation of a sequential program $\delta_1; \delta_2$: Either do something next that is "not δ_1", or if doing δ_1, then do something afterwards that is "not δ_2". The last property defines the negation of a conditional action ("if ϕ then δ_1 else δ_2"), which we can define in GOLOG by means of

$$\phi \to \delta_1/\delta_2 \doteq [\phi?; \delta_1] + [\neg\phi?; \delta_2] \tag{7}$$

With the definition presented in the previous section, we get:

Proposition 2. *(N1)–(N5) are valid in $\mathcal{ESGL}$.*

Meyer's action algebra does not include tests, the Kleene star, or pick operators ($\mathrm{PD_eL}$ is propositional). While for the pick operator, which essentially behaves like an existential quantifier, there is no obvious dual, we note that our transition semantics is compatible for tests and iteration in the following sense:

(N6) $\overline{\phi?} \equiv \neg\phi?$

(N7) $\overline{\delta^*} \equiv \bot$

(N6) follows from (N5) and the fact that $\phi? \equiv \phi \to nil/\bot$, using $\overline{\bot} \equiv nil$ and $\overline{nil} \equiv \bot$. (N7) makes sense when considering the expansion law for the Kleene star

$$\delta^* \equiv nil + \delta; \delta^*$$

as then

$$\overline{\delta^*} \equiv \overline{nil + \delta; \delta^*} \equiv \overline{nil} \times \overline{\delta; \delta^*} \equiv \bot \times \overline{\delta; \delta^*} \equiv \bot.$$

Based on (N1)–(N5), Meyer proposes the system $\mathrm{PD_eL}$ as given by the axioms and inference rules shown in Figure 3. He argues that this is sufficient to entail many important theorems of deontic logic (the paper lists 36 of them) when the deontic modalities are understood according to (5) and (6). We note that $\mathcal{ESGL}$ subsumes $\mathrm{PD_eL}$ as follows, assuming that $duration(\delta)$ denotes the maximal length of action sequences admitted by δ:

[2] See (Claßen and Delgrande 2020, Section 3.2) for a brief discussion. Alternatives to Meyer's definition were proposed e.g. by van der Meyden (1996) and Broersen (2004a).

Axioms

(PC) all tautologies of propositional logic

$(\Box\supset)$ $[\delta](\phi_1 \supset \phi_2) \supset [\delta]\phi_1 \supset [\delta]\phi_2$

$(;)$ $[\delta_1\ ;\ \delta_2]\phi \equiv [\delta_1][\delta_2]\phi$

$(+)$ $[\delta_1 + \delta_2]\phi \equiv [\delta_1]\phi \wedge [\delta_2]\phi$

$(\times)$ $[\delta_1 \times \delta_2]\phi \subset [\delta_1]\phi \vee [\delta_2]\phi$
(provided $duration(\delta_1) = duration(\delta_2)$)

$(\to)$ $[\phi \to \delta_1/\delta_2]\psi \equiv (\phi \supset [\delta_1]\psi) \wedge (\neg\phi \supset [\delta_2]\psi)$

$(\Diamond)$ $\langle\delta\rangle\phi \equiv \neg[\delta]\neg\phi$

$(\overline{;})$ $[\overline{\delta_1\ ;\ \delta_2}]\phi \equiv [\overline{\delta_1}]\phi \wedge [\delta_1][\overline{\delta_2}]\phi$

$(\overline{+})$ $[\overline{\delta_1 + \delta_2}]\phi \subset [\overline{\delta_1}]\phi \vee [\overline{\delta_2}]\phi$
(provided $duration(\delta_1) = duration(\delta_2)$)

$(\overline{\times})$ $[\overline{\delta_1 \times \delta_2}]\phi \equiv [\overline{\delta_1}]\phi \wedge [\overline{\delta_2}]\phi$

$(\overline{\to})$ $[\overline{\phi \to \delta_1/\delta_2}]\psi \equiv (\phi \supset [\overline{\delta_1}]\psi) \wedge (\neg\phi \supset [\overline{\delta_2}]\psi)$

$(\overline{\ })$ $[\overline{\overline{\delta}}]\phi \equiv [\delta]\phi$

$(\bot)$ $[\bot]\phi$

Rules

(MP) From ϕ and $\phi \supset \psi$ infer ψ.

(N) From ϕ infer $[\delta]\phi$.

Figure 3: The system PDeL

Proposition 3. *In $\mathcal{ESGL}$, axioms (PC)–($\bot$) are valid, and inference rules (MP) and (N) are sound.*

We remark that under our transition semantics, programs in general do not constitute a Boolean algebra (Figure 4):

Proposition 4. *Axioms (B1) – (B10) of Boolean algebras are valid in $\mathcal{ESGL}$, but axioms (B11) and (B12) are not.*

This means that while the usual laws of associativity, commutativity, neutral elements, distributivity, and idempotency apply, the complement does not always behave as expected. A simple counterexample for (B12) is the program $\delta = (a + a; b); c$, where a, b and c are primitive actions. A trace τ that executes $\langle a, b, c\rangle$ as its first three actions (i.e. $\tau[0, \aleph] = a$, $\tau[1, \aleph] = b$, $\tau[2, \aleph] = c$) is an execution of *both* δ and its negation:

$$\langle\tau, (a + a; b); c\rangle \xrightarrow{a} \langle\tau^{(1)}, b; c\rangle \xrightarrow{b} \langle\tau^{(2)}, c\rangle \xrightarrow{c} \langle\tau^{(3)}, nil\rangle$$

$$\langle\tau, \overline{(a + a; b); c}\rangle \xrightarrow{a} \langle\tau^{(1)}, \overline{c}\rangle \xrightarrow{b} \langle\tau^{(2)}, nil\rangle$$

Intuitively, this is due to property (N2): One way of executing the negation of a sequence $\delta_1; \delta_2$ is to execute δ_1, followed by the negation of δ_2. Here, doing action a is one way of executing $(a + a; b)$, and doing action b is one way of executing the negation of action c. While this behaviour may (or may not) be undesirable, note that this is already possible in Meyer's original system $\text{PD}_\text{e}\text{L}$ (and does not conflict with any results concerning deontic properties). To avoid it, one would have to include counterparts of (B11) and (B12) as additional axioms for $\text{PD}_\text{e}\text{L}$. The appendix in (Meyer 1988) provides the definition for a semantics satisfying these additional properties, but is (arguably) more involved than what we present here. In particular, it requires to consider *sets* of traces, rather than traces, as the semantical domain.

(B1) $(\delta_1 + \delta_2) + \delta_3 \equiv \delta_1 + (\delta_2 + \delta_3)$ ($+$ is associative)

(B2) $\delta_1 + \delta_2 \equiv \delta_2 + \delta_1$ ($+$ is commutative)

(B3) $\delta_1 + \bot \equiv \delta_1$ ($\bot$ is the neutral element wrt $+$)

(B4) $\delta_1 + (\delta_2 \times \delta_3) \equiv (\delta_1 + \delta_2) \times (\delta_1 + \delta_3)$ ($+$ distrib.)

(B5) $\delta_1 + \delta_1 \equiv \delta_1$ ($+$ is idempotent)

(B6) $(\delta_1 \times \delta_2) \times \delta_3 \equiv \delta_1 \times (\delta_2 \times \delta_3)$ ($\times$ is associative)

(B7) $\delta_1 \times \delta_2 \equiv \delta_2 \times \delta_1$ ($\times$ is commutative)

(B8) $\delta_1 \times \top \equiv \delta_1$ ($\top$ is the neutral element wrt $\times$)

(B9) $\delta_1 \times (\delta_2 + \delta_3) \equiv (\delta_1 \times \delta_2) + (\delta_1 \times \delta_3)$ ($\times$ distrib.)

(B10) $\delta_1 \times \delta_1 \equiv \delta_1$ ($\times$ is idempotent)

(B11) $\delta_1 + \overline{\delta_1} \equiv \top$ (complement wrt $+$)

(B12) $\delta_1 \times \overline{\delta_1} \equiv \bot$ (complement wrt $\times$)

Figure 4: Axioms of a Boolean algebra

3.1 Representing the Chisholm Scenario

In (Meyer, Dignum, and Wieringa 1994) and (Meyer, Wieringa, and Dignum 1998), the authors suggest to address contrary-to-duty obligations by extending the formalism to include multiple violation atoms $V_1, V_2, V_3, \ldots$ and use accordingly indexed deontic modalities. The Chisholm scenario, for example, could then be expressed as follows:

$$\mathbf{O}_1 h \tag{8}$$

$$\mathbf{F}_2(\overline{t}; h) \tag{9}$$

$$\mathbf{F}_3(t; \overline{h}) \tag{10}$$

saying that one ought help the neighbour, that it is forbidden to not tell and then help, and that is also prohibited to tell followed by not helping. The fact that one actually does not go to help cannot be represented explicitly because of dynamic logic being about hypothetical reasoning in the form of "*if* a certain action is taken, *then* a certain result is obtained."

With the additional assumption that violations persist (by including $V_i \supset [\delta]V_i$ as an additional axiom schema), it is now possible to reason about sub-ideal states in terms of which norms have been violated. For example, telling followed by helping will result in V_1 (the first norm is still violated because we didn't help immediately as next action), but telling and not helping in $V_1 \wedge V_2 \wedge V_3$, so the former should be preferred over the latter.

There are multiple drawbacks to this approach. First, a preference relation among states with different violations has to be defined explicitly. While this arguably allows for a certain flexibility, e.g. to say that some violations are more severe than others, the number of combinations to be considered grows exponentially with the number of violation atoms, i.e., constraints. Second, the authors "admit that it would be far nicer to have a representation closer to the natural language representation, but this would call for a non-trivial extension of $\mathrm{PD_eL}$, in which one can also reason 'backward' directly." Third, notice that even in the intuitively ideal case where the agent tells and actually helps, constraint 8 will cause a violation due to the fact that helping was not the immediate next action. What is missing is a notion of an agent *intending* to help in the foreseeable future. We will address these issues in the following sections.

4 Simple Temporal Conditionals

In this section we show that $\mathcal{ESGL}$ is also capable of capturing the approach presented in (Claßen and Delgrande 2020), where deontic constraints are expressed as conditionals over (a restricted set of) GOLOG programs. Specifically, here we are interested in conditionals of the temporal kind that allow to represent scenarios such as the Chisholm set. The set of GOLOG programs in question is as follows:

Definition 2 (Guarded-Action Fragment)**.** *The set of* guarded actions *is given by the following grammar:*

$$\gamma \ ::= \ t \mid \pi x.\gamma \mid \phi?;\gamma$$

The guarded-action fragment *is then given by*

$$\delta \ ::= \ \gamma \mid \overline{\delta} \mid \delta + \delta \mid \delta \times \delta$$

where γ is a guarded action.

Guarded actions hence are primitive actions, possibly preceded by a sequence of picks and test conditions, and the guarded-action fragment is all their Boolean combinations. An important special case is the "wildcard" action $\star \doteq \pi a.a$. We note that

Proposition 5. *The guarded-action fragment is a Boolean algebra with $\star$ as neutral element wrt $\times$.*

This means the laws shown in Figure 4 are valid for this restricted set if we substitute $\star$ for $\top$. Moreover, note that for such programs, joint execution distributes over sequence:

Proposition 6. *Let $\delta_1, \ldots, \delta_4$ be of the guarded-action fragment. Then* $\quad(\delta_1;\delta_2) \times (\delta_3;\delta_4) \equiv (\delta_1 \times \delta_3);(\delta_2 \times \delta_4)$.

We then use programs from the guarded-action fragment to express deontic constraints as described below.

Definition 3 (Temporal Conditionals)**.** *A* temporal deontic conditional *is an expression that is of the form*

$$\delta \Rightarrow_a \gamma \quad \textit{or} \quad \delta \Rightarrow_b \gamma$$

where δ and γ are from the guarded-action fragment. We read $\delta \Rightarrow_a \gamma$ as "if committed to doing δ, the agent ought to do γ afterwards*", and $\delta \Rightarrow_b \gamma$ as "…* before*." This definition includes the special case of unconditional constraints where $\delta = \star$. The* materialization *of a rule is given by*

$$\mathfrak{M}(\delta \Rightarrow_a \gamma) \ \doteq \ \overline{\delta};\star + \star;\gamma$$

$$\mathfrak{M}(\delta \Rightarrow_b \gamma) \ \doteq \ \star\,;\overline{\delta} + \gamma;\star$$

For a finite set of rules $\rho = \{r_1, \ldots, r_k\}$ we understand $\mathfrak{M}(\rho)$ as $\mathfrak{M}(r_1) \times \cdots \times \mathfrak{M}(r_k)$, where $\mathfrak{M}(\emptyset) = \star;\star$.

A set of such defeasible conditionals now induces a ranking over traces using a construction similar to the one for *rational closure* (Kraus, Lehmann, and Magidor 1990):

Definition 4 (Ranking)**.** *Given a finite set ρ of temporal conditionals over programs and a set of traces e, a* ranking *of the rules in ρ wrt e is given by*

$$\begin{aligned} \rho^e_0 \ &= \ \rho \\ \rho^e_{i+1} \ &= \ \{(\delta \Rightarrow_a \gamma) \in \rho^e_i \mid e \cap \|\mathfrak{M}(\rho^e_i) \times (\delta;\star)\| = \emptyset\} \cup \\ &\quad\ \{(\delta \Rightarrow_b \gamma) \in \rho^e_i \mid e \cap \|\mathfrak{M}(\rho^e_i) \times (\star;\delta)\| = \emptyset\} \end{aligned}$$

Rules $r \in \rho^e_{i+1}$ are called exceptional *wrt ρ^e_i. For every $\tau \in e$, the* rank assigned by ρ wrt e *then is*

$$\mathsf{Rank}(\tau, e, \rho) = \min\{i \mid \tau \in \|\mathfrak{M}(\rho^e_i)\|\}.$$

The cumulative rank assigned by e *to any time point $k \in \mathbb{N}$ is given by the sum of all ranks from times 0 up to k:*

$$\mathsf{CRank}(\tau, e, \rho, k) = \sum_{i=0}^{k} \mathsf{Rank}(\tau^{(i)}, e^{(i)}, \rho)$$

where $e^{(i)} = \{\tau^{(i)} \mid \tau \in e\}$.

Here we follow (Claßen and Delgrande 2020) for aggregating ranks over time points by simply summing them up. Intuitively, this means that a trace will be ranked as less ideal the more "bad" actions are performed in it. In particular, there is no way of undoing a bad act, and any further bad deed makes the course of action less and less ideal.

4.1 Representing the Chisholm Scenario

Using simple temporal constraints, the first three statements of the Chisholm scenario can be expressed as:

$$\star \ \Rightarrow_a \ \mathit{help} \tag{11}$$

$$\mathit{help} \ \Rightarrow_b \ \mathit{tell} \tag{12}$$

$$\overline{\mathit{help}} \ \Rightarrow_b \ \overline{\mathit{tell}} \tag{13}$$

The first rule states that generally, the agent ought to go help the neighbours. The second one means that when the agent intends to go and help, it should tell the neighbours immediately before. If on the other hand, says the third rule, the agent does not intend to go and help, it ought not tell them. Again, the fourth statement of the Chisholm set cannot be represented explicitly due to reasoning in this formalism being purely hypothetical.

Assume that $e = \mathcal{T}$ is the set of all traces. In the following, let h stand for the action term help, and t for tell. Materializing $\rho_0 = \{(11), (12), (13)\}$ then yields:

$$\begin{aligned} \mathfrak{M}((11)) \ &= \ (\overline{\star};\star) + (\star;h) && \equiv (\star;h) \\ \mathfrak{M}((12)) \ &= \ && (\star;\overline{h}) + (t;\star) \\ \mathfrak{M}((13)) \ &= \ (\star;\overline{\overline{h}}) + (\overline{t};\star) && \equiv (\star;h) + (\overline{t};\star) \end{aligned}$$

Recall that $\mathfrak{M}(\rho_0)$ is given by the conjunction of these three expressions. Since according to Proposition 5, the usual distributive laws apply, we can "multiply" them out. Observe that $(\star;h)$ is incompatible with $(\star;\overline{h})$, and that $(t;\star)$ contradicts with $(\overline{t};\star)$. The result is hence equivalent to $(\star;h) \times (t;\star)$, which in turn can be simplified to $(t;h)$ using Propositions 5 and 6. We thus get

$$\begin{aligned} \|(t;h) \times (\star;\star)\| &= \|t;h\| && \neq \emptyset \\ \|(t;h) \times (\star;h)\| &= \|t;h\| && \neq \emptyset \\ \|(t;h) \times (\star;\overline{h})\| &= \|\bot\| && = \emptyset \end{aligned}$$

Because rule (13) is the only exceptional one, we obtain $\rho_1 = \{(13)\}$ and $\mathfrak{M}(\rho_1) \equiv \star; h + \bar{t}; \star$. The rule is obviously not exceptional with itself, so $\rho_2 = \emptyset$, hence $\mathfrak{M}(\rho_2) = \star; \star$. We thus end up with

$$\mathfrak{M}(\rho_0) \equiv t; h, \;\; \mathfrak{M}(\rho_1) \equiv \star; h + \bar{t}; \star, \;\; \mathfrak{M}(\rho_2) \equiv \star; \star$$

which induces the following ranking:

$$\mathsf{Rank}(\tau, e, \rho) = \begin{cases} 0, & \tau[0, \aleph] = \mathit{tell} \text{ and } \tau[1, \aleph] = \mathit{help} \\ 1, & \tau[0, \aleph] \neq \mathit{tell} \\ 2, & \tau[0, \aleph] = \mathit{tell} \text{ and } \tau[1, \aleph] \neq \mathit{help} \end{cases}$$

4.2 Compiling Conditionals into BATs

In (Claßen and Delgrande 2020), we also showed how deontic constraints can be compiled into the action theory, so that after a preprocessing step, no special (non-monotonic) reasoning machinery is needed for planning under deontic constraints. The basic idea is to use a new function fluent ideal that represents the degree of ideality of the current situation. For this purpose, we include

$$\mathit{ideal} = 0, \tag{14}$$

into the initial theory Σ_0 of our BAT. The value of this fluent may increase due to actions, as per the SSA

$$\Box[a]\mathit{ideal} = \mathit{ideal} + \mathit{bad}(a) \tag{15}$$

that we include into Σ_{post}. The potential increase is determined by another fluent $\mathit{bad}(a)$ that expresses how "bad" an action a is, and that we define further below. First, to keep track of which programs mentioned in deontic constraints have been executed previously, we introduce finitely many additional fluent predicates $\mathit{Did}(\gamma)$, where for each one Σ_0 contains the axiom

$$\neg \mathit{Did}(\gamma) \tag{16}$$

and Σ_{post} contains the SSA

$$\Box[a]\mathit{Did}(\gamma) \equiv \mathfrak{C}[\gamma, a]. \tag{17}$$

The right-hand side of the SSA uses the compilation operator $\mathfrak{C}$ whose definition is given below:

Definition 5.

1. $\mathfrak{C}[\alpha, a] = (a = \alpha)$

2. $\mathfrak{C}[\phi?; \delta, a] = \phi \wedge \mathfrak{C}[\delta, a]$

3. $\mathfrak{C}[\pi v.\delta, a] = \exists v.\, \mathfrak{C}[\delta, a]$

4. $\mathfrak{C}[\overline{\delta}, a] = \neg\mathfrak{C}[\delta, a]$, *if δ is a guarded action*

5. $\mathfrak{C}[\delta_1 + \delta_2, a] = \mathfrak{C}[\delta_1, a] \vee \mathfrak{C}[\delta_2, a]$

6. $\mathfrak{C}[\delta_1 \times \delta_2, a] = \mathfrak{C}[\delta_1, a] \wedge \mathfrak{C}[\delta_2, a]$

7. $\mathfrak{C}[\gamma; \delta, a] = \mathit{Did}(\gamma) \wedge \mathfrak{C}[\delta, a]$

With this operator, we can now define an axiom for bad. Suppose that we determined a ranking as shown previously, then for a finite number of rule sets $\rho_0, \ldots, \rho_k$, we obtained their materialized counterparts

$$\mathfrak{M}(\rho_0) \equiv \delta_0, \quad \mathfrak{M}(\rho_1) \equiv \delta_1, \quad \ldots \quad \mathfrak{M}(\rho_k) \equiv \delta_k$$

where each δ_i is a program from the guarded-action fragment. We then define the badness of action a as the *minimal* index i whose δ_i admits a:

$$\mathit{bad}(a) = b \equiv \bigvee_{i=0}^{k} (b = i) \wedge \mathfrak{C}[\delta_i, a] \;\wedge \bigwedge_{j=0}^{i-1} \neg\mathfrak{C}[\delta_j, a] \tag{18}$$

Proposition 7. *Let Σ be a BAT, ρ a set of simple temporal constraints, Σ_ρ be the result of extending Σ with axioms (14) – (18), and $e = \{\tau \mid \tau \models \Sigma_\rho\}$. For any $\tau \in e$ and $k \in \mathbb{N}$,*

$$\mathsf{CRank}(\tau, e, \rho, k) = d \;\; \text{iff} \;\; \tau^{(k)} \models (\mathit{ideal} = d).$$

In the Chisholm example we obtain, after simplifications,

$$\begin{aligned} \mathit{bad}(a) = b \;\equiv\; & b = 0 \wedge \mathit{Did}(\mathit{tell}) \wedge a = \mathit{help} \,\vee \\ & b = 1 \wedge \mathit{Did}(\overline{\mathit{tell}}) \,\vee \\ & b = 2 \wedge \mathit{Did}(\mathit{tell}) \wedge a \neq \mathit{help} \end{aligned} \tag{19}$$

where the SSAs for $\mathit{Did}(\mathit{tell})$ and $\mathit{Did}(\overline{\mathit{tell}})$ are given by

$$\Box[a]\mathit{Did}(\mathit{tell}) \;\equiv\; a = \mathit{tell} \tag{20}$$

$$\Box[a]\mathit{Did}(\overline{\mathit{tell}}) \;\equiv\; a \neq \mathit{tell} \tag{21}$$

5 Intentional Conditionals

One shortcoming of the approaches discussed in the previous sections is that it is assumed that one action under consideration will follow immediately after the other. This is obviously not a realistic assumption for many practical scenarios. For example, helping the neighbour might necessitate other actions, such as buying supplies at the hardware store. In this section, we hence explore the idea of including a notion of intention, represented by temporal modalities over programs with a finite horizon. Specifically, for any δ from the guarded-action fragment and $k \geq 1$, we will use $\Box_k\delta$ to say "during k steps, always δ", defined through

$$\Box_k\delta \;\doteq\; \delta^k \;\doteq\; \underbrace{\delta; \cdots; \delta}_{k \text{ times}}, \tag{22}$$

and $\Diamond_k\delta$ to express "within k steps, eventually δ", given by

$$\Diamond_k\delta \;\doteq\; \delta \,+\, \star; \delta \,+\, \star; \star; \delta \,+\, \cdots \,+\, \underbrace{\star; \cdots; \star}_{k-1 \text{ times}}; \delta. \tag{23}$$

We note that the two operators are indeed duals:

Proposition 8.

(N8) $\overline{\Box_k\delta} \;\equiv\; \Diamond_k\overline{\delta}$

(N9) $\overline{\Diamond_k\delta} \;\equiv\; \Box_k\overline{\delta}$

Definition 6 (Intentional Guarded-Action Fragment)**.** *The* intentional guarded-action fragment *is given by*

$$\delta \;::=\; \gamma \mid \Box_k\gamma \mid \Diamond_k\gamma \mid \overline{\delta}$$

where γ is from the guarded-action fragment.

Definition 7 (Intentional Conditionals)**.** *An* intentional deontic conditional *is an expression of the form*

$$\delta \Rightarrow \gamma$$

where δ and γ are programs of the intentional guarded-action fragment. The materialization *of a rule is given by*

$$\mathfrak{M}(\delta \Rightarrow \gamma) \;\doteq\; \overline{\delta} \,+\, \gamma$$

For a finite set $\rho = \{r_1, \ldots, r_k\}$ we understand $\mathfrak{M}(\rho)$ as $\mathfrak{M}(r_1) \times \cdots \times \mathfrak{M}(r_k)$ as before, but using $\mathfrak{M}(\emptyset) = \top$.

Definition 8 (Intentional Situation Ranking)**.** *Given a finite set ρ of intentional conditionals and a set of traces e, an* intentional ranking *of the rules in ρ wrt e is given by*

$$\begin{aligned} \rho_0^e \;&=\; \rho \\ \rho_{i+1}^e \;&=\; \{(\delta \Rightarrow \gamma) \in \rho_i^e \mid e \cap \|\mathfrak{M}(\rho_i^e) \times \delta\| = \emptyset\} \end{aligned}$$

$\mathsf{Rank}(\tau, e, \rho)$ *and* $\mathsf{CRank}(\tau, e, \rho, k)$ *are exactly as in Def. 4.*

5.1 Representing the Chisholm Scenario

Suppose we want to apply a finite horizon of $k \geq 2$. The first three statements of the Chisholm scenario could be expressed as:

$$\begin{aligned} \star &\Rightarrow \Diamond_k \mathit{help} && (24)\\ \Diamond_k \mathit{help} &\Rightarrow \mathit{tell} && (25)\\ \overline{\Diamond_k \mathit{help}} &\Rightarrow \overline{\mathit{tell}} && (26) \end{aligned}$$

Assume again that $e = \mathcal{T}$ is the set of all traces, and let h and t abbreviate help and tell, respectively. Materializing $\rho_0 = \{(24), (25), (26)\}$ then yields:

$$\begin{aligned} \mathfrak{M}((24)) &= \overline{\star} + \Diamond_k h && \equiv \Diamond_k h\\ \mathfrak{M}((25)) &= \overline{\Diamond_k h} + t && \equiv \Box_k \overline{h} + t\\ \mathfrak{M}((26)) &= \overline{\overline{\Diamond_k h}} + \overline{t} && \equiv \Diamond_k h + \overline{t} \end{aligned}$$

Again, we determine the product of these expressions and apply the distributive law. Observe that $\Diamond_k h$ is incompatible with $\Box_k \overline{h}$, and that t contradicts with $\overline{t}$. The result is hence equivalent to $\Diamond_k h \times t$. We thus get

$$\begin{aligned} &\|(\Diamond_k h \times t) \times \star\| && = \|\Diamond_k h \times t\| && \neq \emptyset\\ &\|(\Diamond_k h \times t) \times \Diamond_k h\| && = \|\Diamond_k h \times t\| && \neq \emptyset\\ &\|(\Diamond_k h \times t) \times \Box_k \overline{h}\| && = \|\bot\| && = \emptyset \end{aligned}$$

Similar to before, we obtain $\rho_1 = \{(26)\}$ and $\rho_2 = \emptyset$, hence

$$\mathfrak{M}(\rho_0) \equiv \Diamond_k h \times t,\;\; \mathfrak{M}(\rho_1) \equiv \Diamond_k h + \overline{t},\;\; \mathfrak{M}(\rho_2) \equiv \top$$

which induces the following ranking:

$$\mathsf{Rank}(\tau, e, \rho) = \begin{cases} 0, & \tau[0, \aleph] = \mathit{tell} \text{ and}\\ & \tau[i, \aleph] = \mathit{help} \text{ for some } 1 \leq i \leq k\\ 1, & \tau[0, \aleph] \neq \mathit{tell}\\ 2, & \tau[0, \aleph] = \mathit{tell} \text{ and}\\ & \tau[i, \aleph] \neq \mathit{help} \text{ for all } 1 \leq i \leq k \end{cases}$$

Note that for $k = 2$, we get exactly the same behaviour as with simple temporal conditionals in the previous section.

5.2 Compiling Conditionals into BATs

The compilation method works on intentional conditionals as well. The only thing we have to ensure is that negation is only applied to program expressions from the guarded-action fragment. For this purpose, we introduce the two following rules in addition to ones stated in Definition 5:

8. $\mathfrak{C}[\overline{\Box_k \delta}, a] = \mathfrak{C}[\Diamond_k \overline{\delta}, a]$
9. $\mathfrak{C}[\overline{\Diamond_k \delta}, a] = \mathfrak{C}[\Box_k \overline{\delta}, a]$

In the non-negated case, the existing rules can be applied using (22) and (23). For the Chisholm example, we get for horizon $k = 3$, again after simplifications:

$$\begin{aligned} &\mathit{bad}(a) = b \;\equiv && (27)\\ &\qquad b = 0 \wedge (\mathit{Did}(\mathit{tell} + \mathit{tell}; \star)) \wedge a = \mathit{help} \vee\\ &\qquad b = 1 \wedge (\mathit{Did}(\overline{\mathit{tell}} + \overline{\mathit{tell}}; \star) \vee\\ &\qquad b = 2 \wedge (\mathit{Did}(\mathit{tell} + \mathit{tell}; \star)) \wedge a \neq \mathit{help} \end{aligned}$$

with the additional SSAs

$$\Box[a]\mathit{Did}(\mathit{tell} + \mathit{tell}; \star) \;\equiv\; a = \mathit{tell} \vee \mathit{Did}(\mathit{tell}) \qquad (28)$$

$$\Box[a]\mathit{Did}(\overline{\mathit{tell}} + \overline{\mathit{tell}}; \star) \;\equiv\; a \neq \mathit{tell} \vee \mathit{Did}(\overline{\mathit{tell}}) \qquad (29)$$

6 Discussion

The Chisholm paradox has received a great deal of attention in the literature. To name but a few, works in the early 1980s (van Eck 1982; Loewer and Belzer 1983) presented solution proposals based on temporal extensions of deontic logic, identifying the scenario's temporal nature as a vital aspect that SDL is unable to appropriately represent. Van der Torre and Tan (1998) argued that these were still insufficient for Chisholm's original set, as this requires conditionalization where the antecedent (going to help) refers to a later point in time than the consequent (telling). They go on to present a formalization based on *stit* ("see to it that") semantics (Horty 2001), modified to include a preference relation over histories. While these kinds of analyses yielded valuable theoretical insights into the nature of the problem, the proposed formalisms do not lend themselves well to practical implementations, e.g. due to the fact that actions in stit – other than planning languages or action formalisms such as the Situation Calculus – do not have proper names or types, but are described purely through their effects.

In this paper, we proposed a new formalism for reasoning about contrary-to-duty scenarios based on a modal variant of the Situation Calculus that allows to express postconditions for complex actions in the form of programs from the GOLOG agent language. By employing a special semantics based on linear-time traces rather than branching-time tree models, we could integrate non-trivial notions of action negation and joint execution of programs. We showed that the approach is more general than two existing ones due to Meyer (1988) and Claßen and Delgrande (2020), and presented a third, more expressive alternative involving a simple notion of intention.

This line of research is work in progress, and there are many avenues for future work. On the technical side, it could be argued that having a program semantics where the entire set of programs constitutes a Boolean algebra is desirable, so as to be able to apply all the "usual" laws to such expressions. It is conceivable to do this by adopting a definition more similar to the one of (Meyer 1988), albeit at the cost of being less simple, and less close to the original transition semantics of GOLOG.

Regarding expressivity, besides supporting a larger fragment of GOLOG programs in deontic conditionals, it would be interesting to explore more sophisticated notions of intentions to formulate constraints. While it is certainly possible to come up with infinitary versions of the temporal operators $\Box_k$ and $\Diamond_k$, in most cases it seems reasonable to apply a certain form of deadline. The latter may be of a temporal nature (e.g., the neighbour needs our help on the same day), and so for a more realistic representation we could incorporate an explicit, quantitative notion of time, instead of just crudely counting the number of actions. However, a more general approach could be to allow for arbitrary conditions as deadlines, for instance by adopting an approach due to Broersen (2004b) that uses operators $M(\rho \leq \delta)$ to express the motivation to achieve ρ before δ becomes true (e.g., have the neighbour's roof fixed before it starts raining).

Finally, it will be interesting to combine the notions of actions and obligations we considered here not only with intentions, but also beliefs, and study their interplay. This is similar to how BOID architectures (Broersen et al. 2001) have been proposed to generalize beliefs, desires, intention (Bratman 1987) by including obligations and norms.

Acknowledgements

We gratefully acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada and thank the referees for their valuable comments.

References

Anderson, A. R. 1958. A reduction of deontic logic to alethic modal logic. *Mind* 67(265):100–103.

Bartha, P. 1999. Moral preference, contrary-to-duty obligation and defeasible oughts. *Norms, logics and information systems: new studies in deontic logic and computer science* 93–108.

Bratman, M. 1987. *Intentions, Plans, and Practical Reason*. Cambridge University Press.

Broersen, J. M.; Dastani, M.; Hulstijn, J.; Huang, Z.; and van der Torre, L. W. N. 2001. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proc. AGENTS 2001*, 9–16. ACM Press.

Broersen, J. M. 2004a. Action negation and alternative reductions for dynamic deontic logics. *Journal of Applied Logic* 2(1):153–168.

Broersen, J. M. 2004b. On the logic of 'being motivated to achieve rho, before delta'. In *Proc. JELIA 2004*, 334–346. Springer.

Chellas, B. F. 1980. *Modal Logic: An Introduction*. Cambridge University Press.

Chisholm, R. M. 1963. Contrary-to-duty imperatives and deontic logic. *Analysis* 24(2):33–36.

Claßen, J., and Delgrande, J. 2020. Dyadic obligations over complex actions as deontic constraints in the situation calculus. In *Proc. KR 2020*, 253–263. ijcai.org.

Claßen, J., and Lakemeyer, G. 2008. A logic for non-terminating Golog programs. In *Proc. KR 2008*, 589–599. AAAI Press.

Claßen, J. 2013. *Planning and Verification in the Agent Language Golog*. Ph.D. Dissertation, Department of Computer Science, RWTH Aachen University.

De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence* 121(1–2):109–169.

Gabbay, D.; Horty, J.; Parent, X.; van der Meyden, R.; and van der Torre, L. 2013. *Handbook of Deontic Logic and Normative Systems*. College Publications.

Hansson, B. 1969. An analysis of some deontic logics. *Noûs* 3(4):373–398.

Horty, J. F. 2001. *Agency and Deontic Logic*. Oxford University Press.

Kraus, S.; Lehmann, D. J.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44(1-2):167–207.

Lakemeyer, G., and Levesque, H. J. 2010. A semantic characterization of a useful fragment of the situation calculus with knowledge. *Artificial Intelligence* 175(1):142–164.

Levesque, H. J.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31(1–3):59–83.

Loewer, B., and Belzer, M. 1983. Dyadic deontic detachment. *Synthese* 54(2):295–318.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. New York: American Elsevier. 463–502.

Meyer, J.-J. C.; Dignum, F.; and Wieringa, R. J. 1994. The paradoxes of deontic logic revisited: A computer science perspective. Technical Report UU-CS-1994-38, Utrecht University.

Meyer, J. C.; Wieringa, R. J.; and Dignum, F. 1998. The role of deontic logic in the specification of information systems. In *Logics for Databases and Information Systems (Proceedings of the the Dagstuhl Seminar 9529: Role of Logics in Information Systems, 1995)*, 71–115. Kluwer Academic Publishers.

Meyer, J. C. 1988. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic* 29(1):109–136.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy* 359–380.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

van der Meyden, R. 1996. The dynamic logic of permission. *Journal of Logic and Computation* 6(3):465–479.

van der Torre, L. W. N., and Tan, Y. 1998. The temporal analysis of Chisholm's paradox. In *Proc. AAAI 1998*, 650–655. AAAI Press.

van Eck, J. A. 1982. A system of temporally relative modal and deontic predicate logic and its philosophical applications. *Logique et Analyse* 25(99):249–290.

von Wright, G. H. 1951. Deontic logic. *Mind* 60(237):1–15.

von Wright, G. H. 1963. *Norm and action: a logical enquiry*. Routledge and Kegan Paul.

Inertial Causal Calculus

Alexander Bochman
Computer Science Department, Holon Institute of Technology, Israel
bochmana@hit.ac.il

Abstract

We present a modification of dynamic causal calculus from (Bochman 2014), which is based on a semantic representation of the inertia principle and thereby naturally corresponds to the action description language $\mathcal{B}$ in the classification of Gelfond and Lifschitz (1998). This will allow us to provide a systematic comparison between action languages $\mathcal{B}$ and $\mathcal{C}$ and their extensions in a single formal framework. It will be shown, in particular, that this causal representation constitutes a strengthening of the original transition semantics for the language $\mathcal{B}$ that makes the resulting system fully equivalent to the language $\mathcal{C}$ coupled with syntactic inertia rules.

1 Introduction

Numerous formalisms of reasoning about action and change have been suggested in the AI literature. It is sometimes difficult to adjudicate the relative advantages and shortcomings of these formalisms, not only because they are often formulated in entirely different languages, but also because there seems to be no general agreement about what they *should* describe. Much is left here to intuitions of the authors, or to difficulties in describing some intuitively desirable notions. Worse still, a large part of these intuitions, even in high-level action description languages, is implicitly or explicitly biased toward the main existing low-level representations of actions in logic programming or situation calculus (see, e.g., (Lifschitz and Turner 1999; Turner 1997)). Though such a bias is advantageous from the implementation point of view, it is not helpful in determining the required scope and expressive capabilities of such action theories from the representational perspective.

Our aim in this paper consists in determining and clarifying the main ingredients of reasoning about action and change in AI. We will pursue a top-down approach to this subject, namely we will make use of the dynamic causal calculus, introduced in (Bochman 2014), as a general logical formalism for such a reasoning, but will attempt to generalize it further in order to encompass action description languages that are based on a semantic representation of the inertia principle. This will give us an opportunity to compare in a single formal framework different representations of inertia and thereby to provide further insight into the general scope of dynamic modeling in AI.

Guided mainly by the practical goal of providing an efficient representation for reasoning about actions and change, the majority of theories for such a reasoning in AI have followed the lead of the situation calculus (McCarthy and Hayes 1969) in adopting the inertia assumption as a basis for an alternative representation of temporal dynamics. A salient advantage of the use of inertia in action descriptions is that it provides both a more succinct and more natural representation of such a dynamics. However, the inertia assumption almost inevitably leads to a triple of its own notorious difficulties known as the frame, ramification and qualification problems (see, e.g., (Shanahan 1997)). It has been realized quite early that classical logic and its temporal/dynamic extensions, taken by themselves, encounter difficulties in resolving these problems. More precisely, it has become clear that these problems have an essentially nonmonotonic character, so their proper solution requires augmenting purely logical, monotonic deductive reasoning with an appropriate mechanism for making nonmonotonic conclusions.

A dominant approach to resolving these problems in AI has been based, in one form or another, on causal reasoning. The corresponding causal closure assumption (see, e.g., (Reiter 2001)) is a particular form of the Law of Causation, according to which all facts that hold in any given situation should either be caused by previous actions, or else preserve their truth-values in time due to the accompanying inertia assumption. A direct incorporation of such causal assertions into the language of the situation calculus has been proposed in (Lin 1995; Lin 1996) and has been shown to provide a natural account of both the frame and ramification problems.

The causal calculus has been suggested in (McCain and Turner 1997) as a general logical framework for this kind of dynamic causal reasoning. A 'temporalized' version of the causal calculus has been used as a logical basis of the action description language $\mathcal{C}$ and of its descendant $\mathcal{C}+$ (Giunchiglia et al. 2004). In this version, the inertia principle has been encoded syntactically using a special kind of dynamic causal rules of inertia. On the other hand, a direct *semantic* description of the inertia assumption has also been suggested in (McCain and Turner 1995) (see also (Przymusinski and Turner 1997)), and it has been employed, in effect, as a semantic basis of action languages $\mathcal{A}$ and $\mathcal{B}$ and their descendants. The relations and differences between these two ways of encoding the inertia assumption, as well the corresponding differences between the languages $\mathcal{B}$ and $\mathcal{C}$ have occupied a significant place in subsequent action literature — see, e.g., (Gelfond and Lifschitz 2012; Zhang and Lin 2017). A number of formal systems that combine the features of these two languages has also been suggested — see, e.g., (Lee, Lifschitz, and Yang 2013).

Our background language in this paper will be an ordinary propositional language with the classical connectives and constants $\{\wedge, \vee, \neg, \rightarrow, \mathbf{t}, \mathbf{f}\}$. $\vDash$ will stand for the classical entailment, while Th will denote the classical provability operator. At the end of the paper, however, we will briefly discuss a possibility, and even desirability, of generalizing this underlying logical formalism to a logical system that could incorporate, for instance, meaning postulates and definitions, including recursive ones. Note in this respect that most of the constructions and results, described below, will remain to hold if we would replace the classical entailment in its role as a background logic with an arbitrary *supraclassical* consequence relation, that is, a consequence relation in a classical language that subsumes classical entailment.

2 Dynamic Causal Calculus

Dynamic causal calculus has been introduced in (Bochman 2014) as a nonmonotonic theory purported to serve as a general formalism for dynamic causal reasoning. The formalism has been based on *dynamic* causal rules of the form

$$B \,.\, C \Rightarrow E,$$

where B, C and E are classical propositions. These dynamic causal rules have an informal meaning *"After B, C causes E"*. By the intended interpretation, such a rule describes a dynamic transition from a state that satisfies proposition B to a subsequent state in which E is caused by C. Such rules naturally correspond, for instance, to action rules, or dynamic action laws, of the form

$$\textbf{caused } E \textbf{ if } C \textbf{ after } B$$

that constitute a common syntactic core of many action description languages today (see, e.g., (Lang, Lin, and Marquis 2003)).

In addition to the above action rules, however, an important feature of the majority of action description languages, as well as corresponding theories of action and change, consists in the explicit use of *static* causal rules, or state constraints. In particular, it is these rules that allow us to provide a succinct and efficient description of both ramifications and qualifications in action descriptions. As we will see in what follows, beside the different treatments of the inertia assumption, the difference between action languages $\mathcal{B}$ and $\mathcal{C}$ stems in a large part from a different understanding (and even different underlying logics) of such static rules.

In the framework of the dynamic causal calculus, static causal rules have been identified with a special kind of dynamic rules of the form $\mathbf{t} \,.\, A \Rightarrow B$, where $\mathbf{t}$ is the truth constant. In other words, the following definition has been adopted:

$$A \Rightarrow B \;\equiv_{df}\; \mathbf{t} \,.\, A \Rightarrow B.$$

According to this definition, static causal rules are rules that are valid after any legitimate transition. It has been shown that the resulting formalism is sufficiently expressive to capture the action description language $\mathcal{C}$ in the classification of (Gelfond and Lifschitz 1998) and its generalizations such as the language $\mathcal{C}+$ from (Giunchiglia et al. 2004).

The reduction of static causal rules to dynamic rules has made dynamic causal rules the only kind of rules of the dynamic causal calculus. In accordance with this, by a *dynamic causal theory* we will mean in what follows an arbitrary set of dynamic causal rules.

2.1 Nonmonotonic Transition Semantics

The nonmonotonic transition semantics, described below, can be viewed as a central component of the dynamic causal calculus. It determines, in a sense, even the corresponding logical formalism of dynamic causal inference that will be described in the next section.

The guiding principle behind the nonmonotonic transition semantics is a thorough enforcement of the dynamic Law of Causality, according to which every state of a dynamic model should be explained (i.e., caused) by a preceding state and the active causal rules.

By a world we will mean a maximal consistent set of classical propositions of the language. Such worlds will also be called *states* in this paper. Also, given the current objectives of this study, nonmonotonic transitions as defined in (Bochman 2014) will be called $\mathcal{C}$-transitions in what follows.

Given a dynamic causal theory Δ and two states α, β, we will denote by $\Delta(\alpha \,.\, \beta)$ the set of propositions that are caused due to a transition from α to β.

$$\{C \mid A.B \Rightarrow C \in \Delta \text{ for some } A \in \alpha, B \in \beta.\}$$

Definition 1.(i) A pair of states (α, β) will be called a $\mathcal{C}$-*transition* with respect to a dynamic causal theory Δ if β is the unique model of $\Delta(\alpha.\beta)$, that is

$$\beta = \mathrm{Th}(\Delta(\alpha \,.\, \beta)).$$

(ii) A $\mathcal{C}$-*transition model* of a dynamic causal theory Δ is a set of states $\mathfrak{S}$ such that, for any $\beta \in \mathfrak{S}$ there is $\alpha \in \mathfrak{S}$ such that (α, β) is a $\mathcal{C}$-transition wrt Δ.

A $\mathcal{C}$-transition is a transition between two states in which the resulting state is fully explained (caused), given the preceding state and the causal laws of the domain. In this respect, the above definition of a *transition model* extends the Law of Causality to the states themselves by requiring, in effect, that every state should have sufficient reasons for its occurrence. A discussion about the role and ramifications of this global constraint on the set of possible states can be found in (Bochman 2014).

If (α, β) is a $\mathcal{C}$-transition, then $\Delta(\alpha.\beta)$ is included in β. As a consequence, the output world of any transition is always closed with respect to the static causal rules (for the definition of the latter, given above), namely if $A \Rightarrow B$ belongs to Δ and $A \in \beta$, then $B \in \beta$. Moreover, any state of a $\mathcal{C}$-transition model is also an output of some $\mathcal{C}$-transition.

Accordingly, we immediately obtain that any state of a $\mathcal{C}$-transition model of a dynamic causal theory Δ is closed with respect to the static causal rules of Δ.

The above definition of a $\mathcal{C}$-transition almost coincides with the definition of a *causally explained transition*, given in (Giunchiglia and Lifschitz 1998) for the action description language $\mathcal{C}$, a predecessor of $\mathcal{C}+$. In fact, the only difference between the two definitions is that (Giunchiglia and Lifschitz 1998) required further that both the initial and resulting states of such a transition should be closed with respect to the static causal laws. On the above construction, this additional requirement is accounted for, respectively, as a by-product of the definition of static causal rules on the one hand (for the resulting states), and a $\mathcal{C}$-transition model on the other hand (for the initial states).

It can be easily verified that the union of two $\mathcal{C}$-transition models of a causal theory is also a $\mathcal{C}$-transition model. Consequently, if a dynamic causal theory has at least one $\mathcal{C}$-transition model, it has a unique maximal such model. The latter has been called a *canonical $\mathcal{C}$-transition model* of a dynamic causal theory.

It is difficult to guarantee, in general, that there exists a $\mathcal{C}$-transition from a given state to some other state. This will change, however, if we will take into account the inertia principle. In the framework of $\mathcal{C}$-transition models, this principle can be represented syntactically using the following special dynamic causal rules for literal fluents:

(Inertia) $l \,.\, l \Rightarrow l$

The above rule implies that if a literal l holds in some state, it becomes a default for every state that can be reached from this state by a direct $\mathcal{C}$-transition. In other words, l will continue to hold in such a subsequent state, unless an opposite literal will be caused to hold in it. Consequently, unless there is an inherent inconsistency in the description of causal rules, any state of a canonical $\mathcal{C}$-transition model will have at least one possible subsequent state (including possible persistence in the same state). This syntactic encoding of the inertia principle has been adopted in the action description language $\mathcal{C}$ as an important part of its representation framework.

2.2 Transition Inference Relation

As with other formalisms for nonmonotonic reasoning, dynamic causal rules presuppose a certain underlying logic that agrees with the above nonmonotonic semantics. Such a logic provides a formal description of the associated dynamic causal inference.

A *transition inference relation* is a set of dynamic causal rules $A \,.\, B \Rightarrow C$ that satisfies the postulates described below.

The first group of postulates states that a set of dynamic causal rules with a fixed first premise (B) should satisfy the postulates of 'ordinary' causal inference described first in (Bochman 2003):

(T-Strengthening) $A \vDash C$ and $B \,.\, C \Rightarrow E$ imply $B \,.\, A \Rightarrow E$;

(T-Weakening) $E \vDash D$ and $B \,.\, C \Rightarrow E$ imply $B \,.\, C \Rightarrow D$;

(T-And) $B \,.\, C \Rightarrow E$ and $B \,.\, C \Rightarrow D$ imply $B \,.\, C \Rightarrow E \land D$;

(T-Or) If $B \,.\, C \Rightarrow E$ and $B \,.\, D \Rightarrow E$, then $B \,.\, C \lor D \Rightarrow E$;

(T-Cut) If $B \,.\, A \Rightarrow C$ and $B \,.\, A \land C \Rightarrow D$, then $B \,.\, A \Rightarrow D$;

(T-Truth) $\mathbf{t} \,.\, \mathbf{t} \Rightarrow \mathbf{t}$;

(T-Falsity) $\mathbf{t} \,.\, \mathbf{f} \Rightarrow \mathbf{f}$.

In view of the above postulates, dynamic causal rules $B \,.\, C \Rightarrow E$ can be seen as ordinary causal rules $C \Rightarrow E$ that are conditioned by the preceding (background) context B.

In addition, the next two postulates describe the logical properties of this preceding context in dynamic causal rules:

(B-Strengthening) $A \vDash B$ and $B \,.\, C \Rightarrow E$ imply $A \,.\, C \Rightarrow E$;

(B-Or) If $A \,.\, C \Rightarrow E$ and $B \,.\, C \Rightarrow E$, then $A \lor B \,.\, C \Rightarrow E$.

The combined effect of the above postulates is that the associated semantic interpretation of dynamic causal inference could be a possible world semantics in which both the two premises and conclusion of a dynamic causal rule are evaluated with respect to worlds (complete states). Such a semantics has been described in (Bochman 2014) in terms of Kripke frames with ternary accessibility relations.

Correspondences. Dynamic causal rules can be extended to rules having arbitrary sets of propositions as premises using compactness: for any sets u, v of propositions, we can define $u \,.\, v \Rightarrow A$ as follows:

$$u \,.\, v \Rightarrow A \;\equiv\; \bigwedge a \,.\, \bigwedge b \Rightarrow A, \text{ for some finite } a \subseteq u, b \subseteq v.$$

For a pair (u, v) of sets of propositions, $\mathcal{C}(u \,.\, v)$ denotes the set of propositions caused by the pair, that is

$$\mathcal{C}(u \,.\, v) = \{A \mid u \,.\, v \Rightarrow A\}.$$

The causal operator $\mathcal{C}$ can be viewed as a derivability operator corresponding to a transition inference relation. Note that it is a monotonic operator. Also, due to T-Weakening and T-And, $\mathcal{C}(u \,.\, v)$ will always be a deductively closed set:

$$\mathcal{C}(u \,.\, v) = \mathrm{Th}(\mathcal{C}(u \,.\, v)).$$

For any dynamic causal theory Δ there exists a least transition inference relation $\Rightarrow_\Delta$ that includes Δ. Clearly, $\Rightarrow_\Delta$ is the set of all dynamic causal rules that can be derived from Δ using the postulates for transition inference relations. The derivability operator corresponding to $\Rightarrow_\Delta$ has been denoted by $\mathcal{C}_\Delta$.

Since a transition inference relation can also be viewed as a (rather large) dynamic causal theory, the definition of transitions for the latter can be immediately extended to transition inference relations. Moreover, due to the logical properties of transition inference, the definition of a $\mathcal{C}$-transition can now be simplified, namely a pair of worlds (α, β) will be a $\mathcal{C}$-transition with respect to a transition inference relation if and only if it satisfies the following equality:

$$\beta = \mathcal{C}(\alpha \,.\, \beta).$$

It has been shown in (Bochman 2014) that the logic of transition inference is adequate for reasoning with respect to the nonmonotonic transition semantics of dynamic causal theories, since it preserves the latter. Namely, it has been shown that $\mathcal{C}$-transitions of a dynamic causal theory Δ coincide with $\mathcal{C}$-transitions of $\Rightarrow_\Delta$.

2.3 Determinate and literal causal theories

A common simplifying assumption in theories of action and change amounts to a syntactic restriction of the underlying logical language to classical literals.

A dynamic causal rule will be called *determinate* if its head is a literal or a logical constant $\mathbf{t}$ or $\mathbf{f}$. A dynamic causal theory will be called *determinate* if it consists of only determinate causal rules.

A stronger simplifying assumption amounts to restriction of dynamic causal rules to *literal* causal rules of the form

$$L \,.\, L' \Rightarrow l,$$

where l is a literal, while L and L' are finite sets of literals. Note, however, that due to the postulates *T-Or* and *B-Or*, any determinate causal rule is reducible to a set of such literal rules, and therefore any determinate dynamic causal theory is logically reducible to a literal one.

Under this restriction, it turns out to be convenient to represent states (worlds) as maximal consistent sets of literals.

In what follows, $Lit(\alpha)$ will denote the set of literals that belong to a state α. We will use $s, t, \ldots$ for denoting 'literal' states in this sense. Then it turns out that a $\mathcal{C}$-transition can be characterized as a pair (s, t) of such states that satisfies the following simple fixpoint condition:

$$t = \Delta(s \,.\, t).$$

The following result confirms that this simplified description provides an equivalent characterization of $\mathcal{C}$-transitions for this restricted case:

Lemma 1. *If Δ is literal dynamic causal theory, then a pair of states (α, β) is a $\mathcal{C}$-transition if and only if*

$$Lit(\beta) = \Delta(Lit(\alpha) \,.\, Lit(\beta)).$$

3 Inertial Transition Semantics

Now we are going to modify the formalism of the dynamic causal calculus by introducing an alternative notion of a transition. More precisely, in contrast to the syntactic encoding of the inertia principle that has been used with the notion of a $\mathcal{C}$-transition above, the notion of a $\mathcal{B}$-transition, described below, will provide a direct semantic representation of inertia.

Definition 2.(i) A pair (α, β) of states will be called a $\mathcal{B}$-*transition* with respect to a dynamic causal theory Δ if

$$\beta = \mathrm{Th}((Lit(\alpha) \cap \beta) \cup \Delta(\alpha \,.\, \beta)).$$

(ii) A $\mathcal{B}$-*transition model* of a dynamic causal theory Δ is a set of states $\mathfrak{S}$ such that, for any $\beta \in \mathfrak{S}$ there is $\alpha \in \mathfrak{S}$ such that (α, β) is a $\mathcal{B}$-transition wrt Δ.

In contrast to $\mathcal{C}$-transitions, $\mathcal{B}$-transitions are determined not only by active causal rules, but also by literals that persist in the transition; such literals remain to hold due to inertia, and therefore they do not require causal explanation.

Remark. The restriction of the inertia principle to literals only is based on the idea that compound logical formulas are completely determined *logically* by their constituting literals, so their temporal behavior is also fully dependent on the temporal behavior of the latter; they cannot have life (i.e., temporal evolution) of their own. It is important to note, however, that general definitions of transitions are formulated for complete worlds, so they impose the Law of Causality on all propositional formulas. Consequently, since the reasons why a compound logical formula holds in a given state cannot be based directly on the inertia principle, they should be obtained either in terms of active static or dynamic causal rules, or else as a logical consequence of the corresponding reasons for their (literal) constituents.

For determinate dynamic causal theories, the description of $\mathcal{B}$-transitions can be simplified as follows.

Lemma 2. *A pair of states (α, β) is a $\mathcal{B}$-transition with respect to a determinate dynamic causal theory Δ iff*

$$Lit(\beta) \setminus \alpha \subseteq \Delta(\alpha \,.\, \beta) \subseteq \beta.$$

Proof. The direction from left to right is trivial. For the other direction, since β is deductively closed, the inclusion $\mathrm{Th}((Lit(\alpha) \cap \beta) \cup \Delta(\alpha \,.\, \beta)) \subseteq \beta$ amounts to $\Delta(\alpha \,.\, \beta) \subseteq \beta$. Note now that if Δ is a determinate causal theory, then $(Lit(\alpha) \cap \beta) \cup \Delta(\alpha \,.\, \beta)$ is just a set of literals, and consequently the reverse inclusion holds if and only if

$$Lit(\beta) \subseteq (Lit(\alpha) \cap \beta) \cup \Delta(\alpha \,.\, \beta).$$

The latter inclusion amounts to $Lit(\beta) \subseteq \alpha \cup \Delta(\alpha \,.\, \beta)$, which is equivalent to $Lit(\beta) \setminus \alpha \subseteq \Delta(\alpha \,.\, \beta)$. □

As before with $\mathcal{C}$-transitions, we still have that if (α, β) is a $\mathcal{B}$-transition, then $\Delta(\alpha.\beta)$ is included in β. Therefore, the output state β is also closed with respect to the static causal rules of the causal theory. Moreover, since any state of a $\mathcal{B}$-transition model is also an output of some $\mathcal{B}$-transition, we obtain again

Corollary 3. *Any state of a $\mathcal{B}$-transition model of a dynamic causal theory Δ is closed with respect to the static causal rules of Δ.*

As before, the union of two $\mathcal{B}$-transition models of a causal theory is also a $\mathcal{B}$-transition model. Consequently, if a dynamic causal theory has at least one $\mathcal{B}$-transition model, it has a unique maximal such model; the latter can be called the *canonical $\mathcal{B}$-transition model* of a dynamic causal theory.

As a key result for this study, the next theorem will establish that, modulo the inertia principle, the nonmonotonic semantics based, respectively, on $\mathcal{C}$-transitions and $\mathcal{B}$-transitions are essentially equivalent.

For a dynamic causal theory Δ, we will denote by Δ_I a theory obtained from Δ by adding inertia rules of the form $l \,.\, l \Rightarrow l$ for all literals of the language. Then we can obtain the following theorem.

Theorem 4. *A pair of states (α, β) is a $\mathcal{B}$-transition of a dynamic causal theory Δ if and only if it is a $\mathcal{C}$-transition of Δ_I.*

The proof is actually straightforward; it follows immediately from the fact that, for any states α, β, the set $\Delta_I(\alpha\,.\,\beta)$ coincides with $(Lit(\alpha) \cap \beta) \cup \Delta(\alpha\,.\,\beta)$.

The above result shows that, taken by itself, the difference between syntactic and semantic representations of the inertia principle is not essential for describing causal dynamics; both systems of dynamic causal reasoning can be used much for the same purposes and situations. This implies, in particular, that sources of the actual differences between action languages $\mathcal{B}$ and $\mathcal{C}$ lie elsewhere. As we will see below, they arise solely from a different interpretation of static causal rules.

3.1 Correspondences

A transition inference relation can also be viewed as a dynamic causal theory, so the definition of $\mathcal{B}$-transitions can be immediately extended to transition inference relations. Still, the resulting definition of a $\mathcal{B}$-transition for transition inference will remain much the same:

$$\beta = \mathrm{Th}((Lit(\alpha) \cap \beta) \cup \mathcal{C}(\alpha\,.\,\beta)).$$

The following result will show that the logic of transition inference is adequate also for reasoning with respect to the inertial transition semantics, since it preserves the latter.

Theorem 5. *If Δ is a dynamic causal theory, then $\mathcal{B}$-transitions of Δ coincide with $\mathcal{B}$-transitions of $\Rightarrow_\Delta$.*

Proof sketch. If C_Δ is the provability operator corresponding to $\Rightarrow_\Delta$, then it can be shown that, for any 'causally consistent' pair of worlds α, β, $C_\Delta(\alpha.\beta)$ coincides with $\mathrm{Th}(\Delta(\alpha.\beta))$. Moreover, since $\mathrm{Th}((Lit(\alpha) \cap \beta) \cup \Delta(\alpha\,.\,\beta))$ is classically equivalent to $\mathrm{Th}((Lit(\alpha) \cap \beta) \cup \mathrm{Th}(\Delta(\alpha\,.\,\beta)))$, it becomes easy to verify that $\mathcal{B}$-transitions of Δ will coincide with $\mathcal{B}$-transitions of $\Rightarrow_\Delta$. □

Finally, for literal dynamic causal theories, the description of $\mathcal{B}$-transitions can be reduced to the following equality for the corresponding sets of literals:

$$t = (s \cap t) \cup \Delta(s\,.\,t).$$

The following result shows, in effect, that this simplified description provides an equivalent characterization of $\mathcal{B}$-transitions for the restricted case of literal dynamic causal theories:

Lemma 6. *If Δ is literal dynamic causal theory, then a pair of states (α, β) is a $\mathcal{B}$-transition wrt Δ if and only if*

$$Lit(\beta) = \Delta(Lit(\alpha)\,.\,Lit(\beta)) \cup (Lit(\alpha) \cap Lit(\beta)).$$

4 Comparisons with Action Description Languages $\mathcal{A}$ and $\mathcal{B}$

In this last section we will describe relations between the inertial transition semantics of dynamic causal theories and the action languages $\mathcal{A}$ and $\mathcal{B}$.

Action language $\mathcal{A}$. A closest counterpart of dynamic causal rules has been introduced by Pednault in (Pednault 1989) as conditional action rules of action description language ADL; these rules had the form:

A **causes** l **if** L

where A is an action name, l is a literal, and L is a set (or conjunction) of literals. This language has been called language $\mathcal{A}$ in (Gelfond and Lifschitz 1998).

The semantics of the language $\mathcal{A}$, given in (Gelfond and Lifschitz 1998), can be described as a set of transitions (s, A, t), where A is an action name and s, t are states (maximal consistent sets of fluent literals) that satisfy the following condition:

$$E(A, s) \subseteq t \subseteq E(A, s) \cup s, \qquad (*)$$

where $E(A, s)$ is the set of heads of all action rules of the form A **causes** l **if** L in the action description such that $L \subseteq s$.

Now we will identify the action rules of $\mathcal{A}$ with literal dynamic causal rules of the form

$$L\,.\,A \Rightarrow l.$$

As can be seen, action names are incorporated simply as new propositional atoms in this translation. However, just as in (Bochman 2014), they will be exempted from the inertia principle and treated as exogenous literals. On this translation, a transition (s, A, t) will be represented as a transition (s, t), where t is an 'extended' state that already includes action atom A.

In addition, we should restrict possible transitions to transitions produced by single actions; in other words, we should prevent concurrent actions. This can be achieved by accepting the following *static* causal rules:

$$A, B \Rightarrow \mathbf{f}$$

for any two different actions A and B.

Let Δ_D denote the dynamic causal theory that corresponds in this sense to an action description D in the language $\mathcal{A}$. Then we have

Theorem 7. *Transitions of an action description D in the language $\mathcal{A}$ coincide with $\mathcal{B}$-transitions of Δ_D.*

Proof sketch. Due to the fact that only actions appear as second premises in the causal rules of Δ_D, and that there is no concurrency of actions, it is easy to verify that, for any legitimate transition (s, t), $\Delta(s\,.\,t)$ will coincide with $\Delta(s\,.\,A)$, for some action name A. Consequently, the inertial $\mathcal{B}$-semantics for Δ_D can be described as a set of transitions (s, t) that satisfy the equality

$$t = (s \cap t) \cup \Delta(s\,.\,A),$$

for some action name A (see Lemma 6). Note, however, that $\Delta(s\,.\,A)$ for this causal theory is just the set of direct effects of action A in a state s (plus, of course, A itself); in other words, it coincides with $E(A, s)$ on fluent literals. Moreover, it is easy to show that the above equality is equivalent to the condition (*) above (cf. (Gelfond and Lifschitz 1998, fn.13)). Accordingly, the semantics of $\mathcal{B}$-transitions corresponds precisely to the semantic interpretation of $\mathcal{A}$. □

Action language $\mathcal{B}$. The action description language $\mathcal{B}$ is obtained by augmenting the language $\mathcal{A}$ with *static laws* of the form

$$l \textbf{ if } L,$$

where l is a literal and L a set of literals. These static laws are viewed as plain inference rules that allow, in particular, to derive indirect effects of actions.

For a set Z of static laws, let $\mathbb{C}\mathrm{n}_Z(s)$ denote the least set of literals that contains s and is closed with respect to the rules from Z. Then the semantics of $\mathcal{B}$ is defined as a set of transitions (s, A, t), where s and t are literal states that are closed with respect to the static laws and satisfy the equation

$$t = \mathbb{C}\mathrm{n}_Z((s \cap t) \cup E(A, s)). \qquad (**)$$

As before, $E(A, s)$ above is the set of heads of all action rules of the form A **causes** l **if** L from the dynamic description such that $L \subseteq s$.

The language $\mathcal{B}$ has been generalized to language $\mathcal{AL}$ (see, e.g., (Baral and Gelfond 2005)) which has lifted the restriction to single actions and thereby has allowed concurrency, and has added explicit impossibility conditions for executability of actions; such executability conditions are covered in the dynamic causal calculus by constraints of the form

$$A \,.\, B \Rightarrow \mathbf{f},$$

where A is a fluent proposition, and B is an action formula.

Now let us attempt to represent static laws of the language $\mathcal{B}$ as static causal rules $L \Rightarrow l$, in accordance with the definition of such causal rules in the dynamic causal calculus. Then it is easy to verify that any transition (s, t) that satisfies the above equation (**) will be a $\mathcal{B}$-transition in our sense:

Lemma 8. *Any admissible transition (s, A, t) of an action description in the langage $\mathcal{B}$ corresponds to a $\mathcal{B}$-transition of the corresponding dynamic causal theory.*

Still, the two semantics do not coincide, because there are $\mathcal{B}$-transitions that are not transitions by the original definition for the language $\mathcal{B}$. The following example, adapted from (Zhang and Lin 2017), illustrates this.

Example 1. Let us consider a dynamic causal theory that consists of a single dynamic rule $\mathbf{t} \,.\, A \Rightarrow f_2$ and the following set of static causal rules:

$$f_2, f_3 \Rightarrow f_1 \quad f_2, \neg f_3 \Rightarrow f_1 \quad f_1, f_2 \Rightarrow f_3.$$

It can be verified that the pair of states $(\{\neg f_1, \neg f_2, \neg f_3\}, \{f_1, f_2, f_3\})$ is not an admissible transition for the original semantics of $\mathcal{B}$. However, it will be a $\mathcal{B}$-transition with respect to this causal theory[1], because $\Delta(s \,.\, t)$ in this case will include f_2 as a direct effect of A and will be closed with respect to the above static rules (see below).

In assessing this discrepancy, we should take into account that the underlying logic of static causal rules in the dynamic causal calculus is different from, and even incomparable with, the implicit logic of static laws in the language $\mathcal{B}$.

To begin with, static causal rules of the causal calculus admit the classical logical rule of Disjunction in the Antecedent:

(Or) If $C \Rightarrow E$ and $D \Rightarrow E$, then $C \vee D \Rightarrow E$.

In the dynamic causal calculus, the above rule follows from the postulate *T-Or* of transition inference (see Section 1.2). In contrast, static laws of the language $\mathcal{B}$ do not admit this rule. For instance, the first two static rules in the above example imply $f_2 \Rightarrow f_1$ by the rule *Or*, but if we would add this static rule to the theory, the above transition would become admissible for the language $\mathcal{B}$. Actually, the rule Or has played a key role in the characterization of the difference between the languages $\mathcal{B}$ and $\mathcal{C}$ that has been made in (Zhang and Lin 2017) (see their Postulate 5).

The discrepancy between the inertial semantics of $\mathcal{B}$-transitions and the 'official' semantics of the language $\mathcal{B}$ does not arise in cases when the set of static rules is acyclic. Indeed, (Gelfond and Lifschitz 2012) have shown that the semantics of $\mathcal{B}$ and $\mathcal{C}$ coincide, in effect, for action theories in which the dependence graph of their static rules is acyclic. This result can be immediately adapted for showing that, for such causal theories, the original semantics of the language $\mathcal{B}$ will coincide with the semantics of $\mathcal{B}$-transitions.

This naturally brings us to the second crucial aspect of the difference between the languages $\mathcal{B}$ and $\mathcal{C}$. In the language $\mathcal{B}$, static laws are viewed as plain inference rules, and therefore they freely admit the logical postulate of Reflexivity, namely $A \Rightarrow A$. In contrast, in the causal calculus and language $\mathcal{C}$ such rules have non-trivial content, and they are actually used in a formal representation of defaults, exogenous propositions and actions. On the other hand, the non-causal, inferential understanding of static laws in the framework of $\mathcal{B}$ has allowed to employ them, for instance, for describing defined propositions and predicates, including recursive definitions that play an important role in the general representation methodology behind the use of the languages $\mathcal{B}$ and especially $\mathcal{AL}$. Thus, the ability to use such recursive constructs has been crucial for modeling systems and for the development of industrial size planning and diagnostic applications (see, e.g., (Balduccini, Gelfond, and Nogueira 2006; Son et al. 2006; Tu et al. 2011)).

A comprehensive treatment of the latter discrepancy falls beyond the scope of this study, already because it would require a generalization of our formalism to a first-order logical language. Nevertheless, we suggest that a proper resolution of this problem could be provided only if we will abandon an unjustified 'purist' presumption that a causal theory of reasoning about actions and change should be based exclusively on causal reasoning. Causal reasoning is not a replacement of logic, but its extension, or complement, for situations where we do not have logically sufficient knowledge. It is the logical background of a causal formalism and its underlying logic that should be a proper place for defining

[1] When restricted to fluent literals.

new predicates and connectives, including recursive definitions and meaning postulates. All this could be done while still retaining a separate category of *causal* static rules when they are appropriate. Of course, this could create obvious problems for implementing such combined descriptions, for instance in logic programming. Still, such implementation problems should not detract us from a no less important task of providing an *adequate representation* of reasoning in dynamic domains.

A more detailed and systematic discussion of the relations between causal reasoning and logic can be found in (Bochman 2021).

5 Conclusions

The primary objective of this paper was to show that a suitably generalized dynamic causal calculus could provide a unified logical basis for reasoning about actions and change in AI. Being combined with the wealth of representation capabilities of such a reasoning, demonstrated in corresponding studies based on the languages $\mathcal{C}$, $\mathcal{C}+$, $\mathcal{B}$ and $\mathcal{AL}$, the results described in the paper indicate that a theory of dynamic causal inference can be viewed as a self-subsistent logical theory for reasoning in dynamic domains.

Of course, much work still has to be done in order to extend this causal framework to description of more complex processes that involve, for instance, temporally extended actions and events, concurrency and triggered (natural) events. We see these issues as important topics for future work.

References

Balduccini, M.; Gelfond, M.; and Nogueira, M. 2006. Answer set based design of knowledge systems. *Annals of Mathematics and Artificial Intelligence* 47:183–219.

Baral, C., and Gelfond, M. 2005. Logic programming and reasoning about actions. In Fisher, M.; Gabbay, D.; and Vila, L., eds., *Handbook of Temporal Reasoning in Artificial Intelligence*. Elsevier. 389–426.

Bochman, A. 2003. A logic for causal reasoning. In *IJCAI-03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 141–146. Morgan Kaufmann.

Bochman, A. 2014. Dynamic causal calculus. In Baral, C.; Giacomo, G. D.; and Eiter, T., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*. AAAI Press.

Bochman, A. 2021. *A Logical Theory of Causality*. MIT Press.

Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on Artificial Intelligence* 3:195–210.

Gelfond, M., and Lifschitz, V. 2012. The common core of action languages B and C. In *Working Notes of the International Workshop on Nonmonotonic Reasoning (NMR)*.

Giunchiglia, E., and Lifschitz, V. 1998. An action language based on causal explanation: Preliminary report. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 623–630. AAAI Press.

Giunchiglia, E.; Lee, J.; Lifschitz, V.; McCain, N.; and Turner, H. 2004. Nonmonotonic causal theories. *Artificial Intelligence* 153:49–104.

Lang, J.; Lin, F.; and Marquis, P. 2003. Causal theories of action: A computational core. In *IJCAI-03: Proceedings of the 8th International Joint Conference on Artificial Intelligence*, 1073–1078.

Lee, J.; Lifschitz, V.; and Yang, F. 2013. Action language BC: Preliminary report. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China*.

Lifschitz, V., and Turner, H. 1999. Representing transition systems by logic programs. In *Logic Programming and Nonmonotonic Reasoning, 5th International Conference, LPNMR'99, El Paso, Texas, USA, December 2-4, 1999, Proceedings*, 92–106.

Lin, F. 1995. Embracing causality in specifying the inderect effect of actions. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-95*, 1985–1991. Morgan Kaufmann.

Lin, F. 1996. Embracing causality in specifying the indeterminate effects of actions. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence: AAAI-96*, 670–676.

McCain, N., and Turner, H. 1995. A causal theory of ramifications and qualifications. In Mellish, C., ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1978–1984. San Francisco: Morgan Kaufmann.

McCain, N., and Turner, H. 1997. Causal theories of action and change. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 460–465.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence*. Edinburg University Press. 463–502.

Pednault, E. P. D. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proceedings of the 1st International Conference on Knowledge Representation and Reasoning, KR-89*, 324–332.

Przymusinski, T. C., and Turner, H. 1997. Update by means of inference rules. *The Journal of Logic Programming* 30(2):125–143.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamic Systems*. MIT Press.

Shanahan, M. P. 1997. *Solving the Frame Problem*. The MIT Press.

Son, T. C.; Baral, C.; Tran, N.; and Mcilraith, S. 2006. Domain-dependent knowledge in answer set planning. *ACM Transactions on Computational Logic* 7(4):613–657.

Tu, P. H.; Son, T. C.; Gelfond, M.; and Morales, A. R. 2011. Approximation of action theories and its application to conformant planning. *Artificial Intelligence* 175(1):79–119.

Turner, H. 1997. Representing actions in logic programs and default theories: A situation calculus approach. *J. Log. Program.* 31(1-3):245–298.

Zhang, H., and Lin, F. 2017. Characterizing causal action theories and their implementations in answer set programming. *Artificial Intelligence* 248:1–8.

Formalizing the Three Player Card Game Using the Language m$\mathcal{A}^*$

Loc Pham , Tran Cao Son , Enrico Pontelli
New Mexico State University, Las Cruces NM 88003, USA
$\{$lpham,tson,epontelli$\}$@cs.nmsu.edu

Abstract

This paper presents a formalization of the well-known "*Three Player Card Game*," a simple domain with static causal laws and nondeterministic actions, in the high level action language m$\mathcal{A}^*$. The formalization shows how m$\mathcal{A}^*$ allows us to make the same conclusions as approaches using Dynamic Epistemic Logic (DEL) when observabilities of agents are deterministic. It explains how the initial pointed Kripke structure used in the DEL representation is obtained and discusses some extensions of m$\mathcal{A}^*$ that are inspired by work rooted in DEL approaches.

1 Introduction

The action language $m\mathcal{A}$ and its subsequent developments (e.g.,$m\mathcal{A}^*$) has been introduced for representing and reasoning about actions in dynamic multi-agent systems (e.g., (Baral et al. 2012; Baral et al. 2013; Baral et al. 2020)) in the spirit of action languages that were developed for representing and reasoning about actions in single-agent environment (Gelfond and Lifschitz 1998).

In $m\mathcal{A}^*$, a multi-agent domain is specified by a set of action descriptions and observability statements. The action descriptions capture the actions' preconditions and effects, while the observability statements encode who can (or cannot) fully or partially observe an action occurrence. The semantics of m$\mathcal{A}^*$ is defined by a transition function which specifies the possible pointed Kripke structures reached by the execution of an action by an agent or a group of agents. It has been utilized successfully in the development of epistemic planning systems (Fabiano et al. 2020; Le et al. 2018). Some basic properties of the transition function of m$\mathcal{A}^*$ have been discussed in (Baral et al. 2020). Thus far, uses of m$\mathcal{A}^*$ have been mostly demonstrated in domains without static causal laws (a.k.a. state constraints) and deterministic actions.

In this paper, we illustrate the use of m$\mathcal{A}^*$ *with* static causal laws and nondeterministic actions in formalizing a version of the well-known "*Three Player Card Game*" that has been often used to illustrate the application of Dynamic Epistemic Logic (DEL) in formalizing announcements or sensing actions. The game is interesting from the knowledge representation perspective in that it is rather simple but including nondeterministic actions and sensing with multiple outcomes. We show how conclusions derivable using DEL could be derived in an m$\mathcal{A}^*$ representation. We will also use this example to illustrate different features of m$\mathcal{A}^*$ that have not been the focus of research in reasoning about actions in multi-agent systems. In addition, we discuss the effects of static causal laws and nondeterministic actions on the transition function of m$\mathcal{A}^*$ and identify features that are useful for the development of the language.

2 Background: m$\mathcal{A}^*$

We briefly review the necessary definitions, most are presented in (Baral et al. 2020). A *multi-agent* domain $\langle \mathcal{AG}, \mathcal{F} \rangle$ includes a finite and non-empty set of agents $\mathcal{AG}$ and a set of fluents $\mathcal{F}$ encoding properties of the world. *Belief formulae* over $\langle \mathcal{AG}, \mathcal{F} \rangle$ are defined by the BNF:

$$\varphi \ ::= \ p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \mathbf{B}_i\varphi$$

where $p \in \mathcal{F}$ is a fluent and $i \in \mathcal{AG}$. We refer to a belief formula which does not contain any occurrence of $\mathbf{B}_i$ as *a fluent formula*. In addition, for a formula ψ and a nonempty set $\alpha \subseteq \mathcal{AG}$, $\mathbf{B}_\alpha\psi$ and $\mathbf{C}_\alpha\psi$ denote $\bigwedge_{i\in\alpha} \mathbf{B}_i\psi$ and $\bigwedge_{k=1}^{\infty} \mathbf{B}_\alpha^k\psi$, where $\mathbf{B}_\alpha^1\psi{=}\mathbf{B}_\alpha\psi$ and $\mathbf{B}_\alpha^{k+1}\psi{=}\mathbf{B}_\alpha^k\mathbf{B}_\alpha\psi$ for $k > 1$, respectively. $\mathcal{L}_{\mathcal{AG}}$ denotes the set of belief formulae over $\langle \mathcal{AG}, \mathcal{F} \rangle$.

Satisfaction of belief formulae is defined over *pointed Kripke structures* (Fagin et al. 1995). A Kripke structure M is a tuple $\langle S, \pi, \{\mathcal{B}_i\}_{i\in\mathcal{AG}} \rangle$, where S is a set of worlds (denoted by $M[S]$), $\pi : S \mapsto 2^{\mathcal{F}}$ is a function that associates an interpretation of $\mathcal{F}$ to each element of S (denoted by $M[\pi]$), and for $i \in \mathcal{AG}$, $\mathcal{B}_i \subseteq S \times S$ is a binary relation over S (denoted by $M[i]$). For convenience, we will often draw a Kripke structure M as a directed labeled graph, whose set of labeled nodes represent S and whose set of labeled edges contains $s \xrightarrow{i} t$ iff $(s, t) \in \mathcal{B}_i$. The label of each node has two parts: the name of the world followed by its interpretation. For $u \in S$ and a fluent formula φ, $M[\pi](u)$ and $M[\pi](u)(\varphi)$ denote the interpretation associated to u via π and the truth value of φ with respect to $M[\pi](u)$. For a world $s \in M[S]$, (M, s) is a *pointed Kripke structure*.

The satisfaction relation $\models$ between belief formulae and a pointed Kripke structure (M, s) is defined as follows:

1. $(M, s) \models p$ if p is a fluent and $M[\pi](s)(p)$ is true;
2. $(M, s) \models \mathbf{B}_i\varphi$ if $\forall t.[(s, t) \in \mathcal{B}_i \Rightarrow (M, t) \models \varphi]$;

3. $(M,s) \models \neg\varphi$ if $(M,s) \not\models \varphi$;
4. $(M,s) \models \varphi_1 \vee \varphi_2$ if $(M,s) \models \varphi_1$ or $(M,s) \models \varphi_2$;
5. $(M,s) \models \varphi_1 \wedge \varphi_2$ if $(M,s) \models \varphi_1$ and $(M,s) \models \varphi_2$.

We are now ready to review the basics of m$\mathcal{A}$*. In this language, an action theory over $\langle \mathcal{AG}, \mathcal{F} \rangle$ consists of a set of action instances $\mathcal{AI}$ of the form $a\langle\alpha\rangle$ representing that α, a set of agents, performs a, and a collection of statements of the following forms:

$$\textbf{executable\_if } \mathsf{a} \textbf{ if } \psi \quad (1)$$

$$\mathsf{a} \textbf{ causes } \ell \textbf{ if } \psi \quad (2)$$

$$\mathsf{a} \textbf{ determines } \varphi \quad (3)$$

$$\mathsf{a} \textbf{ announces } \varphi \quad (4)$$

$$z \textbf{ observes } \mathsf{a} \textbf{ if } \varphi \quad (5)$$

$$z \textbf{ aware\_of } \mathsf{a} \textbf{ if } \theta \quad (6)$$

$$e_1, \ldots, e_m \textbf{ if } p_1, \ldots, p_n \quad (7)$$

where ℓ, e_i, p_j are fluent literals, ψ is a belief formula, φ and θ are fluent formulae, $\mathsf{a} \in \mathcal{AI}$, and $z \in \mathcal{AG}$. (1) encodes the executability condition of a. ψ is referred as the *precondition of* a. (2) describes the effect of the ontic-action a, i.e., if ψ is true then ℓ will be true after the execution of a. (3) enables the agents, who execute a, to learn the value of the formula φ. (4) encodes an *announcement* action, whose owner announces that φ is true. Statements of the forms (5)–(6) encode the observability of agents given an occurrence of a. (5) indicates that agent z is a full observer of a if φ holds. (6) states that agent z is a partial observer of a if ψ holds. z, a, and φ (resp. ψ) are referred to as the observed agent, the action instance, and the condition of (5) (resp. (6)). It is assumed that the sets of ontic-actions, sensing actions, and announcement actions are pairwise disjoint. Furthermore, for every pair of a and z, if z and a occur in a statement of the form (5) then they do not occur in any statement of the form (6) and vice versa. (7) represents a *static causal law*; it conveys that whenever the fluent literals $p_1, \ldots, p_n$ hold in a state, then $e_1, \ldots, e_m$ will also hold in that state.

The semantics of m$\mathcal{A}^*$ is defined using the notion of *update models* (see, e.g., (Baltag and Moss 2004; van Benthem, van Eijck, and Kooi 2006)). This notion makes use of the notion of a $\mathcal{L}_{\mathcal{AG}}$-replacement, which[1] maps interpretations of $\mathcal{F}$ into sets of interpretations of $\mathcal{F}$. $REP_{\mathcal{L}_{\mathcal{AG}}}$ denotes the set of all $\mathcal{L}_{\mathcal{AG}}$-replacements. We denote with $\top$ the identity replacement, i.e., $\top(s) = \{s\}$ for every interpretation s. An update model $\mathbf{\Sigma}$ over $\mathcal{AG}$ is a tuple $\langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub \rangle$ where:

1. Σ is a set of *events*;
2. each R_i is a binary relation on Σ;
3. $pre : \Sigma \rightarrow \mathcal{L}_{\mathcal{AG}}$ is a function mapping each event $e \in \Sigma$ to a formula in $\mathcal{L}_{\mathcal{AG}}$; and
4. $sub : \Sigma \rightarrow REP_{\mathcal{L}_{\mathcal{AG}}}$ is a function mapping each event $e \in \Sigma$ to a replacement in $REP_{\mathcal{L}_{\mathcal{AG}}}$.

An *update instance* ω is a pair $(\mathbf{\Sigma}, e)$ where $\mathbf{\Sigma}$ is an update model $\langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub \rangle$ and e, referred to as a *designated event*, is a member of Σ. Again, for simplicity of the presentation, we often draw an update instance as a graph whose events are rectangles and whose links represent the accessibility relations between events with double lines representing designated events.

Given a Kripke structure M and an update model $\mathbf{\Sigma} = \langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub \rangle$, the *update* of M induced by $\mathbf{\Sigma}$, $M' = M \otimes \mathbf{\Sigma}$, is a Kripke structure defined by:

1. $M'[S] = \{(s, x, \tau) \mid s \in M[S], \tau \in \Sigma, x \in sub(\tau)(s)\}$;
2. $((s, x, \tau), (s', u, \tau')) \in M'[i]$ iff $(s, x, \tau), (s', u, \tau') \in M'[S]$ such that $(s, s') \in M[i]$ and $(\tau, \tau') \in R_i$;
3. $\forall((s, x, \tau) \in M'[S]).[M'[\pi]((s, x, \tau)) = x]$.

An *update template* is a pair $(\mathbf{\Sigma}, \Gamma)$, where $\mathbf{\Sigma}$ is an update model with the set of events Σ and $\Gamma \subseteq \Sigma$. The update of pointed Kripke structure (M, s) given an update template $(\mathbf{\Sigma}, \Gamma)$ is a set of pointed Kripke structures, denoted by $(M, s) \otimes (\mathbf{\Sigma}, \Gamma)$, where $(M, s) \otimes (\mathbf{\Sigma}, \Gamma) = \{(M \otimes \mathbf{\Sigma}, (s, \tau)) \mid \tau \in \Gamma, (M, s) \models pre(\tau)\}$.

The semantics of m$\mathcal{A}^*$ is defined by a transition function which maps pairs of action instances and *states* (pointed Kripke structures) into sets of states. It starts with the definitions of the frame of reference for the execution of an action instance a in a state (M, s), followed by the definition of the update template for the execution of a.

Given an action theory D, a state (M, s), and an action instance a, the *frame of reference* for the execution of a in (M, s) is a tuple $(F_D(\mathsf{a}, M, s), P_D(\mathsf{a}, M, s), O_D(\mathsf{a}, M, s))$ where

$$F_D(\mathsf{a}, M, s) = \{x \in \mathcal{AG} \mid [x \textbf{ observes } \mathsf{a} \textbf{ if } \varphi] \in D \text{ such that } (M, s) \models \varphi\}$$

$$P_D(\mathsf{a}, M, s) = \{x \in \mathcal{AG} \mid [x \textbf{ aware\_of } \mathsf{a} \textbf{ if } \varphi] \in D \text{ such that } (M, s) \models \varphi\}$$

$$O_D(\mathsf{a}, M, s) = \mathcal{AG} \setminus (F_D(\mathsf{a}, M, s) \cup P_D(\mathsf{a}, M, s))$$

Intuitively, $F_D(\mathsf{a}, M, s)$ (resp. $P_D(\mathsf{a}, M, s)$ and $O_D(\mathsf{a}, M, s)$) are the agents that are fully observant (resp. partially observant and oblivious) of the execution of a in the state (M, s). m$\mathcal{A}^*$ assumes that the sets $F_D(\mathsf{a}, M, s)$, $F_D(\mathsf{a}, M, s)$, and $P_D(\mathsf{a}, M, s)$ are pairwise disjoint.

To consider static causal laws, we need the following notion. Given a set of fluent literals X, let $Cn(X)$ be the minimal set of literals satisfying $X \subseteq Cn(X)$ and for every static causal law of the form (7), if $\{p_1, \ldots, p_n\} \subseteq Cn(X)$ then $\{e_1, \ldots, e_m\} \subseteq Cn(X)$. We say that an interpretation s of $\mathcal{F}$ is consistent if $Cn(s)$ is consistent. From now on, whenever we refer to an interpretation of $\mathcal{F}$, we assume that it is consistent. Furthermore, for an ontic action instance a and an interpretation s, let $e(a, s) = \{\ell \mid [a \textbf{ causes } \ell \textbf{ if } \psi] \in D, \psi$ is true in $s\}$. We define $r(a, s) = \{s' \mid s'$ is an interpretation of $\mathcal{F}$ and $s' = Cn(e(a, s) \cup (s \cap s'))\}$. Intuitively, $r(a, s)$ is the set of possible worlds resulting from the execution of a in s (see, e.g., (Gelfond and Lifschitz 1998) for a treatment of static causal laws in single agent domains).

Consider an instance of an ontic action a with a frame of reference $\rho = (F, \emptyset, O)$; the update model for a and ρ, denoted by $\omega(\mathsf{a}, \rho)$, is defined by $\langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub \rangle$ where

[1] We modified the definition of a substitution to accommodate the nondeterminism of ontic actions and call it a replacement to avoid confusion.

- $\Sigma = \{\sigma, \epsilon\}$;
- $R_i = \{(\sigma,\sigma),(\epsilon,\epsilon)\}$ for $i \in F$ and $R_i = \{(\sigma,\epsilon),(\epsilon,\epsilon)\}$ for $i \in O$;
- $pre(\sigma) = \psi$ and $pre(\epsilon) = \top$; and
- $sub(\epsilon) = \top$ and $sub(\sigma)(s) = r(\mathsf{a}, s)$.

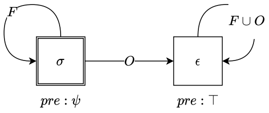


Figure 1: Update Template for an Ontic Action in m$\mathcal{A}^*$

The update template for the ontic-action occurrence a and the frame of reference ρ is $(\omega(\mathsf{a},\rho),\{\sigma\})$. Figure 1 shows the update model of an ontic action a.

Consider an instance of a truthful announcement (or sensing action) a that announces (senses) φ in a state (M,s) whose frame of reference is $\rho = (F,P,O)$. Assume that $\mathbf{executable\_if}$ a $\mathbf{if}$ ψ belongs to D. The update model for the occurrence of a in (M,s), denoted by $\omega(\mathsf{a},\rho)$, is defined by $\langle \Sigma, \{R_i\}_{i\in\mathcal{AG}}, pre, sub\rangle$ where:

- $\Sigma = \{\sigma,\tau,\epsilon\}$;
- R_i is given by
$$R_i = \begin{cases} \{(\sigma,\sigma),(\tau,\tau),(\epsilon,\epsilon)\} & \textit{if } i \in F \\ \{(\sigma,\sigma),(\tau,\tau),(\epsilon,\epsilon),(\sigma,\tau),(\tau,\sigma)\} & \textit{if } i \in P \\ \{(\sigma,\epsilon),(\tau,\epsilon),(\epsilon,\epsilon)\} & \textit{if } i \in O \end{cases}$$
- The preconditions pre are defined by
$$pre(x) = \begin{cases} \varphi \wedge \psi & \textit{if } x = \sigma \\ \neg\varphi \wedge \psi & \textit{if } x = \tau \\ \top & \textit{if } x = \epsilon \end{cases}$$
- $sub(x) = \top$ for each $x \in \Sigma$.

The update template for the announcement or sensing action occurrence a and the frame of reference ρ is $(\omega(\mathsf{a},\rho),\{\sigma\})$ or $(\omega(\mathsf{a},\rho),\{\sigma,\tau\})$, respectively. Figure 2 illustrates details of the update model for an announcement a that truthfully announces φ (or a sensing action a that determines φ).

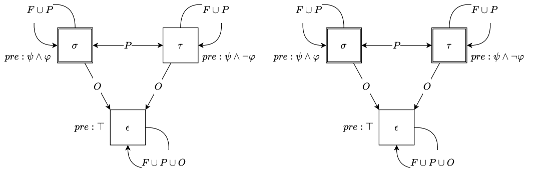


Figure 2: Update Model for Truthful Announcement and Sensing in m$\mathcal{A}^*$

Observe that the update model for a sensing action that senses φ is almost identical to that of an announcement action that announces φ. The only difference between them is the additional designated event in the update model of the sensing action.

Given a state (M,s) and an action a, the state reached after the occurrence of a in (M,s) is specified by $(M,s) \otimes (\omega(\mathsf{a},\rho),x)$ (x being the "real" event).

3 Three Player Card Game

We consider the "*Three Player Card Game*" as discussed in (Ditmarsch, van der Hoek, and Kooi 2007). The description of the game is as follows:

> *There are three agents in a room: Anne, Bill and Cath. There is a stack of three cards 0, 1 and 2 on the table. Each agent has to draw a card from the stack and this is all commonly known. First, Anne picks up her card and it is 0. Then Bill picks up his card and it is 1. Finally, Cath picks up her card and it is 2. Each agent just knows the value of their card but not others. After this, Anne can execute one of the two possible actions:*
>
> - Table*: Anne puts her card on the table to reveal its value to everyone.*
> - Show*: Anne shows Bill her card. Cath cannot see the value of Anne's card, but she notices that the card is being showed to Bill.*

The formalization of the above story in Dynamic Epistemic Logic from (Ditmarsch, van der Hoek, and Kooi 2007) is as follows. The action $pickup_X$ for player X would be represented as follows:

- $pickup_A \overset{def}{=} L_{ABC}(!L_A?0_A \cup L_A?1_A \cup L_A?2_A)$
- $pickup_B \overset{def}{=} L_{ABC}(L_B?0_B \cup !L_B?1_B \cup L_B?2_B)$
- $pickup_C \overset{def}{=} L_{ABC}(L_C?0_C \cup L_C?1_C \cup !L_C?2_C)$

where $?\varphi$ is a test, ! is a choice, $L_x?\varphi$ stands for 'the group of agents x learns φ'. Thus, the sentence $pickup_A \overset{def}{=} L_{ABC}(!L_A?0_A \cup L_A?1_A \cup L_A?2_A)$ is understood as all 3 players A, B, C (representing Anne, Bill and Cath, respectively) would learn that after A picked up her card, A would be able to tell which is the value of that card - and in fact A's card is 0. The other two sentences for $pickup_B$ and $pickup_C$ can be understood in the similar way. The transitions between states after the players drawn their card are illustrated in Figure 3 (from (Ditmarsch, van der Hoek, and Kooi 2007)). Each time a player picks up a card, that player would refine their belief into those worlds that satisfied the deal they got. The last epistemic state (last state in Figure 3) indicates that the players know their own card, and that other players also hold one card that differ from their own.

The execution of *table* and *show* by Anne can be represented as follows:

- $table \overset{def}{=} L_{ABC}?0_A$
- $show \overset{def}{=} L_{ABC}(!L_{AB}?0_A \cup L_{AB}?1_A \cup L_{AB}?2_A)$

where $table \overset{def}{=} L_{ABC}?0_A$ means that after action the execution of $table$ (Anne puts her card on table for everyone to see it), all players A, B and C would learn that the value of Anne's card is 0. Furthermore, $show \overset{def}{=} L_{ABC}(!L_{AB}?0_A \cup L_{AB}?1_A \cup L_{AB}?2_A)$ indicates the fact after Anne shows her card to Bill, all players A, B and C would learn that both A and B now know the value of Anne's card (and it is 0). Figure 4 describes the states after Anne executed $table$ or $show$. If Anne decided to perform $table$, then Bill and Cath not only know about Anne's card

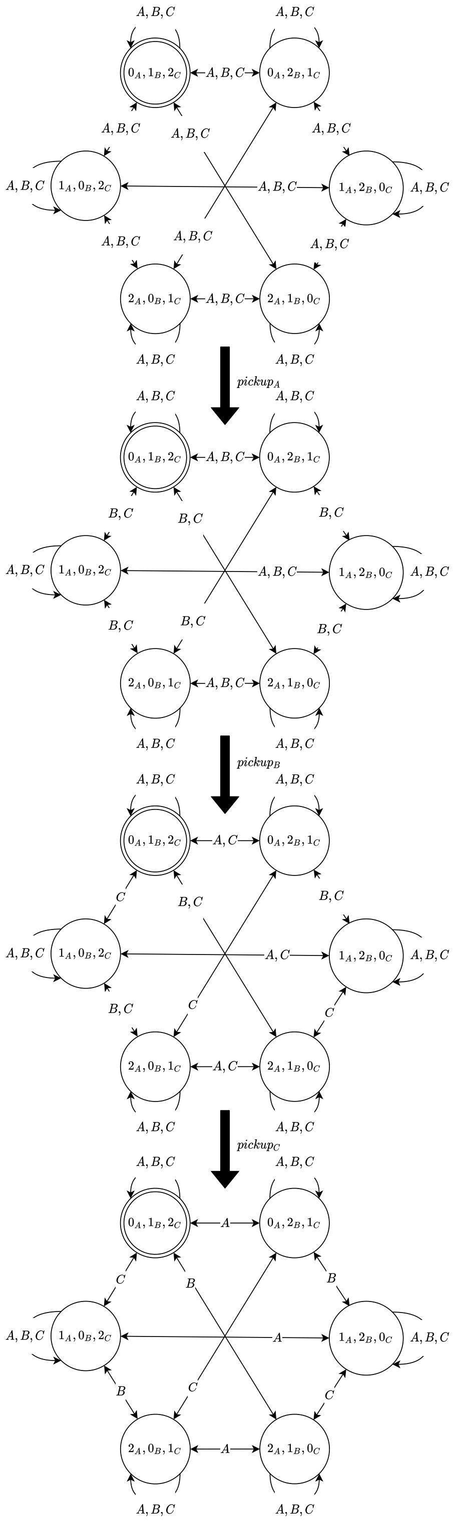


Figure 3: The epistemic states after each player picks up their cards.

but also can figure out what is the actual deal of the game at that point; leaving Anne is the only player left that does not know about the actual state of the world right now. If Anne *show* only Bill her card, then only Bill knows the actual state right now.

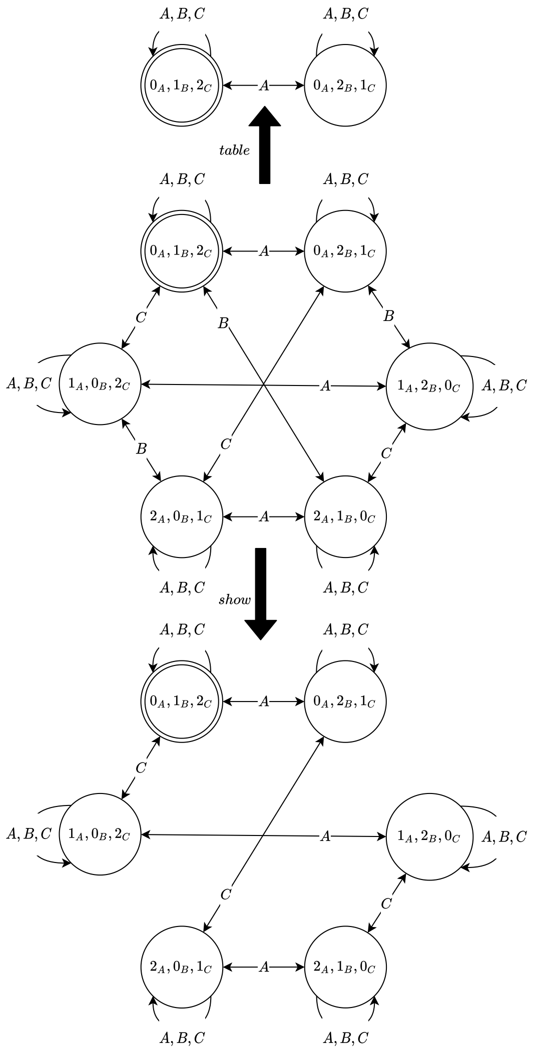


Figure 4: Execution of $table$ and $show$ by Anne

4 A m$\mathcal{A}^*$ Representation of the Card Game

Let us denote the multi-agent domain in *Three Player Card Game* by D_{card}. For this domain, we have that $\mathcal{AG} = \{A, B, C\}$ where A, B, and C represent Anne, Bill and Cath, respectively. The set of fluents $\mathcal{F}$ for this domain consists of:

- 0_x: agent x has card 0;

- 1_x: agent x has card 1; and
- 2_x: agent x has card 2.

where $x \in \{A, B, C\}$.

To describe the ontic action of drawing a card, we write

$$draw\langle x\rangle \ \textbf{causes}\ 0_x \vee 1_x \vee 2_x \qquad (8)$$

where $x \in \{A, B, C\}$. Observe that this action is nondeterministic. Since everyone can observe the action of drawing a card by anyone, the observability statement for this action is

$$y \ \textbf{observes}\ draw\langle x\rangle \qquad (9)$$

for $y, x \in \{A, B, C\}$.

The action of a player checking the card and determining the card that she holds is described by

$$check\langle x\rangle \ \textbf{determines}\ \{0_x, 1_x, 2_x\} \qquad (10)$$

where $x \in \{A, B, C\}$. Observe that this is an extension of the statement of the form (3) as it allows multiple outcomes (more than 2, which is the default number of outcomes in $m\mathcal{A}^*$). This says that checking the card allows the player knows exactly which card she is holding. The presence of multiple outcomes of a sensing action requires some changes in the update model of sensing action occurrences as elaborated in the next section. The observability statements for this action are

$$x \ \textbf{observes}\ check\langle x\rangle \qquad (11)$$

$$y \ \textbf{aware\_of}\ check\langle x\rangle \qquad (12)$$

where $x, y \in \{A, B, C\}$ and $x \neq y$. Observe that if x checks their card, only x is the full observer of the action and other agents are partial observers.

The action of someone tabling a card and its observability statements are described as follows

$$table(k)\langle x\rangle \ \textbf{announces}\ k_x \qquad (13)$$

$$\textbf{executable\_if}\ table(k)\langle x\rangle \ \textbf{if}\ k_x \qquad (14)$$

$$y \ \textbf{observes}\ table(k)\langle x\rangle \qquad (15)$$

where $k \in \{0, 1, 2\}$, $x, y \in \{A, B, C\}$, and $x \neq y$. Observe that the fact that a player can only show the card that they hold leads to the executability condition for this action.

Similarly, the action of someone showing their card to another player while not allowing the other player to see the card and its observability statements are described as follows

$$show(k, y)\langle x\rangle \ \textbf{announces}\ k_x \qquad (16)$$

$$\textbf{executable\_if}\ show(k, y)\langle x\rangle \ \textbf{if}\ k_x \qquad (17)$$

$$y \ \textbf{observes}\ show(k, y)\langle x\rangle \qquad (18)$$

$$z \ \textbf{aware\_of}\ show(k, y)\langle x\rangle \qquad (19)$$

where $k \in \{0, 1, 2\}$, $x, y \in \{A, B, C\}$, $x \neq y$, and $z \in \{A, B, C\} \setminus \{x, y\}$.

The fact that one can hold no card or exactly one card leads to the following static causal laws for this domain:

$$\neg p_x, \neg q_x \ \textbf{if}\ r_x \qquad (20)$$

where $x \in \{A, B, C\}$ and $\{p, q, r\} = \{0, 1, 2\}$. For example, for $x = A$ and $(p, q, r) = (1, 2, 0)$, it means that if 0_A is true then 1_A and 2_A must be false. Furthermore, if one holds a card then other cannot. This is represented as follows:

$$\neg r_y, \neg r_z \ \textbf{if}\ r_x \qquad (21)$$

where $\{x, y, z\} = \{A, B, C\}$ and $r \in \{0, 1, 2\}$. For example, for $r = 0$ and $(x, y, z) = (A, B, C)$, this says that if A holds the card 0 then neither B nor C holds it.

To complete the description of D_{card} we note that initially, none of the player holds any card. Therefore, the initial state is described by the following statements in $m\mathcal{A}^*$:

$$\textbf{initially}\ \mathbf{C}_{\{A,B,C\}}(\neg k_x) \qquad (22)$$

where $k \in \{0, 1, 2\}$ and $x \in \{A, B, C\}$.

Observe that the sentence $pickup_A$ in the DEL representation of (van Ditmarsch, van der Hoek, and Kooi 2007) could be considered as the sequence $draw\langle A\rangle; check\langle A\rangle$. The sequence $pickup_A; pickup_A; pickup_C$ can be summarized by the following sequence:

$$\alpha_{card} = \begin{cases} draw\langle A\rangle; check\langle A\rangle; \\ draw\langle B\rangle; check\langle B\rangle; \\ draw\langle C\rangle; check\langle C\rangle \end{cases} \qquad (23)$$

5 Extending m$\mathcal{A}^*$ to Accommodate Static Causal Laws and Nondeterministic Actions

As we have mentioned earlier, the nondeterministic action $draw\langle A\rangle$, whose execution in an actual world can result in different worlds, or the presence of a sensing action such as $check\langle A\rangle$ that allows A to determine which card she is holding, are slightly different from the standard sensing actions considered in m$\mathcal{A}^*$. Similarly, suppose that A holds the card 0 and tables it; this means that A has made a truthful announcement that 0_A is true. The presence of the static causal laws will allow B and C to derive the cards of the others.

Consider the action $draw\langle A\rangle$. The execution of this action from the world $s = \emptyset$ representing the interpretation[2] that assigns false to every fluent will result in one of the three worlds $s_1 = \{0_A\}$, $s_2 = \{1_A\}$, or $s_3 = \{2_A\}$. This is the result of the application of static causal laws (20) and (21) of the domain, i.e., $r(draw\langle A\rangle, s) = \{s_1, s_2, s_3\}$. The modification of the notions of replacement and of the update template of an ontic action, as described earlier, is therefore necessary to accommodate nondeterministic actions.

Assume that A draws the card 0 and then checks which card she is holding. Intuitively, this means that $check\langle A\rangle$ occurs in the real state of the world that A is holding 0. This will allow her to know that she is holding the card 0 and not the card 1 or 2. This means that the set of designated events for this action should consist of three elements, each represents a possible outcome and has the precondition of 0_A, 1_A, and 2_A, respectively. The update template for this action is adapted from the update template for sensing actions in m$\mathcal{A}^*$ as in Figure 5.

[2] We follow the convention of representing an interpretation as the set of fluents which are true in the interpretation.

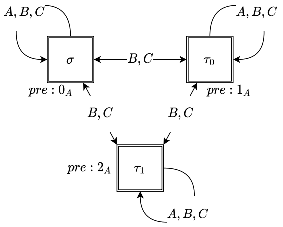


Figure 5: Update Template for $check\langle A\rangle$

In general, the update template of a sensing action that determines the value of a fluent belonging to a mutual exclusive set S of fluents will require $|S|$ designated events for the full observers and one event representing the oblivious agents. Every pair of designated events has a bidirectional edge labeled with the set of partially observant agents. Every designated event is connected to the event representing the oblivious agents. Furthermore, each event has a self-loop labeled with the set of all agents.

Similar extension needs to be done to accommodate the static causal laws for reasoning with announcement actions. We omit this discussion for brevity.

6 Reasoning in D_{card}

Let us now illustrate the execution of the sequence of actions α_{card} from (23). The initial state of the problem, given the initial state specification in (22), is a Kripke structure with a single world s_0, whose interpretation is $\emptyset$, and the self-loop with labels A, B, C (Figure 6).

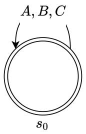


Figure 6: The initial state (M_0, s_0)

6.1 Anne Draws a Card and Checks

Suppose that A draws a card. Because of all agents are full observer of the actions, the update model of $draw\langle A\rangle$ will consist of a single event σ. The presence of the static causal laws (20) and (21) indicates that the replacement of σ is one that maps s_0 to $r(draw\langle A\rangle, s_0) = \{s_1, s_2, s_3\}$ with $s_1 = \{0_A\}$, $s_2 = \{1_A\}$, and $s_3 = \{2_A\}$. The update template of $draw\langle A\rangle$ is depicted in Figure 7. The execution of $draw\langle A\rangle$ in (M_0, s_0) results in multiple states (M_1, s_1), (M_1, s_2), and (M_1, s_3) and they are depicted together in Figure 8.

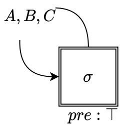


Figure 7: $(\omega(draw\langle A\rangle, (\{A, B, C\}, \emptyset, \emptyset)), \{\sigma\})$: Update Template of $draw\langle A\rangle$ in (M_0, s_0)

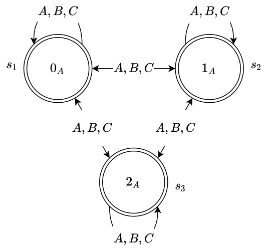


Figure 8: Execution of $draw\langle A\rangle$ in (M_0, s_0) results in (M_1, u) for $u \in \{s_1, s_2, s_3\}$

The execution of $check\langle A\rangle$, whose update template is given in Figure 5, in (M_1, s_1) results in (M_2, s_1) shown in Figure 9. It is easy to see that the execution of $check\langle A\rangle$ in (M_1, s_2) and (M_1, s_3) results in (M_2, s_2) and (M_2, s_3), respectively. M_2 differs from M_1 only in the links labeled A between s_i and s_j for $i \neq j$. We also reuse the world names whenever their interpretations in the new Kripke structure do not change. So s_1 in Figure 9 corresponds to (s_1, s_1, σ) in the definition of the cross product of Kripke structures and update model.

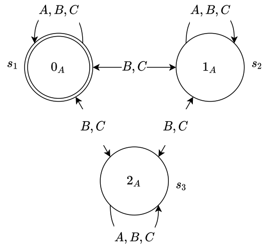


Figure 9: Execution of $check\langle A\rangle$ in (M_1, s_1) results in (M_2, s_1)

Assume that Anne draws card 0, the final state we get after A draws a card and checks her card is (M_2, s_1) (Figure 9). At this time, Anne knows exactly which card is in her hand (the card 0), she also knows that both Bill and Cath have empty hand. For Bill and Cath, they know that Anne has a card but do not know the value of that card.

6.2 Bill Draws a Card and Checks

After Anne draws her card (0), Bill draws his card from the desk. As with the case of Anne, all agents are full observer of this action, so the update model of $draw\langle B\rangle$ will have only one event σ. Because of the static causal laws (20) and (21), the replacement of σ is one that maps u_0 to $r(draw\langle B\rangle, u_0) = \{o_0, o_1\}$ with $o_0 = \{0_A, 1_B\}$ and $o_1 = \{0_A, 2_B\}$ (because A already has card 0). The update template of $draw\langle B\rangle$ is similar to Figure 7 that we omit to save space. The execution of $draw\langle B\rangle$ in (M_2, u_0) result in multiple state as represented in Figure 10.

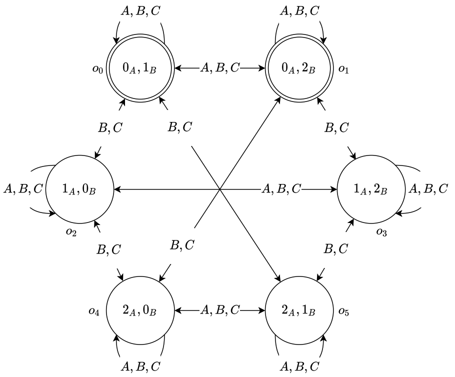


Figure 10: Execution of $draw\langle B\rangle$ in (M_2, u_0) results in (M_3, o) for $o \in \{o_0, o_1\}$

Then Bill checks his card. The update model of $check\langle B\rangle$ is showed in Figure 11 below:

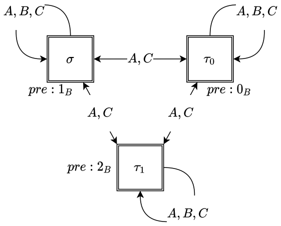


Figure 11: Update template for $check\langle B\rangle$

The execution of $check\langle B\rangle$ in (M_3, o_0) result in (M_4, o_0) (Figure 12). It is easy to see that the execution of $check\langle B\rangle$ in (M_3, o_1) results in (M_4, o_1). As in Anne's case, M_4 differs from M_3 only in the links labeled B between o_i and o_j for $i \neq j$.

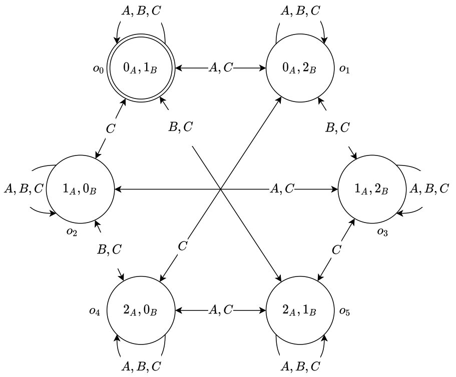


Figure 12: Execution of $check\langle B\rangle$ in (M_3, o_0) results in (M_4, o_0)

Because we assume that Bill draws card 1, the final state we get after B draws a card and checks his card is (M_4, t_0) (Figure 12). After these actions, all agents know that both Anne and Bill have a card in their hand; each of them knows the value of her/his card, but not other's. Cath's hand is still empty.

6.3 Cath Draws a Card and Checks

Finally, Cath takes the last card from the table. Similarly to previous cases, all agents are full observer of $draw\langle C\rangle$. The update template of $draw\langle C\rangle$ is showed in Figure 15.

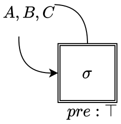


Figure 13: Update Template of $draw\langle C\rangle$ in (M_4, t_0)

The execution of $draw\langle C\rangle$ in (M_4, t_0) is depicted in Figure 14.

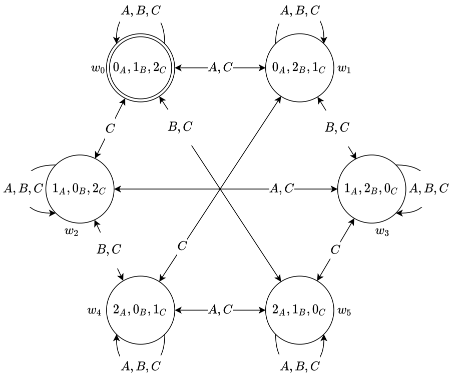


Figure 14: Execution of $draw\langle C\rangle$ in (M_4, t_0) results in (M_5, w_o)

Then Cath checks her card. The update model of $check\langle C\rangle$ is showed in Figure 15 below:

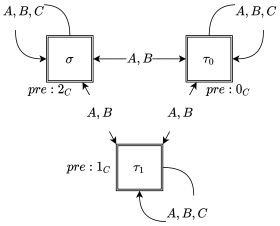


Figure 15: Update template for $check\langle C\rangle$

Figure 16 illustrates the execution of $check\langle C\rangle$ in (M_5, w_0).

For our discussion in the next part of this section, following (van Ditmarsch, van der Hoek, and Kooi 2007), let us

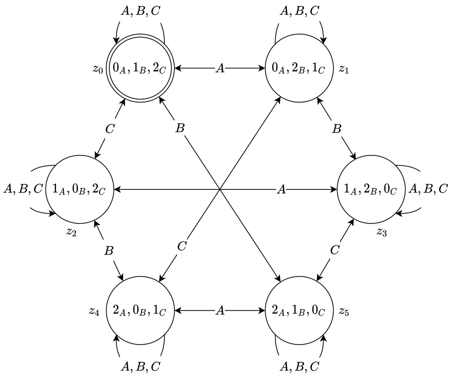


Figure 16: Execution of $check\langle C\rangle$ in (M_5, w_0) results in (M_6, z_0)

assume that A indeed holds the card 0, B holds the card 1, and C holds the card 2. Then, the state resulting from the sequence α_{card} on (M_0, s_0) is (M_6, z_0) in Figure 16. Observe that this is the state after $pickup_A; pickup_B; pickup_C$ used in Figure 3.

6.4 Anne Puts her Card on the Table

Let us consider the action of A putting her card on the table, i.e., $table(0)\langle A\rangle$. This is a truthful announcement action that announces to all players the value of Anne's card. Since everyone is a fully observer of this actions, the update template $(\omega(table(0)\langle A\rangle, (\{A, B, C\}, \emptyset, \emptyset)), \{\sigma\})$ consists of only one event σ. The update template and the result of its occurrence in (M_6, z_0) are given in Figure 17.

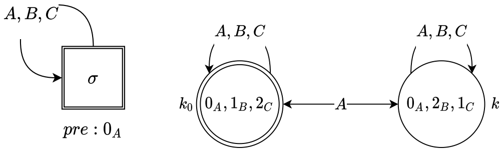


Figure 17: Update template for $table(0)\langle A\rangle$ and the result of execution of $table(0)\langle A\rangle$ in (M_6, z_0)

Afterwards, all three players know Anne holds the card 0. Moreover, Bill and Cath also discover the actual world (who has what card), while Anne does not realize what is Bill's card and what is Cath's card. This is also the result obtained in (van Ditmarsch, van der Hoek, and Kooi 2007) (Figure 4).

6.5 Anne Privately Shows Bill her Card

Suppose that A privately shows B her card, i.e., the action $show(0, B)\langle A\rangle$ occurs. Because of (18) and (19), A and B are full observer while C is a partial observer of the occurrence of the action. This allows that the update template for $show(0, B)\langle A\rangle$ is as in Figure 18.

The execution of $show(0, B)\langle A\rangle$ in (M_6, z_0) results in Figure 19.

The above state shows that only Bill recognizes the true world; Anne and Cath still do not know that but they are aware that Bill is the only one know about the actual world. Again, this is also the result obtained in (van Ditmarsch, van der Hoek, and Kooi 2007) (Figure 4).

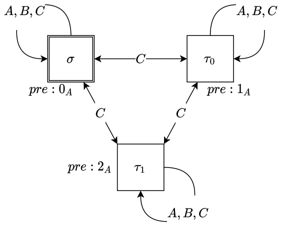


Figure 18: Update template of action $show(0, B)\langle A\rangle$

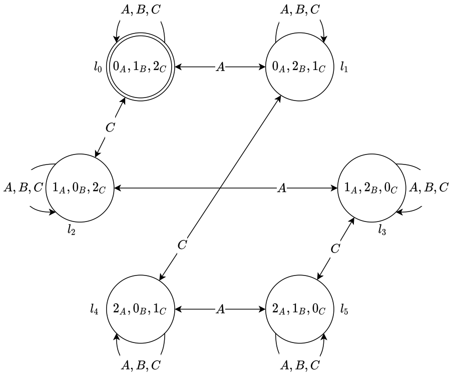


Figure 19: The result (M_8, l_0) of action $show(0, B)\langle A\rangle$ in (M_6, z_0)

7 Discussion

The above section illustrates the use of m$\mathcal{A}^*$ in representing a domain with static causal laws and nondeterministic actions. In this section, we discuss other aspects of reasoning about actions and change in multi-agent domains that have been considered in the literature but not in m$\mathcal{A}^*$. First, we discuss the *whisper* action that is also discussed in (van Ditmarsch, van der Hoek, and Kooi 2007). B asks A what card does she have and A responds that it is not 2. C knows that B asking A but cannot see and hear. This action differs from *table* or *show* in C's observability. While C anticipates an answer from A to B, different options are possible. Suppose that A whisper to B that she does not have card 2. To be consistent with the notation used in previous sections, we write $whisper(2, B)\langle A\rangle$ to denote this action. Following the approach used in m$\mathcal{A}^*$, A and B are full observer while the observability of C is not as clear as her observability with respect to the *table* or *show* action occurrences. C could imagine that $whisper(2, B)\langle A\rangle$ occurs and is considered as a partial observer of the action occurrence. In addition, the anticipation of C also indicates that different actions could have occurred, e.g., A tells B that she does not have the card 1, or A tells B that she has the card 0. We take the liberty

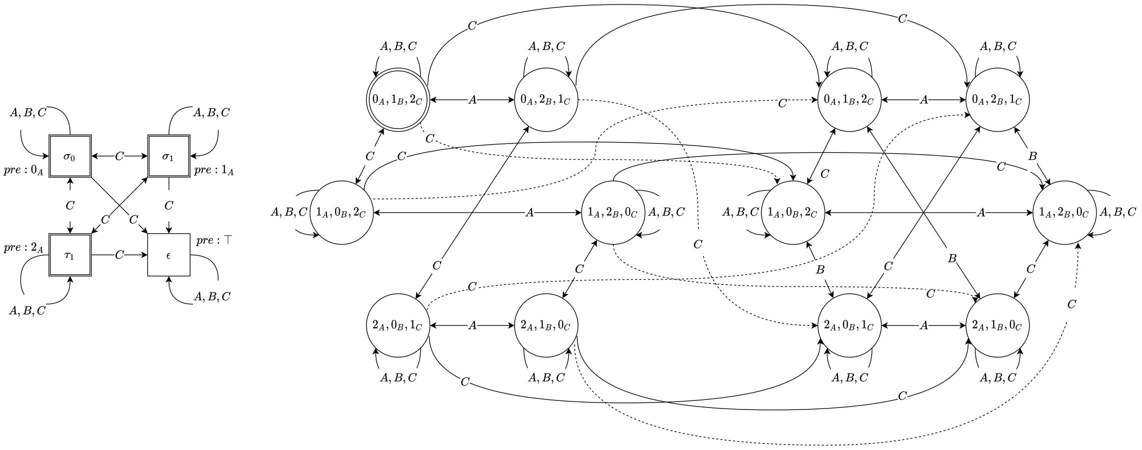


Figure 20: Update template and the result (M_9, m_0) of action $whisper(2, B)\langle A \rangle$ in (M_6, z_0)

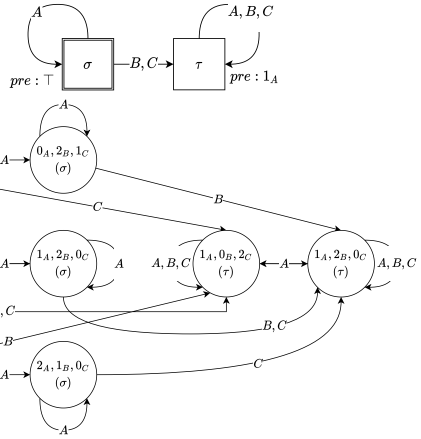
of assuming that C imagines that either $whisper(2, B)\langle A \rangle$ or $whisper(1, B)\langle A \rangle$ occurs and create the update template for this view (Figure 20, left). The state resulting from applying this update template on (M_6, z_0) is given in Figure 20 (right). We note that this result is different from the outcome presented in (van Ditmarsch, van der Hoek, and Kooi 2007) (the dashed links). It shows that only B recognizes the true world. A does not know that but realizes that B is the only one who knows the actual world.

We will now focus on *false announcements* as discuss in (van Ditmarsch 2014; van Ditmarsch et al. 2012). In this work, false announcements are treated as public announcement, i.e., all agents are fully observant of the action occurrence. Let us consider the action "*A tells to B and C that she has card 1*". We note that this announcement satisfies both assumptions in (van Ditmarsch et al. 2012). There are two different scenarios here:

- Both B and C could believe in A's announcement regardless of their belief before the announcement; or
- One of them would detect that A was lying because she/he has the card that A announces (in this case this is B).

In the first scenario, the update model and the result state of this announcement, following (van Ditmarsch et al. 2012), are described in Figure 21 (top and bottom, respectively). After A's announcement, C now believes that the actual deal is $\{1_A, 0_B, 2_C\}$ (which is not the true state), and B does not know what to believe in anymore. This is certainly not a realistic scenario.

For the second scenario, the update model and the result state of this announcement are described in Figure 22 (top and bottom, respectively). Because B is holding card 1, so he must know that A is lying to everyone and should not believe in the lie. In the result, A also knows that B discover that she was lying. Both A and B also know that C believes in the announcement and think the real state is $\{1_A, 0_B, 2_C\}$ (which is wrong).



A
σ
B, C
τ
A, B, C
pre : $\top$
pre : 1_A
$0_A, 1_B, 2_C$ (σ)
$0_A, 2_B, 1_C$ (σ)
$1_A, 0_B, 2_C$ (σ)
$1_A, 2_B, 0_C$ (σ)
$1_A, 0_B, 2_C$ (τ)
$1_A, 2_B, 0_C$ (τ)
$2_A, 0_B, 1_C$ (σ)
$2_A, 1_B, 0_C$ (σ)
B
C

Figure 21: A lies that she has the card 1 and everyone believes her

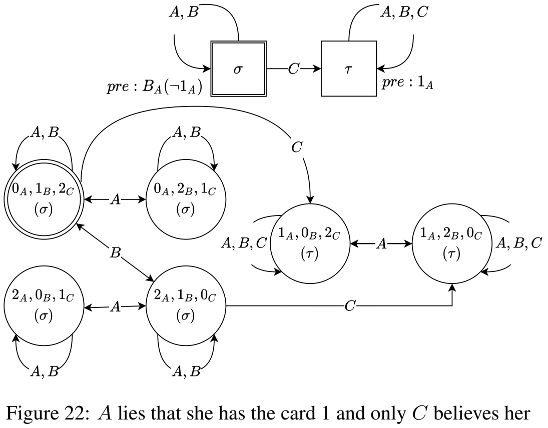
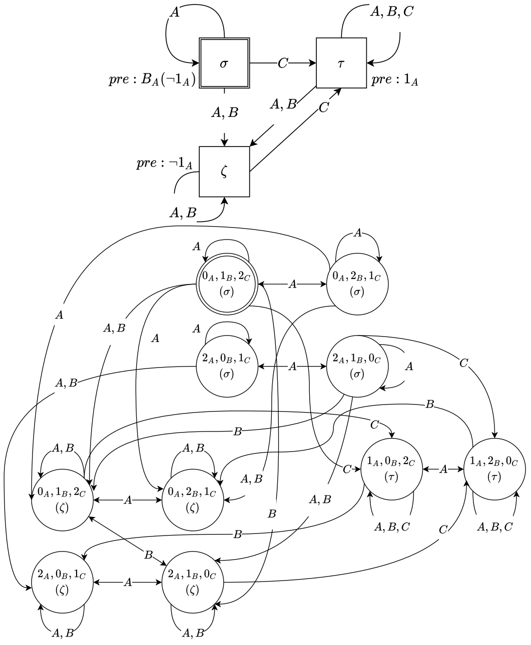


Figure 22: A lies that she has the card 1 and only C believes her

In our view, the construction of the update template in the second scenario, as proposed in (van Ditmarsch et al. 2012), could have an alternative that we present in Figure 23. The update model has a third event, ζ, encoding the fact that there are agents who do not believe in the lie because they know the truth, which are A and B. They know the actual event (σ) but believe in ζ, and hence, the accessibility relation from σ and τ to ζ. They also know that those who believes in the lie will not believe in this event and hence the link from ζ to τ. The resulting state of applying this update template on (M_6, z_0) (Figure 23) is significantly more complicated than that in Figure 22. Observe that in both states, A and B do not know the actual world while C has the wrong belief; A realizes that B knows that she is lying; etc. On the other hand, the state in Figure 22 allows us to conclude $B_C B_B 1_A$ but this is not the case in the state in Figure 23. This reflects the difference between the two views: in Figure 22, C believes that B also believes in the lie while in Figure 23, C does not. In our view, both are reasonable and worth considering.



A
A, B, C
σ
C
τ
$pre : B_A(\neg 1_A)$
$pre : 1_A$
A, B
A, B
C
$pre : \neg 1_A$
ζ
A, B
$0_A, 1_B, 2_C$ (σ)
$0_A, 2_B, 1_C$ (σ)
$2_A, 0_B, 1_C$ (σ)
$2_A, 1_B, 0_C$ (σ)
$1_A, 0_B, 2_C$ (τ)
$1_A, 2_B, 0_C$ (τ)
$0_A, 1_B, 2_C$ (ζ)
$0_A, 2_B, 1_C$ (ζ)
$2_A, 0_B, 1_C$ (ζ)
$2_A, 1_B, 0_C$ (ζ)
A
A, B
B
C
A, B, C

Figure 23: Alternative to "A lies that she has the card 1 and only C believes her"

8 Conclusions

In this paper, we present a formalization of a popular example "*Three Player Card Game*" in the action language m$\mathcal{A}^*$. We discuss an extension of the current definitions in m$\mathcal{A}^*$ that accommodates static causal laws and nondeterministic actions and allows us to reach the same conclusions as in DEL-based approaches when observabilities of agents are deterministic (e.g., in the *table* and *show* actions). The discussion also illustrates how the initial pointed Kripke structure of the problem is obtained. We also explore various features that have been discussed in the literature and identify possible extensions of m$\mathcal{A}^*$ such as how to incorporate actions with nondeterministic observabilities (e.g., the *whisper* action) or lying announcements. We leave this for future work on m$\mathcal{A}^*$.

References

Baltag, A., and Moss, L. 2004. Logics for epistemic programs. *Synthese*.

Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2012. An action language for reasoning about beliefs in multi-agent domains. In *Proceedings of NMR*.

Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2013. Reasoning about the beliefs of agents in multi-agent domains in the presence of state constraints: The action language mal. In Leite, J.; Son, T. C.; Torroni, P.; van deer Torre, L.; and Woltran, S., eds., *Proceedings of the 14th International Workshop, Computational Logic in Multi-Agent Systems, CLIMA VIX, Coruna, Spain, September 16-18, 2013*, volume 8143 of *Lecture Notes in Computer Science*, 290–306. Springer.

Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2020. An Action Language for Mutli-Agent Domains: Foundations. *arXiv.org* https://arxiv.org/abs/1511.01960v3.

Ditmarsch, H. v.; van der Hoek, W.; and Kooi, B. 2007. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition.

Fabiano, F.; Burigana, A.; Dovier, A.; and Pontelli, E. 2020. Efp 2.0: A multi-agent epistemic solver with multiple e-state representations. *Proceedings of the International Conference on Automated Planning and Scheduling* 30(1):101–109.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 1995. *Reasoning about Knowledge*. MIT press.

Gelfond, M., and Lifschitz, V. 1998. Action Languages. *Electronic Transactions on Artificial Intelligence* 3(6).

Le, T.; Fabiano, F.; Son, T. C.; and Pontelli, E. 2018. EFP and PG-EFP: Epistemic Forward Search Planners in Multi-Agent Domains. In *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press.

van Benthem, J.; van Eijck, J.; and Kooi, B. P. 2006. Logics of communication and change. *Inf. Comput.* 204(11):1620–1662.

van Ditmarsch, H.; van Eijck, J.; Sietsma, F.; and Wang, Y. 2012. On the logic of lying. In van Eijck, J., and Verbrugge, R., eds., *Games, Actions and Social Software - Multidisciplinary Aspects*, volume 7010 of *Lecture Notes in Computer Science*. Springer. 41–72.

van Ditmarsch, H.; van der Hoek, W.; and Kooi, B. 2007. *Dynamic Epistemic Logic*. Springer.

van Ditmarsch, H. 2014. Dynamics of lying. *Synth.* 191(5):745–777.

New Experiments on Reinstatement and Gradual Acceptability of Arguments

Elfia Bezou Vrakatseli[1] , **Henry Prakken**[1,2] , **Christian P. Janssen**[3]
[1]Department of Information and Computing Sciences, Utrecht University, The Netherlands
[2]Faculty of Law, University of Groningen, The Netherlands
[3]Experimental Psychology and Helmholtz Institute, Utrecht University, The Netherlands.
e.bezouvrakatseli@students.uu.nl, h.prakken@uu.nl, c.p.janssen@uu.nl

Abstract

This paper investigates whether empirical findings on how humans evaluate arguments in reinstatement cases support the 'fewer attackers is better' principle, incorporated in many current gradual notions of argument acceptability. Through three variations of an experiment, we find that (1) earlier findings that reinstated arguments are rated lower than when presented alone are replicated, (2) ratings at the reinstated stage are similar if all arguments are presented at once, compared to sequentially, and (3) ratings are overall higher if participants are provided with the relevant theory, while still instantiating imperfect reinstatement. We conclude that these findings could at best support a more specific principle 'being unattacked is better than attacked', but alternative explanations cannot yet be ruled out. More generally, we highlight the danger that experimenters in reasoning experiments interpret examples differently from humans. Finally, we argue that more justification is needed on why, and how, empirical findings on how humans argue can be relevant for normative models of argumentation.

1 Introduction

Rahwan et al. (2010) presented an empirical study of how people evaluate arguments in the context of counterarguments. Their aim was to assess how the abstract argumentation semantics of Dung (1995) treat so-called reinstatement patterns, in which an argument that is attacked by another argument is defended or 'reinstated' by an argument attacking the attacker, so that if there are no further arguments, the first and third argument are acceptable but the second argument must be rejected. They found that people by-and-large assess arguments according to Dung's semantics but not fully: on a 7-point scale, the first argument was rated significantly more acceptable when presented on its own than when presented together with its attacker and defender.

There are several reasons to reconsider these experiments. A general reason is that it has been claimed that the psychological sciences face a 'replicability crisis' since the results of many well-known experiments appear not to be replicable (Pashler and Wagenmakers, 2012). In light of this, one aim of this paper is to test whether the results of Rahwan et al. (2010) can be replicated. A more specific reason is that since the study of Rahwan et al. appeared, the study of gradual notions of argument acceptability has become popular. These studies include probabilistic (Hunter and Thimm, 2017), graded (Grossi and Modgil, 2019), and ranking-based (Amgoud and Ben-Naim, 2013) approaches. Some of this work refers to Rahwan et al.'s study for support of their approaches, either for gradual notions of acceptability in general (Polberg and Hunter, 2018; Hunter, Polberg, and Thimm, 2020) or for specific features of the new semantics (Grossi and Modgil, 2015, 2019; Amgoud, 2019).

In particular, Grossi and Modgil (2015) cite Rahwan et al. in support for a principle that everything else being equal, having fewer attackers is better. This principle is also a key element in several of the new semantics. For instance, all six ranking-based semantics studied by Bonzon et al. (2016) satisfy the principle of 'void precedence' (Amgoud and Ben-Naim, 2013), according to which an argument that has no attackers is more acceptable than an argument that has attackers, even if these attackers are counterattacked.

Accordingly, another aim of this paper is to investigate whether Rahwan et al.'s study indeed provides support for these recent developments, in particular for the 'fewer attackers is better' or 'void precedence' principle. In doing so, we will regard these formalisms not as descriptive but as prescriptive, or normative models of argumentation, that is, as modeling how people *should argue*. Our investigations are in part motivated by discussions of Cramer and Guillaume (2018a,b) and Prakken and de Winter (2018) of Rahwan et al.'s study, which give reasons to be cautious when referring to Rahwan et al. in support of the new semantics, suggesting alternative explanations for Rahwan et al.'s findings. In doing so, we do not aim to question the importance of graduality in argumentation as such. We take it for granted that graduality plays important roles in argument evaluation; the question that concerns us is how these roles can best be modelled. Moreover, we would also like to note that not all graduality semantics regard the void precedence principle as generally acceptable; for example, Bonzon et al. (2021) and Thimm and Kern-Isberner (2014) independently challenge the principle for separate reasons.

In this paper we report on three experiments in which humans evaluate arguments. The first experiment succeeded in replicating Rahwan et al.'s results on imperfect recovery from attack. The other two were aimed to test two versions of an alternative explanation for Rahwan et al.'s results suggested by Rahwan et al. and Prakken and de Winter (2018), namely, that the imperfect recovery of arguments from attack is not because the participants in the experiments applied the 'having fewer attackers is better' principle when rating the arguments, but it is due to the specific way in which the arguments were presented to them. These experiments yielded mixed results. We evaluate the results of our experiments in light of the above-mentioned literature but also in light of the question whether empirical studies have anything to say at all about the assessment of normative theories of argumentation. Our main conclusion will be that Rahwan et al. (2010)'s results cannot (yet) be considered supporting evidence for the idea that all other things being equal, having fewer attackers is better, as embodied in the 'void precedence' principle, since alternative explanations for the effect they found cannot be ruled out and since a more convincing explanation is needed for why empirical findings are relevant for normative theories of argumentation.

2 Preliminaries

In this section the basics of Dung's theory of abstract argumentation frameworks are summarised and applied to the reinstatement pattern that was the subject of the studies of Rahwan et al. (2010). We present Dung's semantics in a labelling version, which is equivalent to Dung's original semantics (Jakobovits, 2000; Caminada, 2006).

An *abstract argumentation framework* (*AF*) is a pair $\langle \mathcal{A}, \mathcal{C} \rangle$, where $\mathcal{A}$ is a set of *arguments* and $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation of *attack*. The labelling approach characterises the various semantics in terms of labellings of $\mathcal{A}$.

A *labelling* of an abstract argumentation framework $\langle \mathcal{A}, \mathcal{C} \rangle$ is any assignment of either the label *in* or *out* (but not both) to zero or more arguments from $\mathcal{A}$ such that:

1. an argument is *in* iff all arguments attacking it are *out*.
2. an argument is *out* iff it is attacked by an argument that is *in*.

Then *stable semantics* labels all arguments, while *grounded semantics* minimises and *preferred semantics* maximises the set of arguments that are labelled *in*. Relative to a semantics, an argument is *sceptically acceptable* if it is labelled *in* in all labellings, it is *rejected* if it is labelled *out* in all labellings, and it is *credulously acceptable* if it is labelled *in* in some but not all labellings.

The reinstatement pattern studied by Rahwan et al. is displayed in Figure 1. In both AFs argument A is sceptically acceptable in all three semantics. With only A this is trivial since A has no attackers. When also B and C are present, C has to be made *in* by constraint (1), since it has no attackers, and B has to be made *out* by constraint (2), thus A has to be made *in* by constraint (1). Thus C reinstates A by defending A against B.

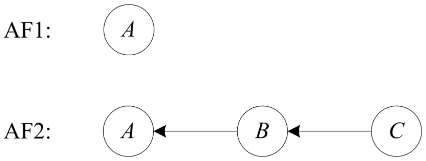


Figure 1: The reinstatement pattern

This outcome for $AF2$ is the same if the attack from B on A is made symmetric but it changes if the attack from C on B is made symmetric (regardless whether the same is done for B's attack on A). If C and B attack each other then the just-given labelling is still possible but it is not the only one: a labelling in which B is *in* and both A and C are *out* also satisfies the constraints. Both of these labellings are preferred and stable but not grounded, since the empty labelling also satisfies the constraints. Thus all three arguments are credulously acceptable in preferred and stable semantics while they are not acceptable in grounded semantics.

Rahwan et al. presented six examples to the participants in their experiments, all having the same pattern and all assumed to instantiate $AF2$ from Figure 1. The participants were first confronted with a single argument, for instance:

A: *The battery of Alex's car is not working. Therefore, Alex's car will halt.*

They were then asked to rate their confidence in its conclusion. Only then were they subsequently confronted with an attacker and defender, for instance:

B: *The battery of Alex's car has just been changed today. Therefore, the battery of Alex's car is working.*

C: *The garage was closed today. Therefore, the battery of Alex's car has not been changed today.*

After both arguments, the participants were again asked to rate their confidence in the conclusion of the initial argument. After argument B their average rating of A's conclusion went down while after argument C was presented to them, their average rating went up again, but to a significantly lower level than after being presented with A only. Rahwan et al. concluded that their results support the notion of reinstatement but not fully, since a reinstatement argument does not fully recover from an attack.

One explanation Rahwan et al. consider for their result is in terms of an effect of 'suspension of disbelief', according to which participants are capable of thinking of different kinds of objections to the presented arguments but they suspend these objections for the sake of the experiment. However, when one objection is presented by the experimenter, this suspension is disrupted and some participants start to let their private beliefs 'leak' into their assessments of the arguments. Prakken and de Winter (2018) suggest a variation of this explanation, advocating that after being introduced to an attacker, a participant's degree of belief in other possible attackers increases as well since the very introduction of an attacker leads them to consider other possible objections.

3 The Experiments

We conducted three experiments to further test these ideas. The methods of the experiments overlap and are presented together for brevity. Experiment 1 is an online replication of the study by Rahwan et al. (2010). Specifically, we test whether rating is lower at the reinstated stage compared to the base case when arguments are presented one-by-one (cf. Rahwan et al.). Based on this replication, we then test ideas proposed by Rahwan et al. (2010) and Prakken and de Winter (2018). Specifically, experiment 2 tests whether the rating is different if all arguments (including the attack and the defense) are presented at once. Finally, experiment 3 tests what happens if first all possible scenarios are presented — i.e., generalised forms of the arguments the participants encounter during evaluation — and then the arguments are presented one-by-one. As an example of (3), the generalised form of the car battery example was

- *A car will halt if its battery is not working.*
- *A car's battery is working if it has been changed the same day.*
- *When the garage is closed, a car's battery cannot be changed.*

3.1 Hypotheses

We tested the following four hypotheses.

> **Hypothesis 1:** When arguments are presented sequentially (experiment 1), participants' ratings for the conclusion of argument A in the reinstated stage are lower than in the base stage but higher than after argument B is presented.

The first hypothesis merely predicts a successful replication of Rahwan et al.'s results. Note that our participant number (130 aimed) is significantly higher than that used by Rahwan (20), to gain further confidence in the result.

> **Hypothesis 2:** When all arguments are presented at once (experiment 2), participants' ratings for the conclusion of argument A are higher than the (corresponding) ratings in the reinstated stage of the first case/manner-of-presentation (where all arguments are also available but have been introduced sequentially).

The second hypothesis suggests that when all the information is presented at the same time to the participants, the confidence in the conclusion of argument A is higher than the corresponding confidence in the reinstated stage when arguments have been presented one-by-one. Since the introduction of an attacker may change the participant's belief in the initially presented argument even after it has been reinstated, it is possible that it is the very gradual process of presentation that influences the participant's degree of belief. To quote Rahwan et al., "[p]articipants can easily generate all sorts of objections to the arguments presented to them by the experimenter, but they suspend their disbelief in these arguments for the sake of the experiment. When one objection is presented by the experimenter herself, though, suspension of disbelief is disrupted". Thus, if we eliminate the gradual factor of presentation, the initial suspension of disbelief may remain, since there is no stage where a *new* objection is presented that can disrupt it.

Possibly, when an attacker is introduced after one has placed their confidence in an argument, a kind of 'breach of confidence' is generated, one that cannot be later eradicated (by introducing another attacker) and that has caused the disruption of the initial experiment's 'convention/contract' (i.e., the suspension of disbelief). Hence, if all arguments were presented at once, they could all be considered as the aforementioned 'arguments presented by the experimenter' and participants would suspend their disbeliefs for all of them (as suggested). Provided with all the information (i.e., all the arguments in play) at the beginning, participants can make a deliberation without the element of surprise, resulting in giving the conclusion of argument A a higher confidence rating than in the reinstated stage of a gradual presentation.

> **Hypotheses 3a+3b:** When participants are first presented with all possible scenarios (experiment 3) — i.e., when they are presented with generalised forms of the arguments they will encounter during evaluation, before evaluating them — and are then asked to evaluate the arguments one-by-one (the same way as in experiment 1):
>
> a their ratings for the conclusion of argument A in the reinstated stage are higher than the corresponding ratings in the reinstated stage of the first experiment (where participants have not seen all the possible scenarios beforehand);
>
> b their ratings for the conclusion of argument A in the base stage are lower than the corresponding ratings in the base stage of the first experiment.

In our statistical test, we ran an Analysis Of Variance (ANOVA) with experiment (experiment 1 or 3) as between-subjects factor, and moment (base stage versus reinstated stage) as within subjects factor. Based on the hypotheses above, we would expect a significant interaction effect: rating is lower in the reinstated stage for participants in experiment 1 (compared to its base stage), whereas this is not the case for experiment 3 (i.e., no imperfect reinstatement is expected in experiment 3).

To further comment on hypotheses 3a and 3b, and extending on our thinking concerning the second hypothesis, we ought to consider another possible explanation and, thus, another manner of presentation. When a participant initially evaluates an argument, no evidence for or against its premises, inference, or conclusion has been offered, whereas after being presented with the attacker and defender, further evidence is overall provided, allowing the subject to form a more complete image of a precise situation.

Hypotheses 3a and 3b are based on Prakken and de Winter (2018), who argue that the introduction of an attacker increases the participants' degree of belief in other possible attackers, which are not explicitly ruled out in the presented arguments. They suggest that the introduction of a relevant theory prior to participants' evaluations will cause the confidence degree in the conclusion of argument A in the base stage to decrease (compared to ratings from the first manner of presentation) and to increase in the reinstated stage. Their suggestion is based on the assumption that if a participant was aware from the beginning of (all) the reasons argument A can be vulnerable, their belief in the possibility of the attacker that is presented (here, argument B) would increase from the base stage, resulting in a lower rating for the conclusion A at that stage. By the same logic, their degree of belief in all other attackers, which are not ruled out (but neither presented) in the experiment, would have no reason to increase after the actual introduction of the attacker in the defeated stage (contrarily to when one is not initially introduced to the whole theory) and, thus, confidence in argument A's conclusion would increase in the reinstated stage.

A confirmation of hypotheses 2, 3a and 3b would underline the importance of the way in which subjects are presented with arguments, proving it affects participants' confidence. Such confirmations would support the observations of Rahwan et al. and Prakken and de Winter (2018) on the possible effects of suspension of disbelief, as, then, said findings could be interpreted as a result of the two aforementioned suggested explanations and not as support for graded notions of argument acceptability.

3.2 Method

We conducted three experiments. In all three experiments, participants had to evaluate the acceptability status of natural language arguments, in which we followed Rahwan et al. (2010)'s method as closely as possible in terms of materials, procedure and measurement, discussed in more detail below.

Participants In each experiment, 130 participants took part (390 total). All were 18-65 years old. The average age was comparable between experiments (mean age for experiment 1, 2, and 3 respectively: 30, 33, and 28 years of age). All participants were volunteers, recruited through personal contact, and had no pre-knowledge on the topic of study. Participants were required to be over 18 years of age, and able to read and speak English, for which we probed participants at the start of the survey. All participants were deemed suitable according to their responses.

Materials The materials followed original stimuli of Rahwan et al. as close as possible. In each experiment, participants had to rate eight sets of arguments, consisting of three arguments each, where the conclusion of each next argument contradicts a premise of the preceding argument. The first six sets were taken from Rahwan et al. while the two remaining sets were added by us in a similar style. Specifically, these were:

A: *The power is out, so Claire cannot charge her phone.*

B: *The TV is playing, so the power is not out.*

C: *The TV is broken, so the TV is not playing.*

and

A: *Animals have the right to be left unharmed, so we should ban animal testing.*

B: *Animals are very dissimilar to humans, so animals do not have such a right.*

C: *Animals resemble us anatomically, physiologically, and behaviourally (e.g., recoiling from pain, fearing tormentors), therefore they are not very dissimilar to humans.*

At various points (see design), participants had to rate the acceptability of the conclusion of argument A. The ratings were given on a 7-point scale ranging from *Certainly false* to *Certainly true* as in Rahwan et al. (2010).

Design In experiment 1, we replicate Rahwan et al. (2010). Arguments A, B, and C were added in sequence. After each added statement, participants had to rate the acceptability of the conclusion of argument A. Consistent with hypothesis 1 and Rahwan et al. (2010), we expect ratings to be higher after presentation of argument A (base stage) compared to after presentation of argument C (reinstated stage). This is tested with a paired t-test.

In experiment 2, all arguments are presented at once, and participants only provide one rating. We test whether this rating is different from the ratings at reinstated stage of experiment 1. Cf. hypothesis 2 we expect ratings to be higher for participants from experiment 2.

In experiment 3, for each set of arguments, participants first received a text that included generalisations of all three arguments (an example of which can be found at the beginning of section 3). They then had to rate the conclusion of argument A in a similar fashion as in experiment 1. As we now have a measurement at base and at reinstated stage for experiments 1 and 3, we analyze the results using an analysis of variance with experiment as between-subjects factor, and moment (base versus reinstated stage) as within-subjects factor. Conform hypothesis 3, we expect a significant interaction effect: in experiment 1 rating is lower in reinstated stage; in experiment 3 we expect there to be no or little difference between reinstated and base stage.

Procedure Participants did the experiment online using a Qualtrics (https://www.qualtrics.com/) survey. Participants were first asked a brief set of questions about their age and language capability. They then received a brief explanation of the study. Participants were then asked to rate four sets of arguments. The nature of questioning depended on which experiment they took part in (1, 2, or 3, see design). Although we had 8 sets of arguments, each participant only rated 4 sets (randomised across participants).

Analysis We removed data from participants whose response set was not complete (27, 34, and 20 participants in experiments 1, 2, and 3 respectively). We then calculated the average score for each rating type (reinstated stage, and base stage for experiments 1 and 3). In statistical analysis, we use alpha at .05 for significance.

3.3 Results

Experiment 1 and hypothesis 1 First we test if our replication finds the same pattern of effect as Rahwan et al. (2010). A paired t-test on the data of our experiment 1 found that ratings at the base stage ($M = 5.61, SD = 0.99, 95\%\ CI = [5.42, 5.81]$) were significantly higher compared to the reinstated stage ($M = 5.21, SD = 0.96, 95\%\ CI = [5.02, 5.40]$), $t(102) = 4.636, p < .001$. Thus ratings of argument A's conclusion are found to decrease after attacker B and increase again after counterattacker C, but not to the initial level, like in the original experiment of Rahwan et al. (2010). Figure 2 shows this result and also presents the values observed in Rahwan et al. (2010). It can be seen that apart from the significant difference between conditions/stages, the observed values are also com-

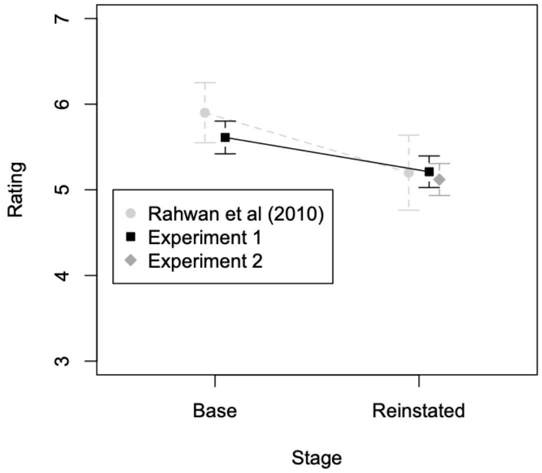


Figure 2: Rating at Base and Reinstated for 2 experiments and Rahwan et al (2010). Error bars show 95% Confidence Intervals; points are horizontally plotted slightly to the side of each other for better readability.

parable between our study and Rahwan et al. (2010) (specifically: there is a strong overlap between the error bars; the means of the two studies fall inside the region defined by the error bars). This confirms the first hypothesis, and replicates the result of Rahwan et al. (2010), this time with a considerably larger set of participants.

Experiment 2 and hypothesis 2 Next, we test if participants give higher ratings if information is presented all at once (experiment 2) compared to sequentially (experiment 1). As the groups had unequal numbers of participants, we ran an independent Welch t-test. There was no significant effect of presentation manner on rating, $t(196.56) = -0.683, p = .496$. Thus, presenting all arguments at once before asking a rating of argument A's conclusion does not lead to higher ratings and, so, the second hypothesis cannot be confirmed. Indeed, Figure 2 shows that the ratings in experiment 2 (M = 5.12, SD = 0.93) are similar (i.e., means are close, error bars overlap largely).

Experiment 3 and hypothesis 3a and 3b Next, we test if it makes a difference if participants are provided with generalisations of all three arguments first. To this end, after checking the equality of variances of each group/experiment with Levene's test, we ran a 2 (experiment: 1 or 3) x 2 (stage: base versus reinstated) mixed ANOVA. We found a significant effect of experiment, $F(1, 211) = 12.906, p < .001$. There was also a significant effect of stage, $F(1, 211) = 53.66, p < .001$. There was no interaction between study and stage, $F(1, 211) = 1.227, p = .269$. Figure 3 illustrates this effect. The parallel lines suggest that in both experiment 1 and experiment 3 ratings are higher in the base stage compared to the reinstated stage, and the reduction in rating between the two is comparable (i.e.: main effect of stage). In addition, ratings in experiment 3 were higher in general (i.e., main effect of experiment). In other words, when participants first see the possible scenarios and then rate the arguments one-by-one, they rate A's conclusion higher in both the reinstated and base stage (compared to the corresponding stages of experiment 1). Thus hypothesis 3a is confirmed but hypothesis 3b is rejected. This is contrary to our expectation of an interaction effect (i.e., crossing lines in Figure 3, with the line for experiment 3 being relatively flat). The expectation was that for experiment 3 the ratings in the reinstated stage are higher than those of experiment 1 (hypothesis 3a), but that in the base stage participants from experiment 3 provided a lower rating than those in experiment 1(hypothesis 3b). We did not observe this interaction, as hypothesis 3a was confirmed but hypothesis 3b was rejected.

4 Discussion

This study purported to (1) replicate the findings of Rahwan et al. (2010) and (2) investigate whether these findings support the void precedence/'fewer attackers is better' principle incorporated in many current graded notions of argument acceptability or whether alternative explanations suggested by Rahwan et al. (2010) and Prakken and de Winter (2018) undercut such support. To summarise our results, our experiment found that participants' ratings of argument A's conclusion decrease after seeing attacker B and increase again after seeing counterattacker C, but not to the initial level. This confirms our hypothesis 1 and replicates Rahwan et al. (2010)'s findings. This is an important result, since replicability is one of the cornerstones of scientific method and since, as we noted in the introduction, social psychology is currently facing a replication crisis. In experiment 2 we found that presenting all arguments at once before asking a rating of argument A's conclusion did not lead to higher ratings compared to those observed in the sequential study of experiment 1 (rejecting hypothesis 2). In experiment 3, we found the opposite when the participants first see the possible scenarios and then rate the arguments after seeing the arguments one-by-one (confirming hypothesis 3a). Finally, in experiment 3 we found that the participants rate A's conclusion higher in the base stage as well, compared to the base stage of experiment 1 (rejecting hypothesis 3b). Thus, we did not find the interaction effect that the confirmation of both hypotheses would entail.

We now discuss various issues relevant to the question whether our results strengthen the arguments for the 'fewer attackers is better' principle.

4.1 Generalisation to Other Patterns

We first recall an observation of Prakken and de Winter (2018) that even if the results support a principle that 'an argument is better if it is unattacked than if it is attacked' in examples following the pattern of Figure 1, the findings cannot be used as support for the more general intuition formalised in Grossi and Modgil (2015, 2019)'s 'fewer attackers is better' principle and Amgoud and Ben-Naim (2013)'s void precedence principle, which, as noted above, is at the heart of many current gradual and ranking-based

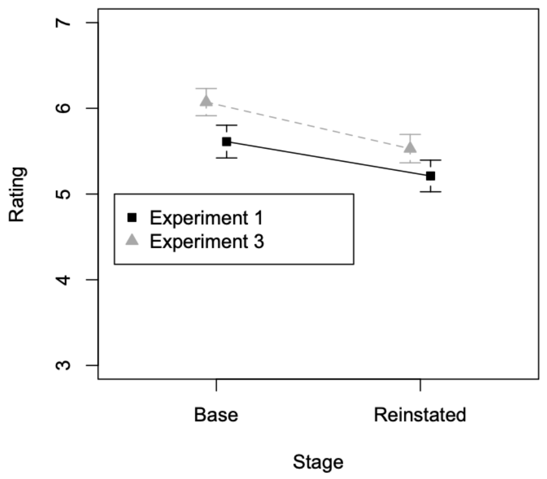


Figure 3: Rating at Base and Reinstated for experiment 1 and 3. Error bars show 95% Confidence Intervals; points are horizontally plotted slightly to the side of each other for better readability.

approaches. The point is that the more general intuition also applies to structures where, unlike in Figure 1, arguments A in $AF1$ and $AF2$ refer to different arguments. Neither in Rahwan et al. (2010)'s nor in our experiments examples of this kind were shown to the participants. So at best Rahwan et al. (2010)'s and our experiments confirm the special case of the void precedence/'fewer attackers is better' principle in which the arguments A in both AFs in Figure 1 are the same argument.

4.2 Suspension of Disbelief

We next note that our results cast some doubts on Rahwan et al. (2010)'s suggested explanation in terms of suspension of disbelief and its variant suggested by Prakken and de Winter (2018). Rahwan et al. do not claim that the introduction of an attacker makes the subjects think/come up with objection, but rather that it causes them to disrupt their suppressing of their already existent objections. In this study, we hypothesised that if confronted with all three arguments at the same time, participants would apply their suspension of disbelief to all the (initially) presented arguments. As our hypothesis 2 is rejected — i.e., introducing all three arguments at the same time does not have a significant effect on the subjects' confidence in A's conclusion — Rahwan et al.'s explanation regarding the disruption of suspension of disbelief cannot be validated.

The same holds for Prakken and de Winter (2018)'s variant of the explanation in terms of suspension of disbelief, according to which the initial introduction of the relevant theory would have made the participants in group 3 aware of possible counterarguments from the start, unlike the participants in group 1. This should have led to the ratings for the conclusion of argument A in the base stage of group 3 being significantly lower than those of group 1, which was our hypothesis 3b. However, this hypothesis was rejected and, surprisingly, not only are the ratings of the third group not lower in the base stage, but they are actually significantly higher. Thus, this is a case where the possibility of an attacker was present from the beginning without it influencing negatively the ratings of the argument that could be attacked. The absence of the expected interaction effect suggests that — despite the introduction of the relevant theory beforehand — the recovery was not complete in the third group either and, thus, Prakken and de Winter's suggestion cannot explain imperfect reinstatement.

What is puzzling is the confirmation of hypothesis 3a in contrast to the rejection of hypothesis 3b, as what we expected was that the introduction of the theory would have opposite results on the base and reinstated stage. One reason why the introduction of the corresponding theory results in an increase of the ratings' level in both stages could be that when introduced with a theory beforehand, the participant gains reassurance. Even though aware of the possibility of an attacker, when an argument is unattacked, the participant has no reason/evidence not to believe it. Thus the introduction of a *possible* attacker might in this case strengthen the attacker's *absence in the base stage*, thus increasing confidence in the conclusion of argument A. This could even be extended to the reinstated stage: participants might feel more reassured after being presented with the instantiation of the possibilities they were originally introduced with. This could also explain why a similar effect did not appear in the second group; in the third group, a participant is originally introduced to possibilities, which are later realised, whereas in the second group a participant misses this intermediate step of reassurance. However, the results of the second group could also be explained by the task of group 2, as we will further discuss in Section 4.4.

4.3 Natural Language versus Formalisation

At this point, it might be thought that our findings strengthen the support for the 'fewer attackers is better' principle. The underlying idea here would be that the participants rated the arguments' conclusions with this principle in mind. We first discuss whether this explanation can be accepted on the basis of Rahwan et al. (2010)'s and our experiments. Later we will discuss to which extent such empirical claims and explanations are relevant for assessing normative models of argumentation.

There is yet another alternative explanation of the results, independently suggested by Prakken and de Winter (2018) and Cramer and Guillaume (2018a,b), namely, that when rating the arguments, the participants may not have had the reinstatement pattern of Figure 1 in mind but a different pattern. All argument sets in the studies of Rahwan et al. (2010) and ourselves were such that the conclusion of argument B attacks a premise of argument A and, likewise, the conclusion of argument C attacks a premise of argument B. Consider again the car battery example from Section 2. It is not obvious that the attacks of B on A and of C on B are asymmetric: since the conclusions and premises involved in these attacks are contradictory, the attacks might also be regarded as symmetric. This is, for instance, possible in *AS*-

PIC$^+$ (Modgil and Prakken, 2014) in which a so-called 'ordinary' premise can rebut an argument with a 'defeasible top rule'. Moreover, AFs generated in *ASPIC*$^+$ include the subarguments of all arguments as separate arguments, including arguments corresponding to a premise.

Thus another plausible AF modelling of the car battery example is AF_2' as shown in Figure 4, where Ap, Bp, and Cp are the subarguments of, respectively, A, B, and C, consisting of their premise. Note that B and Ap attack each other since B undermines Ap (and A) while Ap rebuts B. Likewise for the other attacks. Note also that unlike in $AF2$, in $AF2'$ argument A is not skeptically acceptable. Now it is important to note that a number of participants may have interpreted the examples as in $AF2'$ instead of $AF2$. In an experiment conducted by Cramer and Guillaume (2018a) this was indeed found to be the case. The participants who did so may have rated the conclusion of A lower in the reinstatement stage since A is only credulously acceptable in that stage.

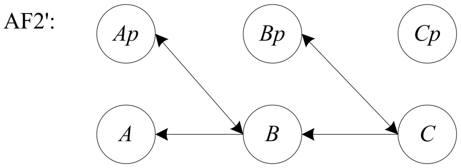


Figure 4: An alternative interpretation of Rahwan et al. (2010)'s examples

This is an instance of a more general problem with this kind of empirical research. In experiments like these, a natural-language reasoning example is formalised, then humans are asked to express an opinion of the natural-language version of the example, and then the humans' responses are compared to the outcome yielded by the semantics of the formalised version of the example. If there is a mismatch between the two, then it is tempting to conclude that humans do not reason according to the formal semantics but such a conclusion is premature, since the mismatch may also be caused by the fact that the formalisation does not correspond to what the humans had in mind (this point is also made by Polberg and Hunter (2018)). Formalising informal reasoning examples is far from trivial since natural language is ambiguous and the same informal way of reasoning may be formalised in the same formalism in different ways. The danger of such mismatches between a formalised example and how humans interpret its natural-language version increases the more abstract the formalism is. As noted by Prakken and de Winter (2018), directly formalising natural-language examples in abstract argumentation formalisms without being guided by a theory of the nature of arguments and their relations may result in ad-hoc modellings (or in the present case in a modelling that is not the only possible one).

This danger may also have materialised in a study of Rosenfeld and Kraus (2015), who modelled natural-language examples in a bipolar argumentation framework (an AF with also support relations) and then observed that the participants did not assess arguments according to its semantics, including the reinstatement pattern. This result was cited by Amgoud (2019) as support for the 'having fewer attackers is better' principle. However, in Rosenfeld and Kraus's examples several attack relations modelled as asymmetric can also be regarded as symmetric. For example, the arguments "We should buy an SUV; it's the right choice for us" and "But we can't afford an SUV, it's too expensive" (where according to Rosenfeld and Kraus the second asymmetrically attacks the first) could by some participants be regarded as two arguments with contradictory conclusions 'we should buy an SUV' and 'we should not buy an SUV'.

A related problem with such empirical reasoning experiments is that it is often hard to make the participants stick to the information that was explicitly given; often they will, either implicitly or explicitly, also take other beliefs and background information into account. Van Benthem (2008) (cited by Rahwan et al. (2010) in support of the relevance of empirical research for normative theories) notes that people in such experiments first go through a "representation" or "modelling" phase in which they construe the relevant scenario of facts and events, and only then make inferences from the construed scenario. He points at the possibility that experimenters overlook that the participants may have added information to the example in the representation phase. Other recent empirical studies in computational argumentation have also pointed at the possibly confounding effect of background information (Cerutti, Tintarev, and Oren, 2014; Polberg and Hunter, 2018; Cramer and Guillaume, 2018b, 2019).

In the present study we tried to avoid the unwanted influence of background information as follows. Overall, the arguments that were used were simple sentences and of a neutral subject matter, to avoid unwanted influence of subjective views. Moreover, the levels of confidence in the eighth set (i.e., the one regarding animal rights, which is not a neutral subject matter), do not deviate from the rest in any way. This suggests a good level of impartiality from the participants. Nevertheless, we cannot exclude the possibility that the results are partly influenced by the *content* of the arguments rather than their *relations*. In order to render such experiments less precarious, future empirical research could try to control for such issues by including manipulation checks, where separate groups of participants evaluate the arguments independently, indicate how they perceive the type and the directionality of the attacks, and so on.

4.4 Order and Cognitive Load

There are further possible explanations of some of the findings. First, the results of the second group could also be explained by the task of group 2, (i.e., the version of manner of presentation that corresponded to group 2) being more challenging. As mentioned by Cramer and Guillaume (2019), a cognitively challenging task might lead to participants choosing a simplifying strategy, in this case, more likely to choose a 'neutral' rating (in this experiment, that would translate to a rating being closer to 4, hence being the lowest rated). The low overall ratings of argument A in group 2, along with the fact that group 2 had the highest dropout rate

(26% of the participants of the second group left the survey unfinished, compared to the 21% of the first and then 15% of the third) might be an indication that the manner of presentation in the second group was more challenging to the subjects.

Second, the imperfect recovery from attack could be a result of *order*. For example, the order of presentation may have had an effect on how the participants perceived the directionality of the attacks; it may be that attacks are more often regarded as originating from the last-presented argument. Moreover, it is possible to assume that participants' confidence in A's conclusion does not go back to its original level because the sooner we are introduced to something, the more likely we are to believe it. As observed by Polberg and Hunter (2018): "presenting a new and correct piece of information that a given person was not aware of does not necessarily lead to changing that person's beliefs". Both in our study and in Rahwan et al. (2010), the arguments were always presented in the same order. Even in group 2, where all the arguments were presented together, argument A is always first. We cannot, therefore, rule out the possibility that the order of arguments also plays a role in participants' confidence.

4.5 The Jump from Is to Ought

Nevertheless, suppose that future experiments are able to reproduce Rahwan et al. (2010)'s findings in examples that unambiguously correspond to Figure 1 and in which background information has been controlled for. Then there is another hurdle to take before it can be concluded that these results support the 'having fewer attackers is better' principle as a normative principle of rational argumentation. This hurdle is that it is not immediately obvious how empirical findings on how people *actually* argue can be relevant for a normative theory on how they *should argue*. Given the growing number of empirical studies in computational argumentation (Cerutti, Tintarev, and Oren, 2014; Rosenfeld and Kraus, 2015; Cramer and Guillaume, 2018a,b, 2019; Polberg and Hunter, 2018) this question is important, but it has no simple answers.

Rahwan et al. (2010) argue that insights from psychological experiments can be relevant to the design of software agents that can argue persuasively with humans. We could think here of IBM's Debater project (Slonim, Bilu, and Alzate, 2021). They may very well be right in this: persuasiveness is a psychological phenomenon, so psychological experiments can obviously yield relevant insights on the persuasiveness of argumentation patterns. However, in our opinion formal models like Dung (1995)'s abstract argumentation theory or more concrete structured accounts like *ASPIC*$^+$, Defeasible Logic Programming (Garcia and Simari, 2004) or assumption-based argumentation (Toni, 2014) do not aim to model *persuasiveness* of arguments. Instead they model the (nonmonotonic) *logical* status of arguments as part of a set of arguments and their logical relations of attack and support.

Rahwan et al. also argue that empirical findings on how humans actually argue are relevant for validating formal semantics of argumentation. However, they are not explicit on when a formal semantics should be changed because of empirical findings on how humans argue and when humans should change their way of arguing to make it fit the formal semantics. One reason to change the formal semantics might be an assumption that humans by-and-large reason correctly. For example, Pollock (1986) argued that the reasoning of humans is guided by internalised rules, while Jackson (1989) argued that any descriptive attempt constitutes a "reconstruction of people's own normative ideas". However, a compelling counterexample is formed by abundant evidence that people are generally very poor at reasoning correctly with and about probabilities (Kahneman, 2011). This is generally not regarded as invalidating probability theory as a normative theory of reasoning with probabilities (here too the relevance of background information has been noted; cf. van Benthem (2008)).

One of us has argued in Prakken (2020) that there is a weaker sense in which empirical findings on how humans reason can be relevant for normative theories of reasoning. Such normative theories should not only be rationally well-founded but also 'cognitively plausible' in that it is not too difficult for people to adhere to their standards. For this reason theories of reasoning should be stated in terms that are natural to people, such as argumentation-related concepts. Such cognitively plausible normative theories may still deviate from how people actually reason, as long as they are stated in terms that are natural to people.

Applying this to the present discussion, this means that empirical research can tell us that people tend to assess arguments in gradual terms, so that it is important to develop normative theories of gradual argument evaluation. However, the specific designs of such theories cannot be based on empirical research alone but should also apply philosophical insights. In the case of gradual and ranking-based semantics, these insights must, to the best of our knowledge, still largely be developed. For instance, the only defence of the 'having fewer attackers is better' principle besides references to empirical findings that we could find is Amgoud and Ben-Naim (2013)'s claim that this principle is "natural". We suggest that a philosophical analysis should aim to clarify what is meant by argument acceptability or argument strength and should take the nature of arguments and their relations into account.

5 Conclusion

In this paper we returned to the experiments of Rahwan et al. (2010) on 'simple reinstatement' patterns in formal argumentation for two reasons. First, we wanted to see whether their results can be replicated. We were able to do so with a considerably larger number of participants, which is a significant result given the current concerns about replicability of results in the social sciences, specifically in social psychology. Second, we wanted to investigate with two variants of Rahwan et al.'s experiments whether empirical findings on how humans evaluate arguments in reinstatement cases can support the 'fewer attackers is better' principle incorporated in many current graded notions of argument acceptability. We can draw the following main conclusions from our investigations.

To start with, our results casted doubt on explanations suggested by Rahwan et al. (2010) and Prakken and de Winter (2018) in terms of suspension of disbelief. According to these explanations, the imperfect recovery of arguments from attack in reinstatement patterns would be due to the triggering at various moments of awareness or consideration of other counterarguments than those presented in the experiment. In our new experiments we did not find evidence for these explanations.

However, we concluded that this does not imply that the experimental results of Rahwan et al. and the present paper support the 'fewer attackers is better' principle. We first noted that the experiments at best support a special case of this principle, namely, 'an argument is better if it is unattacked than if it is attacked' (void precedence). Next we concluded that even the special case is not supported since several alternative explanations cannot yet be ruled out, such as that a number of participants may have had different attack relations in mind. More generally, we highlighted the danger that humans involved in reasoning experiments model and/or interpret examples differently than the experimenters. Finally, we argued that even if future experiments extend to the general case and can rule alternative explanations out, still convincing arguments are needed why and how empirical findings on how humans argue can be relevant for normative models of argumentation. We suggested that the importance of such empirical findings does not lie in what they say about the validity of specific reasoning patterns but in what they say about the general concepts that a normative theory should have in order to be applicable by humans. The issue concerning the jump from 'is' to 'ought' is important since the 'having fewer attackers is better' principle implies that it is rational for arguers to utter as many counterarguments to an argument as possible, even if these arguments are silly and can be easily refuted. Should our normative models of argumentation really encourage arguers to build their arguments on fake news and alternative facts as much as possible?

References

Amgoud, L., and Ben-Naim, J. 2013. Ranking-based semantics for argumentation frameworks. In Liu, W.; Subrahmanian, V.; and Wijsen, J., eds., *Scalable Uncertainty Management. SUM 2013*, number 8078 in Springer Lecture Notes in Computer Science, 134–147. Berlin: Springer Verlag.

Amgoud, L. 2019. A replication study of semantics in argumentation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19)*, 6260–6266.

Benthem, J. v. 2008. Logic and reasoning: Do the facts matter? *Studia Logica* 88:67–84.

Bonzon, E.; Delobelle, J.; Konieczny, S.; and Maudet, N. 2016. A comparative study of ranking-based semantics for abstract argumentation. In *Proceedings of the 30st AAAI Conference on Artificial Intelligence (AAAI 2016)*, 914–920.

Bonzon, E.; Delobelle, J.; Konieczny, S.; and Maudet, N. 2021. A parametrized ranking-based semantics compatible with persuasion principles. *Argument and Computation* 12:49–85.

Caminada, M. 2006. On the issue of reinstatement in argumentation. In Fischer, M.; van der Hoek, W.; Konev, B.; and Lisitsa, A., eds., *Logics in Artificial Intelligence. Proceedings of JELIA 2006*, number 4160 in Springer Lecture Notes in AI, 111–123. Berlin: Springer Verlag.

Cerutti, F.; Tintarev, N.; and Oren, N. 2014. Formal arguments, preferences, and natural language interfaces to humans: an empirical evaluation. In *Proceedings of the 21st European Conference on Artificial Intelligence*, 207–212.

Cramer, M., and Guillaume, M. 2018a. Directionality of attacks in natural language argumentation. In *Proceedings of the fourth Workshop on Bridging the Gap between Human and Automated Reasoning*, 40–46.

Cramer, M., and Guillaume, M. 2018b. Empirical cognitive study on abstract argumentation semantics. In Modgil, S.; Budzynska, K.; and Lawrence, J., eds., *Computational Models of Argument. Proceedings of COMMA 2018*. Amsterdam etc: IOS Press. 413–424.

Cramer, M., and Guillaume, M. 2019. Empirical study on human evaluation of complex argumentation frameworks. In Calimeri, F.; Leone, N.; and Manna, M., eds., *Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA 2019)*, number 11468 in Springer Lecture Notes in AI, 102–115. Berlin: Springer Verlag.

Dung, P. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *n*–person games. *Artificial Intelligence* 77:321–357.

Garcia, A., and Simari, G. 2004. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming* 4:95–138.

Grossi, D., and Modgil, S. 2015. On the graded acceptability of arguments. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 868–874.

Grossi, D., and Modgil, S. 2019. On the graded acceptability of arguments in abstract and instantiated argumentation. *Artificial Intelligence* 275:138–173.

Hunter, A., and Thimm, M. 2017. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research* 59:565–611.

Hunter, A.; Polberg, S.; and Thimm, M. 2020. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. *Artificial Intelligence* 281:103236.

Jackson, S. 1989. What can argumentative practice tell us about argumentation norms? In Maier, R., ed., *Norms in Argumentation. Proceedings of the Conference on Norms*, 113–122. Dordrecht/Providence RI: Foris Publication.

Jakobovits, H. 2000. *On the Theory of Argumentation Frameworks*. Doctoral dissertation Free University Brussels.

Kahneman, D. 2011. *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.

Modgil, S., and Prakken, H. 2014. The ASPIC+ framework for structured argumentation: a tutorial. *Argument and Computation* 5:31–62.

Pashler, H., and Wagenmakers, E. 2012. Editors' introduction to the special section on replicability in psychological science: a crisis in confidence? *Perspectives on Psychological Science* 7:528–530.

Polberg, S., and Hunter, A. 2018. Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *International Journal of Approximate Reasoning* 93:487–543.

Pollock, J. 1986. *Contemporary Theories of Knowledge*. Littlefield, NY: Rowman & Littlefield.

Prakken, H., and de Winter, M. 2018. Abstraction in argumentation: necessary but dangerous. In Modgil, S.; Budzynska, K.; and Lawrence, J., eds., *Computational Models of Argument. Proceedings of COMMA 2018*. Amsterdam etc: IOS Press. 85–96.

Prakken, H. 2020. On validating theories of abstract argumentation frameworks: the case of bipolar argumentation frameworks. In *Proceedings of the 20th Workshop on Computational Models of Natural Argument*, volume 2669 of *CEUR Workshop Proceedings*, 21–30.

Rahwan, I.; Madakkatel, M.; Bonnefon, J.-F.; Awan, R.; and Abdallah, S. 2010. Behavioural experiments for assessing the abstract semantics of reinstatement. *Cognitive Science* 34:1483–1502.

Rosenfeld, A., and Kraus, S. 2015. Providing arguments in discussions based on the prediction of human argumentative behavior. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, 1320–1327.

Slonim, N.; Bilu, Y.; and Alzate, C. 2021. An autonomous debating system. *Nature* 591:397–384.

Thimm, M., and Kern-Isberner, G. 2014. On controversiality of arguments and stratified labelings. In Parsons, S.; Oren, N.; Reed, C.; and Cerutti, F., eds., *Computational Models of Argument. Proceedings of COMMA 2014*. Amsterdam etc: IOS Press. 413–420.

Toni, F. 2014. A tutorial on assumption-based argumentation. *Argument and Computation* 5:89–117.

Arguing about Complex Formulas: Generalizing Abstract Dialectical Frameworks

Jesse Heyninck[1,2] , **Matthias Thimm**[3] , **Gabriele Kern-Isberner**[1] , **Tjitze Rienstra**[3] , **Kenneth Skiba**[3]

[1]Technische Universität Dortmund, Dortmund, Germany
[2]University of Cape Town and CAIR, South-Africa
[3]University of Koblenz-Landau, Koblenz, Germany

Abstract

Abstract dialectical frameworks (in short, ADFs) are a unifying model of formal argumentation, where argumentative relations between arguments are represented by assigning acceptance conditions to atomic arguments. This idea is generalized by letting acceptance conditions being assigned to complex formulas, resulting in conditional abstract dialectical frameworks (in short, cADFs). We define the semantics of cADFs in terms of a non-truth-functional four-valued logic, and study the semantics in-depth, by showing existence results and proving that all semantics are generalizations of the corresponding semantics for ADFs.

1 Introduction

Formal argumentation is one of the major approaches to knowledge representation. In the seminal paper (Dung 1995), *abstract argumentation frameworks* were conceived of as directed graphs where nodes represent arguments and edges between these nodes represent attacks. So-called *argumentation semantics* determine which sets of arguments can be reasonably upheld together given such an argumentation graph. Various authors have remarked that other relations between arguments are worth consideration. For example, in (Cayrol and Lagasquie-Schiex 2005), *bipolar argumentation frameworks* are developed, where arguments can support as well as attack each other.

The last decades saw a proliferation of such extensions of the original formalism of (Dung 1995), and it has often proven hard to compare the resulting different dialects of the argumentation formalisms. To cope with the resulting multiplicity, (Brewka et al. 2013) introduced *abstract dialectical argumentation* that aims to unify these different dialects (Polberg 2016). Just like in (Dung 1995), *abstract dialectical frameworks* (in short, ADFs) are directed graphs. In difference to abstract argumentation frameworks, however, in ADFs, edges between nodes do not necessarily represent attacks but can encode *any* relationship between arguments. Such a generality is achieved by associating an *acceptance condition* with each argument, which is a Boolean formula in terms of the parents of the argument that expresses the conditions under which an argument can be accepted. This results in an ADF being defined as a triple (At, L, C) where At represents a set of atoms or arguments, $L \subseteq \mathsf{At} \times \mathsf{At}$ represents a set of argumentative relations between the atoms and C is a set of acceptance conditions C_s for every s. As such, ADFs are able to capture all of the major semantics of abstract argumentation and offer a general framework for argumentation-based inference. Furthermore, ADFs were shown to be able to capture a number of non-argumentative formalisms such as logic programming (Brewka et al. 2013). Recently, first attempts were made to translate non-monotonic conditional logics in ADFs (Heyninck et al. 2019).

However, there are limits to the representative capabilities of ADFs, both on a conceptual as well as a more technical level. On the conceptual level, acceptance conditions are assigned to atoms, which means that, e. g., an attack on a set of arguments cannot be captured by ADFs. For example, to state that the set $\{p, q\}$ is attacked by r we would have to be able to set the acceptance condition of $p \wedge q$ to $\neg r$, which is not possible in ADFs. Likewise, it is not immediately obvious how to represent more complicated logic programming languages in ADFs, such as disjunctive logic programming. Such limitations are, not unsurprisingly, also reflected on a more technical level. For example, a (polynomial) translation of disjunctive logic programming into ADFs is impossible in view of complexity results on disjunctive logic programming and ADFs. Finally, in (Heyninck et al. 2019) shows that only a fragment of the full language of conditional logics can be translated in ADFs in view of their limited syntax.

In this paper, we generalize ADFs as to allow for the assignment of acceptance conditions to complex formulas. This results in *conditional abstract dialectical frameworks* (in short, cADFs) which are sets of acceptance pairs of the form $\phi \lhd \psi$ with arbitrary formulas ϕ and ψ, interpreted as a defeasible version of "ϕ is the case if and only if ψ is the case". The semantics of cADFs are formulated as a generalization of the semantics of ADFs, with the Γ-function, on its turn based on a non-truth-functional four-valued logic, as a central component. Some of the main results include existence results for all the major semantics, as well as the definition of the so-called *grounded state*, a single-state semantics which can be iteratively constructed and represents the minimal information entailed by a given cADF.

Outline of this Paper: We first state all the necessary preliminaries in Section 2 on propositional logic (Section

2.1), and abstract dialectical argumentation (Section 2.2). The syntax of *conditional abstract dialectical frameworks* cADFs is introduced in Section 3. In Section 4, a four-valued logic, which will form the basis of the semantics of cADFs, is defined and studied. In Section 5, we then define and study the admissible, complete, preferred and grounded semantics for cADFs. A unique, iteratively constructible analogue to the grounded extension, called the *grounded state*, is introduced in Section 6. Related work is discussed in Section 7 and a conclusion is drawn in Section 8.

2 Preliminaries

In the following, we briefly recall some general preliminaries on propositional logic, as well as technical details on conditional logic and ADFs (Brewka et al. 2013).

2.1 Propositional Logic

For a set At of atoms let $\mathcal{L}(\mathsf{At})$ be the corresponding propositional language constructed using the usual connectives $\wedge$ (*and*), $\vee$ (*or*), $\neg$ (*negation*) and $\rightarrow$ (*material implication*). A (classical) *interpretation* (also called *possible world*) ω for a propositional language $\mathcal{L}(\mathsf{At})$ is a function $\omega : \mathsf{At} \rightarrow \{\mathsf{T}, \mathsf{F}\}$. Let $\mathcal{V}^2(\mathsf{At})$ denote the set of all interpretations for At. We simply write $\mathcal{V}^2$ if the set of atoms is implicitly given. An interpretation ω *satisfies* (or is a *model* of) an atom $a \in \mathsf{At}$, denoted by $\omega \models a$, if and only if $\omega(a) = \mathsf{T}$. The satisfaction relation $\models$ is extended to formulas as usual. For $\Phi \subseteq \mathcal{L}(\mathsf{At})$ we also define $\omega \models \Phi$ if and only if $\omega \models \phi$ for every $\phi \in \Phi$. Define the set of models $\mathsf{Mod}^2(X) = \{\omega \in \mathcal{V}^2(\mathsf{At}) \mid \omega \models X\}$ for every formula or set of formulas X. A formula or set of formulas X_1 *entails* another formula or set of formulas X_2, denoted by $X_1 \vdash X_2$, if $\mathsf{Mod}^2(X_1) \subseteq \mathsf{Mod}^2(X_2)$. A formula ϕ is a tautology if $\mathsf{Mod}^2(\phi) = \mathcal{V}^2(\mathsf{At})$ and a falsity if $\mathsf{Mod}^2(\phi) = \emptyset$.

2.2 Abstract Dialectical Frameworks

We briefly recall some technical details on ADFs following loosely the notation from (Brewka et al. 2013). An ADF D is a tuple $D = (\mathsf{At}, L, C)$ where At is a finite set of atoms, $L \subseteq \mathsf{At} \times \mathsf{At}$ is a set of links, and $C = \{C_s\}_{s \in \mathsf{At}}$ is a set of total functions $C_s : 2^{par_D(\mathsf{At})} \rightarrow \{\top, \bot\}$ for each $s \in \mathsf{At}$ with $par_D(s) = \{s' \in \mathsf{At} \mid (s', s) \in L\}$ (also called acceptance functions). An acceptance function C_s defines the cases when the statement s can be accepted (truth value $\top$), depending on the acceptance status of its parents in D. By abuse of notation, we will often identify an acceptance function C_s by its equivalent *acceptance condition* which models the acceptable cases as a propositional formula.

Example 1. We consider the following ADF $D_1 = (\{a, b, c, d\}, L, C)$ with $L = \{(a, b), (b, a), (a, c), (b, c)\}$ and $C_a = \neg b$, $C_b = \neg a$, and $C_c = \neg a \vee \neg b$.
Informally, the acceptance conditions can be read as "a is accepted if b is not accepted", "b is accepted if a is not accepted" and "c is accepted if a is not accepted or b is not accepted".

An ADF $D = (\mathsf{At}, L, C)$ is interpreted through 3-valued interpretations $\nu : \mathsf{At} \rightarrow \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$. We denote the set of all 3-valued interpretations over At by $\mathcal{V}^3(\mathsf{At})$. We define the information order $<_i$ over $\{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ by making U the minimal element: $\mathsf{U} <_i \mathsf{T}$ and $\mathsf{U} <_i \mathsf{F}$, and $\dagger \leq_i \ddagger$ iff $\dagger <_i \ddagger$ or $\dagger = \ddagger$ for any $\dagger, \ddagger \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$. This order is lifted point-wise as follows (given $\nu, \nu' \in \mathcal{V}^3(\mathsf{At})$): $\nu \leq_i \nu'$ iff $\nu(s) \leq_i \nu'(s)$ for every $s \in \mathsf{At}$. The set of two-valued interpretations extending a 3-valued interpretation v is defined as $[\nu]^2 = \{\omega \in \mathcal{V}^2(\mathsf{At}) \mid \nu \leq_i \omega\}$. Given a set of 3-valued interpretations $V \subseteq \mathcal{V}^3(\mathsf{At})$, $\sqcap_i V$ is the 3-valued interpretation defined via $\sqcap_i V(s) = \dagger$ if for every $\nu \in V$, $\nu(s) = \dagger$, for any $\dagger \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$, and $\sqcap_i V(s) = \mathsf{U}$ otherwise. Truth values based on a three-valued interpretations can now be assigned to complex formulas ϕ by taking $\sqcap_i[\nu]^2(\phi)$. All major semantics of ADFs single out three-valued interpretations in which the truth value of every atom $s \in \mathsf{At}$ is, in some sense, in alignment or agreement with the truth value of the corresponding condition C_s. The Γ-function enforces this intuition by mapping an interpretation ν to a new interpretation $\Gamma_D(\nu)$, which assigns to every atom s exactly the truth value assigned by ν to C_s, i.e.:

$$\Gamma_D(\nu) : \mathsf{At} \rightarrow \{\mathsf{T}, \mathsf{F}, \mathsf{U}\} \text{ where } s \rightarrow \sqcap_i\{\omega(C_s) \mid \omega \in [\nu]^2\}.$$

Definition 1. Let $D = (\mathsf{At}, L, C)$ be an ADF with $\nu \in \mathcal{V}(\mathsf{At})$ a 3-valued interpretation:

- ν is *admissible for* D iff $\nu \leq_i \Gamma_D(\nu)$.
- ν is *complete for* D iff $\nu = \Gamma_D(\nu)$.
- ν is *preferred for* D iff ν is $\leq_i$-maximal among all admissible interpretations.
- ν is *grounded for* D iff ν is $\leq_i$-minimal among all complete interpretations.

We denote by $\mathsf{admissible}$, $\mathsf{complete}(D)$, $\mathsf{prf}(D)$, and $\mathsf{grounded}(D)$ the sets of complete, preferred, and grounded interpretations of D, respectively.

Notice that ν is admissible iff $\nu(s) \leq_i \sqcap_i[\nu]^2(C_s)$ for every $s \in S$ and likewise, ν is complete iff $\nu(s) = \sqcap_i[\nu]^2(C_s)$ for every $s \in S$. It can thus be observed that the logic defined by $\sqcap_i[\nu]^2$ is, essentially, the logic underlying ADFs, in the sense that the evaluation of acceptance conditions under $\sqcap_i[\nu]^2$ is the fundamental operation underlying every semantical notion of ADFs. It should be furthermore noted that $\sqcap_i[\nu]^2$ does not give rise to a *truth-functional logic*. Recall that a truth-functional logic is a logic in which the truth value assigned to a complex formula is a function of the truth values of its component formulas. E.g. for a truth-functional logic, the truth value of $a \vee \neg b$ is determined completely by the truth value of a and $\neg b$. For example, given $\nu(a) = \mathsf{U}$ and $\nu(b) = \mathsf{U}$, $\sqcap_i[\nu]^2(a \vee \neg a) = \mathsf{T}$ whereas $\sqcap_i[\nu]^2(a \vee \neg b) = \mathsf{U}$.

Example 2 (Example 1 continued)**.** The ADF of Example 1 has three complete models ν_1, ν_2, ν_3 with:

$$\begin{array}{lll} \nu_1(a) = \mathsf{T} & \nu_1(b) = \mathsf{F} & \nu_1(c) = \mathsf{T} \\ \nu_2(a) = \mathsf{F} & \nu_2(b) = \mathsf{T} & \nu_2(c) = \mathsf{T} \\ \nu_3(a) = \mathsf{U} & \nu_3(b) = \mathsf{U} & \nu_3(c) = \mathsf{U} \end{array}$$

ν_3 is the grounded interpretation whereas ν_1 and ν_2 are both preferred.

3 Syntax of cADFs

The syntactical representation $D = (S, L, C)$ of an ADF contains some superfluous information. In particular, as there is a link between a statement s and s' iff s is mentioned in the acceptance condition of s', the set of links does not contain any information not already derivable from the set of acceptance conditions C. As such, given a set of atoms S, we can simply write an ADF as a set of statements $s \lhd C_s$ if C_s is the acceptance condition of s. So the ADF D_1 from Example 1 can be simply written as:

$$D_1 = \{a \lhd \neg b, b \lhd \neg a, c \lhd \neg a \vee \neg b\}$$

An ADF is determined by a set of propositional formulæ that, when evaluated to true, make a certain statement, which is a simple atom, true as well, and when evaluated to false, make the simple atom false as well. In other words, $\lhd$ can be read as a *approximate if and only if*: $s \lhd C_s$ means that the truth-values s and C_s should be aligned. $\lhd$ can truly be read as a *approximate iff*, since it might not always be possible to align the truth values of s and C_s in such a way that they take on exactly the same (determinate) truth value. To see this, consider, e. g., $a \lhd \neg a$. We generalise this framework by allowing these statements to be arbitrary propositional formulæ:

Definition 2. Given a set of atoms At, a *conditional abstract dialectical framework* cADF Π w.r.t. At is a finite set of *acceptance pairs* over At, where an *acceptance pair* is of the form:

$$\phi \lhd \psi$$

with ϕ and ψ being propositional formulæ over At.

In order to stick to ADF terminology we call ϕ the *statement* and ψ the *condition* of the acceptance pair $\phi \lhd \psi$. We omit the reference to the signature At when it is clear from context.

Example 3. Consider a cADF $\Pi_1 = \{c_1, c_2, c_3\}$ with

$$\begin{aligned} c_1 : &\quad p \vee s \vee q \lhd \top \\ c_2 : &\quad p \wedge s \lhd \neg q \\ c_3 : &\quad (p \wedge q) \vee (p \wedge s) \lhd t \end{aligned}$$

This cADF can be used to model an argument of a group of friends about making plans on a Sunday. They are discussing whether to go to a party (p), to the swimming pool (s) or go to a pub quiz (q). They want to do at least one of these three things (c_1). However, if they go to the quiz, they won't be able to still go to the pool *and* go to the party (represented by the attack of q on $p \wedge s$ in c_2). If everyone arrives on time (t), they would like to go to both the quiz and the party, or to both the pool and the party (c_3). We notice that without adding further atoms, an attack from q on the set $\{p, s\}$, as formalized by c_2, cannot be represented in ADFs.

We observe that this simple generalization w.r.t. ADFs results in the following additional points of expressiveness in comparison to ADFs:

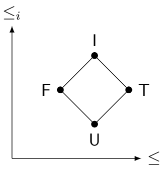
- cADFs allow for complex formulas as statements, as demonstrated by $(p \wedge q) \vee (p \wedge s) \lhd t$ in Example 3.
- cADFs allow for "incomplete" specifications, i.e. they do not force the user to formulate an acceptance condition for every atom, as demonstrated in Example 3, where t has no acceptance condition.
- cADFs allow for "overspecifications" or conflicting specifications, as demonstrated by the cADF $\{a \lhd b, \neg a \lhd b\}$ where both a and $\neg a$ have the acceptance condition b.
- cADFs allow for indeterminism, as demonstrated by the cADF $\{a \vee b \lhd \top\}$, where $a \vee b$ is required to be true, but no further information on which of the disjuncts is required to be true is given.

To cope with this higher expressiveness semantically, it will prove useful to move from three-valued interpretations to four-valued interpretations. To assign truth values to complex formulas on the basis of four-valued interpretations, we generalize the logic defined by $\sqcap_i[v]^2$ to a four-valued setting in Section 4. We then generalize the semantics of ADFs to cADFs on the basis of this four-valued logic in Section 5.

4 A Four-Valued Logic Based on Completions

We first define a four-valued logic 4CL which generalizes the idea of completions known from the logic underlying ADFs defined by $[\nu]^2$, which preserves classical tautologies and falsities. We first recall four-valued interpretations. A *four-valued interpretation* v : At $\rightarrow$ $\{\mathsf{T}, \mathsf{F}, \mathsf{I}, \mathsf{U}\}$ assigns to every atom a truth value T (true), F (false), U (undecided) or I (inconsistent). We will also write an interpretation $v \in \mathcal{V}^4(\{a_1, \ldots, a_n\})$ as $v(a_1) \ldots v(a_n)$, e. g., v over $\{p, q\}$ with $v(p) = \mathsf{T}$ and $v(q) = \mathsf{U}$ will be written as TU. We denote the set of four-valued interpretations over At by $\mathcal{V}^4(\mathsf{At})$. Notice that $\mathcal{V}^2(\mathsf{At}) \subseteq \mathcal{V}^3(\mathsf{At}) \subseteq \mathcal{V}^4(\mathsf{At})$. If it is clear that an interpretation is two- respectively three-valued, we will denote it by (a possibly indexed) ω respectively ν.

Two useful orders over these truth values are the *information order* $\leq_i$ and the *truth order* $\leq_t$, which form the following bilattice-structure (Fitting 2006):



Notice that $\mathcal{V}^4(\mathsf{At})$ also forms a bounded lattice under $\leq_i$ with v_U and v_I as least and greatest element respectively (where v_U is defined as the interpretation that sets $v_\mathsf{U}(a) = \mathsf{U}$ for every $a \in \mathsf{At}$ and v_I is defined as $v_\mathsf{I}(a) = \mathsf{I}$ for every $a \in \mathsf{At}$).

We shall interpret the four truth values, at least for atoms, in the same way as (Belnap 2019): U (*undecided*) means that we have no explicit information for either the truth nor the falsity of an atom. T (*true*) respectively F (*false*) means that we have explicit information only for the truth respectively the falsity of the atom in question. Finally, I (*inconsistent*) means that we have explicit information for both the truth and the falsity of the atom in question. When it comes to complex formulas, we shall see that we take a somewhat hybrid position between the truth values expressing merely explicit information and the truth values standing for objective truth. In particular, the logic we will define here will allow for *logically contingent* formulas, i. e., formulas which are neither classical tautologies nor classical falsities, to be assigned any of the four truth values, whereas classical tautologies and classical falsities will always be assigned T respectively F by any interpretation. Intuitively, this means that even though the truth value of $s \in \mathsf{At}$ might be undetermined (U) or inconsistent (I), the logic will still evaluate $s \vee \neg s$ as true. This is in complete agreement with ADFs, where tautologies and logical falsities are always evaluated in agreement with classical logic by $\sqcap_i[v]^2$.

Semantically, we proceed as follows: we construct a set of sets of (two-valued) worlds on the basis of a four-valued interpretation v that represents the beliefs expressed by v. Just like in the logic underlying ADFs $\sqcap_i[\nu]^2$, a set of (two-valued) worlds will be used to represent a three-valued interpretation ν. The worlds in $[\nu]^2$ represent equally plausible candidates of the actual world in view of the beliefs expressed by the three-valued interpretation ν. Likewise, a set of three-valued interpretations $[v]^3$ will be used to represent the information expressed by a four-valued interpretation v. $[v]^3$ consists of the three-valued interpretations that *jointly* represent the information expressed by v. Notice the difference with $[\nu]^2$: $[\nu]^2$ consists of equally plausible candidates of the actual world in view of the information expressed by v, whereas $[v]^3$ contains interpretations that *taken together* represent the information expressed by v. We now develop this idea in more formal details.

Given a four-valued interpretation, we define the set of two-valued completions of v, $[v]^2$, in two steps. First, we construct $[v]^3$, which converts $v \in \mathcal{V}^4(\mathsf{At})$ to a set of three-valued interpretations $[v]^3 \subseteq \mathcal{V}^3(\mathsf{At})$. Then, we obtain $[v]^2 \subseteq \wp(\mathcal{V}^2(\mathsf{At}))$ by converting every three-valued interpretation $\nu \in [v]^3$ to a set of two-valued interpretations $[\nu]^2$.

Definition 3. Given a four-valued interpretation $v \in \mathcal{V}(\mathsf{At})$, $[v]^3 = \{\nu \in \mathcal{V}^3(\mathsf{At}) \mid \text{for every } s \in \mathsf{At} : \text{ if } v(s) = \mathsf{I} \text{ then } \nu(s) \in \{\mathsf{T},\mathsf{F}\}, \nu(s) = v(s) \text{ otherwise}\}$

In other words, $[v]^3$ is obtained by replacing every assignment of an atom s to I to an assignment of s to T or to F.

Notice that $[v]^3$ consists of the $\leq_i$-maximal three-valued interpretations that v extends:

Fact 1. For any $v \in \mathcal{V}^4(\mathsf{At})$, $[v]^3 = \max_{\leq_i}(\{\nu \in \mathcal{V}^3(\mathsf{At}) \mid \nu \leq_i v\})$.[1]

Example 4. Consider $v = \mathsf{TUI}$ over $\Sigma = abc$. Then $[v]^3 = \{\mathsf{TUT}, \mathsf{TUF}\}$.

We are now ready to define the *four-valued completions* $[v]^4$ of v:

Definition 4. Given some $v \in \mathcal{V}^4(\mathsf{At})$, the *four-valued completions of* v are defined as: $[v]^4 = \{[v']^2 \mid v' \in [v]^3\}$.

Thus, $[v]^4$ is obtained by first constructing $[v]^3$, and then taking for every $\nu \in [v]^3$ the set of two-valued completions of ν. The intuition behind this is as follows: $v(s) = \mathsf{I}$ means that we have information for both s being true and s being false. Thus, the interpretations where we set $\nu_1(s) = \mathsf{T}$ and $\nu_2(s) = \mathsf{F}$ are both (partial yet consistent) representations of the state of the world represented by v. Hence $[v]^3$ can be viewed as the set of three-valued interpretations that together form the representation of the state of the world represented by v. We then construct for every such representation a set of two-valued interpretations, which represent equally plausible candidates of the state of the world represented by $\nu \in [v]^3$. Altogether, $[v]^4$ contains a set of set of possible worlds, which together represent our knowledge about the actual state of the world.

It is useful to notice that for a three-valued interpretation $v \in \mathcal{V}^3(\mathsf{At})$, $[v]^4 = \{[v]^2\}$.

Example 5. Consider $v = \mathsf{TUI}$ over $\Sigma = \{abc\}$. Since $[v]^3 = \{\mathsf{TUT}, \mathsf{TUF}\}$, $[\mathsf{TUT}]^2 = \{\mathsf{TTT}, \mathsf{TFT}\}$ and $[\mathsf{TUF}]^2 = \{\mathsf{TTF}, \mathsf{TFF}\}$, we see that $[v]^4 = \{\{\mathsf{TTT}, \mathsf{TFT}\}, \{\mathsf{TTF}, \mathsf{TFF}\}\}$.

Notice that, in order to retain the four-valued structure of an interpretation v in its four-valued completion $[v]^4$, the two-step nature of the construction of $[v]^4$ and the resulting nested structure of $[v]^4$ is essential. Indeed, if $[v]^4$ would merely consist of possible worlds, we would somehow have to choose between letting the members $\omega \in [v]^4$ stand as equally plausible candidates of the actual world or partial descriptions of the information given by v, i. e., we would have to choose between U and I. Conceiving of $[v]^4$ as a set of sets of worlds avoids this choice: sets of worlds $\mathcal{V}' \in [v]^4$ represent partial descriptions of the information given by v, and members of these sets of worlds $\omega \in \mathcal{V}'$ represent equally plausible candidates of the information in $\mathcal{V}'$.

We can now define the assignment of truth values of complex formulas given an interpretation v based on our set of four-valued completions $[v]^4$:

Definition 5. Given a formula ϕ and an interpretation v, then:

$$v(\phi) = \begin{cases} \mathsf{T} & \text{if for every } \Omega' \in [v]^4, \sqcap_i\Omega'(\phi) = \mathsf{T} \\ \mathsf{F} & \text{if for every } \Omega' \in [v]^4, \sqcap_i\Omega'(\phi) = \mathsf{F} \\ \mathsf{I} & \text{if for some } \Omega_1 \in [v]^4, \sqcap_i\Omega_1(\phi) = \mathsf{T} \\ & \text{and for some } \Omega_2 \in [v]^4, \sqcap_i\Omega_2(\phi) = \mathsf{F} \\ \mathsf{U} & \text{otherwise} \end{cases}$$

Thus, a complex formula ϕ is assigned T (respectively F) relative to an interpretation v if every four-valued completion $\Omega' \in [v]^4$ of v, assigns T (respectively F) to ϕ. If there is disagreement among the four-valued completions of v on which determinate truth value ϕ should be assigned, $v(\phi) = \mathsf{I}$. Finally, if some of the four-valued completions of v do not assign any determinate truth value to ϕ, $v(\phi) = \mathsf{U}$.

This way of deriving a truth value for complex formulas on the basis of a four-valued interpretation is, to the best of our knowledge, completely new. It is perfectly in line with $\sqcap_i[v]^2$, the logic underlying ADFs, in the sense that for any three-valued interpretation $\nu \in \mathcal{V}^3(\mathsf{At})$ and any formula $\phi \in \mathcal{L}$, $\nu(\phi) = \sqcap_i[\nu]^2(\phi)$.

Fact 2. For any $\nu \in \mathcal{V}^3(\mathsf{At})$ and any $\phi \in \mathcal{L}(\mathsf{At})$, $\nu(\phi) = \sqcap_i[\nu]^2(\phi)$.

[1] Some proofs have been left out in view of spatial limitations.

Example 6. Consider $v = \mathsf{TUI}$ over $\Sigma = abc$. Observe that $[v]^4 = \{\{\mathsf{TTT}, \mathsf{TFT}\}, \{\mathsf{TTF}, \mathsf{TFF}\}\}$. Thus, we have the following assignments to complex formulas:

- $v(a \wedge c) = \mathsf{I}$, since $\sqcap_i\{\mathsf{TTT}, \mathsf{TFT}\}(a \wedge c) = \mathsf{T}$ and $\sqcap_i\{\mathsf{TTF}, \mathsf{TFF}\}(a \wedge c) = \mathsf{F}$;
- $v(b \wedge c) = \mathsf{U}$, since $\sqcap_i\{\mathsf{TTT}, \mathsf{TFT}\}(b \wedge c) = \mathsf{U}$ and $\sqcap_i\{\mathsf{TTF}, \mathsf{TFF}\}(b \wedge c) = \mathsf{F}$;
- $v(a \wedge \neg a) = \mathsf{F}$, since $\sqcap_i\{\mathsf{TTT}, \mathsf{TFT}\}(a \wedge \neg a) = \mathsf{F}$ and $\sqcap_i\{\mathsf{TTF}, \mathsf{TFF}\}(a \wedge \neg a) = \mathsf{F}$;

We first observe that 4CL preserves classical tautologies and falsities:

Proposition 1. If $\vdash \phi$ then for any $v \in \mathcal{V}^4$, $v(\phi) = \mathsf{T}$. Likewise, if $\vdash \neg\phi$ then for any $v \in \mathcal{V}^4$, $v(\phi) = \mathsf{F}$.

Proof. This is so because for any $v \in \mathcal{V}^4$, $[v]^2(\phi) = \mathsf{T}[\mathsf{F}]$ for any tautology[falsity]. □

We can also define entailment in 4CL in the usual way. We set T and I as designated truth values in compliance with (Belnap 2019):

Definition 6. Given a set of formulas $\Psi \cup \{\phi\} \subseteq \mathcal{L}(\mathsf{At})$, $\mathsf{Mod}^4(\Psi) = \{v \in \mathcal{V}^4(\mathsf{At}) \mid v(\psi) \in \{\mathsf{T}, \mathsf{I}\}$ for every $\psi \in \Psi\}$ and $\Psi \models_{\mathsf{4CL}} \phi$ iff $\mathsf{Mod}^4(\Psi) \subseteq \mathsf{Mod}^4(\phi)$.

We now show that $\models_{\mathsf{4CL}}$ is paraconsistent:

Proposition 2. There exists a set of formulas $\Phi \subseteq \mathcal{L}(\mathsf{At})$ s.t. $\mathsf{Mod}(\Phi) = \emptyset$ yet $\mathsf{Mod}^4(\Phi) \neq \emptyset$.

Proof. Consider the signature $\mathsf{At} = \{p, q\}$, $\Phi = \{p, \neg p\}$ and $v \in \mathcal{V}^4(\mathsf{At})$ with $v(p) = \mathsf{I}$ and $v(q) = \mathsf{U}$. $[v]_4^2 = \{\{\mathsf{TT}, \mathsf{TF}\}, \{\mathsf{FT}, \mathsf{FF}\}\}$ and thus $v(\neg p) = v(p) = \mathsf{I}$ and $v(q) = \mathsf{U}$. □

We notice, though, that there might still be sets of formulas $\Phi \in \mathcal{L}(\mathsf{At})$ for which no $v \in \mathcal{V}^4(\mathsf{At})$ exists s.t. $v(\phi) \in \{\mathsf{T}, \mathsf{I}\}$ for every $\phi \in \Phi$. To see this, it suffices to observe that for any falsity ϕ and any interpretation $v \in \mathcal{V}^4(\mathsf{At})$, $v(\phi) = \mathsf{F}$. In other words, the logic defined above is still explosive for contradictions. But for inconsistent sets of formulas containing no contradictions, the logic is non-explosive.

Proposition 3. For every set of formulas $\Phi \subseteq \mathcal{L}(\mathsf{At})$ s.t. for every $\phi \in \Phi$, $\mathsf{Mod}(\phi) \neq \emptyset$, there is some $v \in \mathcal{V}^4(\mathsf{At})$, s.t. $v(\phi) \in \{\mathsf{I}, \mathsf{T}\}$ for every $\phi \in \Phi$.

Remark 1. Observe that the logic 4CL, like the logic defined by $\sqcap_i[v]^2$, is *not* truth-functional. To see this consider the interpretation v with $v(a) = \mathsf{U}$ and $v(b) = \mathsf{U}$. Then $v(a \vee \neg a) = \mathsf{T}$ yet $v(b \vee \neg a) = \mathsf{U}$. Thus, we see that 4CL is not truth-functional, as $v(a) = v(b) = \mathsf{U}$ yet $v(a \vee \neg a) \neq v(b \vee \neg a)$.

We finally notice the following useful property:

Proposition 4. Let $v_1, v_2 \in \mathcal{V}^4(\mathsf{At})$ and $\phi \in \mathcal{L}(\mathsf{At})$ be given. Then $v_1 \leq_i v_2$ implies $v_1(\phi) \leq_i v_2(\phi)$.

5 Semantics of cADFs

In this section, we define, motivate and study the semantics of cADFs. In more detail, in Section 5.1 we define the central Γ_Π-function and use it to define the main semantics for cADFs. In Section 5.2 we motivate the design choices made in generalizing the Γ-function from ADFs to cADFs. In Section 5.3 we show some central semantical properties of the semantics of cADFs.

5.1 The Γ_Π-Function and Resulting cADF-Semantics

A cADF Π over At is interpreted through 4-valued interpretations. Just like for ADFs, it is of crucial importance to construct a Γ-function that allows to characterize all semantics in terms of (post-)fixpoints of this function.

The Γ-function, conceptually, performs the following operation for ADFs: given an interpretation ν and an ADF D, $\Gamma_D(\nu)$ assigns to every atom s the truth value determined by ν and C_s. In other words, $\Gamma_D(\nu)(s)$ is the value s should take in view of the information expressed by $s \lhd C_s$ and ν. If (for every $s \in S$), this value is compatible (in terms of $\leq_i$) with the actual value $v(s)$, then v will be admissible or even complete. We generalize this idea to the case of cADFs, and take, intuitively, $\Gamma_\Pi(v)$ as the *set of interpretations* that evaluate ϕ in accordance with the information given by $\phi \lhd \psi \in \Pi$ and v. More formally, we define the Γ-function $\Gamma_\Pi : \mathcal{V}^4(\mathsf{At}) \to \wp(\mathcal{V}^4(\mathsf{At}))$ for a cADF Π and an interpretation $v \in \mathcal{V}^4(\mathsf{At})$ as follows:

$$\Gamma_\Pi(v) = \min_{\leq_i}\{v' \in \mathcal{V}^4 \mid \forall \phi \lhd \psi \in \Pi : v'(\phi) \geq_i v(\psi)\}$$

Example 7. Let $\Pi = \{p \vee s \lhd \top; \neg s \lhd p\}$ formulated over the signature $\Sigma = \{p, s\}$. We have the following interpretations and corresponding outcomes of the Γ_Π-function:

v	$\Gamma_\Pi(v)$	v	$\Gamma_\Pi(v)$
UU	{UT, TU}	FU	{UT}
UT	{UT, TU}	FT	{UT}
UF	{UT, TU}	FF	{UT}
UI	{UT, TU}	FI	{UT}
TU	{TF, FI}	IU	{TI, FI}
TT	{TF, FI}	IT	{TI, FI}
TF	{TF, FI}	IF	{TI, FI}
TI	{TF, FI}	II	{TI, FI}

We explain $\Gamma_\Pi(\mathsf{UU})$ as follows: in view of $p \vee s \lhd \top$ and $\mathsf{UU}(\top) = \mathsf{T}$, every interpretation $v' \in \Gamma_\Pi(\mathsf{UU})$ has to assign a truth value at least as informative as T to $p \vee s$, i.e. $v'(p \vee s) \geq_i \mathsf{T}$. Likewise, since $\mathsf{UU}(p) = \mathsf{U}$ and $\neg s \lhd p \in \Pi$, $v' \in \Gamma_\Pi(\mathsf{UU})$ has to set $v'(\neg s) \geq_i \mathsf{U}$, which is trivially the case. The two $\leq_i$-minimal interpretations that satisfy this constraint are: UT and TU.

As a second example, consider FF. Like with UU, every interpretation $v' \in \Gamma_\Pi(\mathsf{FF})$ has to assign $v'(p \vee s) \geq_i \mathsf{T}$. However, since $\mathsf{FF}(p) = \mathsf{F}$ and $\neg s \lhd p \in \Pi$, any $v' \in \Gamma_\Pi(\mathsf{FF})$ has to set $v'(\neg s) \geq_i \mathsf{F}$. UT is the unique $\leq_i$-minimal interpretation satisfying these constraints.

We first notice that Γ_Π is indeed a generalization of the Γ_D-function for ADFs. To show this in a more formally precise manner, we first define the cADF Π_D associated with an ADF D.

Definition 7. Given an ADF $D = (S, L, C)$, we define the *cADF* Π_D *associated with* D as $\Pi_D = \{s \lhd C_s \mid s \in S\}$.

We can now show that for any three-valued interpretation ν, $\Gamma_{\Pi_D}(\nu)$ coincides with $\Gamma_D(\nu)$, i.e. the Γ-function for ADFs coincides with the Γ-function for the associated cADFs for three-valued interpretations.

Proposition 5. For any ADF $D = (S, L, C)$ and any $\nu \in \mathcal{V}^3(S)$, $\Gamma_{\Pi_D}(\nu) = \{\Gamma_D(\nu)\}$.

Proof. Consider an ADF $D = (S, L, C)$ and some $\nu \in \mathcal{V}^3(S)$. $v \in \Gamma_{\Pi_D}$ iff v is among the $\leq_i$-minimal interpretations s.t. $v(s) \geq_i \nu(C_s)$ for every $s \in S$. With Fact 2, $\nu(C_s) = \sqcap_i[\nu]^2(C_s)$ for every $s \in S$. This means that $\Gamma_D(s) = \nu(C_s)$ and thus Γ_D is the unique $\leq_i$-minimal interpretation s.t. $v(s) \geq_i \nu(C_s)$. □

The above result shows that the Γ_Π-function is a direct generalization of the well-studied Γ_D-function known from ADFs. This allows us to define the main semantics of cADFs in terms of (post-)fixpoints of the Γ_Π-functions, just like in the case of ADFs.

With our generalized Γ_Π-function at hand, we can now define the main semantics for cADFs as straightforward generalizations of the ADF-semantics:

Definition 8. Let a cADF Π over At and an interpretation $v \in \mathcal{V}^4(\mathsf{At})$ be given, then:

- v is *admissible* for Π iff there is some $v' \in \Gamma_\Pi(v)$ s.t. $v \leq_i v'$.
- v is *complete* for Π iff $v \in \Gamma_\Pi(v)$.
- v is *preferred* for Π if it is a $\leq_i$-maximal among all admissible interpretation for Π;
- v is *grounded* for Π if it is a $\leq_i$-minimal among all complete interpretation for Π;
- v is a *two-valued model* for Π iff $v \in \mathcal{V}^2(\mathsf{At})$ and v is complete.

Example 8 (Example 7 ctd.)**.** We see that for Π from Example 7, there are two complete interpretations: TF and UT. This can be seen by observing that $\mathsf{TF} \in \Gamma_\Pi(\mathsf{TF})$ and $\mathsf{UT} \in \Gamma_\Pi(\mathsf{UT})$. Since these interpretations are $\leq_i$-incomparable, both interpretations are also grounded. The admissible interpretations are: UU, UT, TU and TF. Thus, UT and TF are also preferred.

Example 9. Let $\Pi = \{b \lhd p, f \lhd b, \neg f \lhd p\}$ formulated over $\Sigma = \{b, f, p\}$ be given. $v_\mathsf{U} = \mathsf{UUU}$ is the unique complete interpretation and thus also grounded. It is also the unique admissible interpretation.

Notice that e.g. TIT is *not* complete, since $\Gamma_\Pi(\mathsf{TIT}) = \{\mathsf{TIU}\}$. The reason for $\Gamma_\Pi(\mathsf{TIT})(p) = \mathsf{U}$ is since there is no acceptance pair $p \lhd \phi \in \Pi$. The intuition is that p is only accepted if we have good information to do so, but no such information is given by any $\phi \lhd \psi \in \Pi$.

It is interesting to note that for $\Pi' = \Pi \cup \{p \lhd p\}$, $\mathsf{TIT} \in \Gamma_{\Pi'}(\mathsf{TIT}) = \{\mathsf{TIU}, \mathsf{TIT}, \mathsf{TIF}\}$.

As can be seen in the example above, if an atom a occurs in no statement of ϕ of any acceptance pair $\phi \lhd \psi \in \Pi$, then $v(a) = \mathsf{U}$ for any admissible or complete interpretation v. However, should this be undesired, one can simply add the acceptance pair $a \lhd a$ for such an atom.

5.2 Design Choices in Γ_Π and Comparison with Γ_D

We now discuss the design choices that had to be made when generalizing the Γ-function from ADFs to cADFs. In particular, given the increase in syntactical expressiveness, we had to generalize Γ_Π as to adequately handle this increased expressiveness semantically.

A first generalization is caused by the fact that statements ϕ of acceptance pairs $\phi \lhd \psi$ are possibly non-atomic formulas. Since Γ_Π contains all interpretations v' that align, for any $\phi \lhd \psi \in \Pi$, the truth value of ϕ with $v(\psi)$, there might now be more than one interpretation v' which achieves this. As a case in point, consider the cADF $\Pi = \{p \vee q \lhd \top\}$, where acceptance of $p \vee q$ (which is required by any $v \in \mathcal{V}^4$, since $v(\top) = \mathsf{T}$ for any $v \in \mathcal{V}^4$) can be guaranteed by any interpretation that satisfies p or q. Therefore, the Γ-function might contain multiple interpretations which all do an equally good job of aligning the truth values of statements ϕ with their respective conditions ψ. Thus, Γ_Π is defined as a *non-deterministic operator* (Pelov and Truszczynski 2004; Heyninck and Arieli 2021), in the sense that a single interpretation v might give rise to a non-singleton set of interpretations $\{v_1, \ldots, v_n\} = \Gamma_\Pi(v)$. In the example above, we have e.g. $\Gamma_\Pi(v) = \{\mathsf{TU}, \mathsf{UT}\}$ for any $v \in \mathcal{V}^4(\{p, q\})$.

A second generalization w.r.t. the Γ-function for ADFs is the fact that alignment of statements ϕ with their corresponding condition ψ cannot always be done in an exact way. In more detail, for ADFs D, alignment by Γ_D of s is always exact, in the sense that $\Gamma_D(v)(s)$ *coincides* with the truth value assigned by $\sqcap_i[v]^2(C_s)$. This is not always possible for cADFs, since we might have conflicting specifications in a cADF. Take for example the cADF $\Pi = \{p \lhd \top; \neg p \lhd \top\}$. Clearly, for any $v \in \mathcal{V}^4(\mathsf{At})$, there exists no $v' \in \mathcal{V}^4(\mathsf{At})$ s.t. $v'(\phi) = v(\psi)$ for every $\phi \lhd \psi$. Indeed, this is one of the reasons we had to move to a four-valued logic, since now we can at least specify an interpretation v' which brings $v'(p)$ and $v'(\neg p)$ in alignment with $v(\top)$, in the sense that $v'(p)$ and $v'(\neg p)$ are at least as informative as $v(\top)$, i.e. $v'(p) \geq_i v(\top)$ and $v'(\neg p) \geq_i v(\top)$ (for any $v \in \mathcal{V}^4(\mathsf{At})$).

5.3 Semantical Properties of cADF-semantics

In this section, we show central semantical results on the semantics of cADFs. In particular, we show some relationships between the semantics, and we show under which conditions admissible, complete, grounded and preferred interpretations are guaranteed to exist.

We start by observing that, just like for ADFs, complete interpretations are admissible:

Proposition 6. Let a cADF Π and a complete interpretation v for Π be given. Then v is admissible.

Proof. Suppose v is complete for Π. Then $v \in \Gamma_\Pi(v)$ and thus $v \leq_i v'$ for some $v' \in \Gamma_\Pi(v)$. □

For showing the existence of admissible and preferred interpretations, it will be useful to limit attention to what we will call *well-formed cADFs*. The main idea is that we want to avoid cADFs Π for which $\Gamma_\Pi(v) = \emptyset$ for some $v \in \mathcal{V}^4(\mathsf{At})$, as occurs in e.g. the following example:

Example 10. $\Pi = \{p \lhd \top, \neg p \lhd \top, p \vee \neg p \lhd p\}$.

v	$\Gamma_\Pi(v)$	v	$\Gamma_\Pi(v)$
T	$\{\mathsf{I}\}$	F	$\{\mathsf{I}\}$
U	$\{\mathsf{I}\}$	I	$\emptyset$

Notice that $\Gamma(\mathsf{I}) = \emptyset$.

Definition 9. A *well-formed cADF* is a cADF Π s.t. $\Gamma_\Pi(v) \neq \emptyset$ for any $v \in \mathcal{V}^4(\mathsf{At})$.

We observe that a syntactic sufficient condition for well-formedness of a cADF Π is to simply require that for every acceptance pair $\phi \lhd \psi \in \Pi$, the statement ϕ is a logically contingent formula. We call such cADFs *unconstrained*:

Definition 10. A cADF Π is *unconstrained* iff for every $\phi \lhd \psi \in \Pi$, ϕ is logically contingent.

We explain the term of *unconstrained* cADF as follows. Notice that an acceptance pair $\phi \lhd \psi$, where ϕ is a tautology or a falsity, can be seen as a constraint, in the sense that it forces ψ to be set to the value of ϕ (i.e. $v(\psi) = \mathsf{T}$ if ϕ is a tautology and $v(\psi) = \mathsf{F}$ if ψ is a falsity) for any complete extension. To see this, observe that $v(\phi) = \mathsf{T}[\mathsf{F}]$ for any $v \in \mathcal{V}^4$ if ϕ is a tautology[falsity]. In particular, for any $v' \in \Gamma_\Pi(v)$, it will hold that $v(\phi) = \mathsf{T}[\mathsf{F}]$. It is quite interesting that the framework naturally allows for the formulation of constraints, but for the development of the meta-theory, it will prove useful to restrict attention to well-formed cADFs. It is an interesting question for future work to see whether *constrained argumentation frameworks* (Coste-Marquis, Devred, and Marquis 2006) can be captured using such constraints.

Proposition 7. Any unconstrained cADF Π is well-formed.

Proof sketch. Suppose that Π is an unconstrained cADF. It can be shown that for every $\phi \lhd \psi \in \Pi$, $v_{\mathsf{I}}(\phi) = \mathsf{I}$. Thus, for every $v' \in \mathcal{V}^4(\mathsf{At})$ there is some $v \in \mathcal{V}^4(\mathsf{At})$ (namely $v = v_{\mathsf{I}}$) s.t. $v(\phi) \geq_i v'(\psi)$ for every $\phi \lhd \psi \in \Pi$. Since $\leq_i$ is well-founded and Π is finite, $\Gamma_\Pi(v') \neq \emptyset$ for any $v' \in \mathcal{V}^4(\mathsf{At})$. □

However, there are well-formed cADFs that are not unconstrained:

Example 11. Consider $\Pi = \{a \vee \neg a \lhd a \vee \neg a\}$. Then clearly, for any $v \in \mathcal{V}^4(\mathsf{At})$, $\Gamma_\Pi(v) = \{\mathsf{T}\}$ (since $\mathsf{U}(a \vee \neg a) = \mathsf{T}$ with Lemma 1).

We now show the first existence result, which states that any well-formed cADF admits admissible interpretations:

Proposition 8. For any well-formed cADF, there exists an admissible interpretation.

Proof. For any well-formed cADF Π, $\Gamma_\Pi(v_{\mathsf{U}}) \neq \emptyset$. Since $v_{\mathsf{U}} \leq_i v$ for any $v \in \mathcal{V}^4(\mathsf{At})$, v_{U} is admissible. □

We immediately obtain an existence result for preferred interpretations:

Corollary 1. For any well-formed cADF, there exists a preferred interpretation.

We now show an existence result for the complete and grounded interpretations. This is done by first showing that Γ_Π satisfies monotonicity under the *Smyth-order* (Smyth 1976). The *Smyth-order* $\preceq_i^S \subseteq \wp(\mathcal{V}^4) \times \wp(\mathcal{V}^4)$ is defined as follows: $\mathcal{V}_1 \preceq_i^S \mathcal{V}_2$ iff for every $v_2 \in \mathcal{V}_2$ there is some $v_1 \in \mathcal{V}_1$ s.t. $v_1 \leq_i v_2$.

Remark 2. Notice that $\preceq_i^S$ is a transitive and reflexive relation over $\wp(\mathcal{V}^4(\mathsf{At}))$. Furthermore, $\preceq_i^S$ is a partial order over the set of $\leq_i$-minimal subsets $\mathcal{V}^4$ (i.e. $\preceq_i^S$ is transitive, reflexive and anti-symmetric over $\wp_{\leq_i}(\mathcal{V}^4(\mathsf{At})) = \{\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At}) \mid \mathcal{V}' = \min_{\leq_i}(\mathcal{V}')\}$).

Proposition 9. For any well-formed cADF Π, Γ_Π is $\preceq_i^S$-monotonic.

Proof. First observe that for any $v_1 \leq_i v_2$ and any $\phi \lhd \psi \in \Pi$, $v_1(\psi) \leq_i v_2(\psi)$. Suppose now that $v' \in \mathcal{V}^4$ s.t. $v'(\phi) \geq_i v_2(\psi)$ for every $\phi \lhd \psi \in \Pi$. Then $v'(\phi) \geq_i v_1(\psi)$ for every $\phi \lhd \psi \in \Pi$. Thus, there is some $v \in \Gamma_\Pi(v_1)$ s.t. $v \leq_i v'$. In particular, this means that for every $v' \in \Gamma_\Pi(v_2)$, there is some $v \in \Gamma_\Pi(v_1)$ s.t. $v \leq_i v'$. □

Proposition 10. For any well-formed cADF Π, there exists a complete interpretation.

Proof. Notice that since $v_{\mathsf{I}} \geq_i v$ for every $v \in \mathcal{V}^4(\mathsf{At})$, $v_{\mathsf{I}} \geq_i v_1$ for any $v_1 \in \Gamma_\Pi(v_{\mathsf{I}})$ (notice that since Π is well-formed, $\Gamma_\Pi(v_{\mathsf{I}}) \neq \emptyset$). Since Γ_Π is $\preceq_i^S$-monotonic with Proposition 9, $\Gamma_\Pi(v_1) \preceq_i^S \Gamma_\Pi(v_{\mathsf{I}})$ for any $v_1 \in \Gamma_\Pi(v_{\mathsf{I}})$. Thus, for any $v_1 \in \Gamma_\Pi(v_{\mathsf{I}})$, there is some $v_2 \in \Gamma_\Pi(v_1)$ s.t. $v_2 \leq_i v_1$. We can use the above line of argument to construct a chain of interpretations $\ldots \leq_i v_n \leq_i v_{n-1} \leq_i \ldots v_2 \leq_i v_1 \leq_i v_0 = v_{\mathsf{I}}$ s.t. for every $1 \leq i < n$, $v_i \in \Gamma_\Pi(v_{i-1})$ and $\Gamma_\Pi(v_i) \preceq_i^S \Gamma_\Pi(v_{i-1})$. Since $\mathcal{V}^4(\mathsf{At})$ is finite, this chain ends, i.e. there some $i \in \mathbb{N}$ s.t. $v_i = v_{i+1}$. Since $v_{i+1} \in \Gamma_\Pi(v_i) = \Gamma_\Pi(v_{i+1})$, v_i is a complete interpretation (notice that $\Gamma_\Pi(v_i) = \Gamma_\Pi(v_{i+1})$ follows from the anti-symmetry of $\preceq_i^S$ over $\wp_{\leq_i}(\mathcal{V}^4(\mathsf{At}))$ (Remark 2) and the fact that $\Gamma_\Pi(v) \in \wp_{\leq_i}(\mathcal{V}^4(\mathsf{At}))$ for any $v \in \mathcal{V}^4(\mathsf{At})$). □

We immediately obtain an existence result for the grounded interpretation as well:

Corollary 2. For every well-formed cADF Π, there exists a grounded interpretation.

Another useful order on $\wp(\mathcal{V}^4) \times \wp(\mathcal{V}^4)$ is the *Hoare-order* $\preceq_i^H$ defined as: $\mathcal{V}_1 \preceq_i^S \mathcal{V}_2$ iff for every $v_1 \in \mathcal{V}_1$ there is some $v_2 \in \mathcal{V}_2$ s.t. $v_1 \leq_i v_2$.

Proposition 11. For every well-formed cADF Π s.t. Γ_Π is $\preceq_i^H$-monotonic, if v is preferred then it is complete.

Property	Condition on Π	Result
$\exists$ of admissible int.	well-formed	Prop. 8
$\exists$ of preferred int.	well-formed	Cor. 1
$\exists$ of complete int.	well-formed	Prop. 10
$\exists$ of grounded int.	well-formed	Cor. 2
preferred $\subseteq$ complete	well-formed & $\preceq_i^H$-monotonic	Prop. 11

Table 1: Summary of results from Section 5.3

Proof. Let a well-formed cADF Π s.t. Γ_Π is $\preceq_i^H$-monotonic be given and consider a preferred interpretation $v \in \mathcal{V}^4(\mathsf{At})$. Suppose towards a contradiction that $v \notin \Gamma_\Pi(v)$. Since v is preferred, it is admissible and thus there is some $v' \in \Gamma_\Pi(v)$ s.t. $v \leq_i v'$. Since $v \notin \Gamma_\Pi(v)$, $v <_i v'$. With $\preceq_i^H$-monotonicity of Γ_Π, we obtain that $\Gamma(v) \preceq_i^H \Gamma(v')$ and thus there is some $v'' \in \Gamma(v')$ s.t. $v' \leq_i v''$. But then v' is admissible, contradicting v being preferred. □

We observe, however, that not every cADF has a $\preceq_i^H$-monotonic Γ_Π function:

Example 12. Let $\Pi = \{p \vee (q \wedge s) \lhd s, p \wedge q \lhd s\}$ over the signature $\{p, q, s\}$. Then $\Gamma_\Pi(\mathsf{UUT}) = \{\mathsf{UTT}, \mathsf{TUU}\}$ and $\Gamma_\Pi(\mathsf{TUT}) = \{\mathsf{TTU}\}$. Since $\mathsf{UUT} \leq_i \mathsf{TUT}$, yet there is no $v \in \Gamma_\Pi(\mathsf{TUT})$ s.t. $\mathsf{UTT} \leq_i v$, we see that Γ_Π is not $\preceq_i^H$-monotonic.

We summarize our results in Table 1.

6 Grounded Interpretations and the Grounded State

One of the crucial properties of ADFs is that a unique grounded interpretation is guaranteed to exist. This property does not generalize to the grounded semantics of cADFs, in view of the indeterminism that cADFs allow to express. As a case in point consider $\Pi = \{p \vee q \lhd \top\}$, which has two $\leq_i$-minimal complete interpretations: v_1 and v_2 with:

$$v_1(p) = \mathsf{T} \quad v_1(q) = \mathsf{U} \quad \text{and} \quad v_2(p) = \mathsf{U} \quad v_2(q) = \mathsf{T}$$

Thus, there might be cADFs that do not have a unique grounded interpretation. This might be seen as problematic, since the grounded interpretation for ADFs can be calculated efficiently and straightforwardly by iterating Γ_D starting from v_U. Since the grounded interpretation v_g is $\leq_i$-minimally complete and unique for ADFs, it approximates any other complete interpretation of the ADF in question (in the sense that $v_g \leq_i v$ for any complete interpretation v). We are now interested in defining a similar concept for cADFs, that is, a unique representation of the $\leq_i$-minimal information expressed by a cADF that can be unambigously obtained by application of Γ_Π and approximates any complete interpretation. This can be done by looking at a set of interpretations instead of a single interpretation. We note that this idea is not new. For example, many well-founded semantics for disjunctive logic programming take up this idea, resulting in a well-founded state (Baral, Lobo, and Minker 1992; Alcântara, Damásio, and Pereira 2005).[2] Accordingly, we will be interested in a *grounded state* $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$ that represents the minimal knowledge entailed by a cADF. This grounded state can be defined as the $\preceq_i^S$-minimal fixpoint of Γ'_Π, a generalization of Γ_Π to sets of interpretations. Γ'_Π is obtained as follows:

Definition 11. Given a cADF Π and $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$:

$$\Gamma'_\Pi(\mathcal{V}') = \min_{\leq_i} \bigcup_{v \in \mathcal{V}'} \Gamma_\Pi(v)$$

The following fact gives an equivalent characterization of Γ'_Π, which avoids the superfluous $\leq_i$-minimization in $\Gamma_\Pi(v)$:

Fact 3. Given a cADF Π and $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$, $\Gamma'_\Pi(\mathcal{V}') =$

$$\min_{\leq_i}\{v' \in \mathcal{V}^4 \mid \exists v \in \mathcal{V}' : \forall \phi \lhd \psi \in \Pi : v'(\phi) \leq v(\psi)\}$$

Proof. Let some cADF Π and $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$ be given. Then $\Gamma'_\Pi(\mathcal{V}') = \min_{\leq_i} \bigcup_{v \in \mathcal{V}'} \Gamma_\Pi(v)$ by definition. By definition of Γ_Π, this means that $\Gamma'_\Pi(\mathcal{V}') = \min_{\leq_i} \bigcup_{v \in \mathcal{V}'} \min_{\leq_i}\{v' \in \mathcal{V}^4 \mid v'(\phi) \geq_i v(\psi) \text{ for every } \phi \lhd \psi \in \Pi\}$. But then $\Gamma'_\Pi(\mathcal{V}') = \min_{\leq_i}\{v' \in \mathcal{V}^4 \mid \exists v \in \mathcal{V}' : \forall \phi \lhd \psi \in \Pi : v'(\phi) \leq v(\psi)\}$. □

Definition 12. Let a cADF Π be given. Then we say $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$ is:

- a *complete state* (for Π) iff $\mathcal{V}' = \Gamma'_\Pi(\mathcal{V}')$.
- a *grounded state* (for Π) iff $\mathcal{V}'$ is a $\preceq_i^S$-minimally complete state (for Π).

Proposition 12. Let a cADF Π be given. Then there exists a unique grounded state which can be obtained by iterating Γ'_Π, starting with v_U.

Proof. We now show that Γ'_Π is a $\preceq_i^S$-monotonic operator over $\wp_{\leq_i}(\mathcal{V}^4(\mathsf{At}))$. For this, define $G_\Pi(v) = \{v' \in \mathcal{V}^4(\mathsf{At}) \mid v'(\phi) \geq_i v(\psi) \text{ for every } \psi \lhd \phi\}$. We first show that $\mathcal{V}_1 \preceq_i^S \mathcal{V}_2$ implies $\bigcup_{v \in \mathcal{V}_1} G_\Pi(v) \preceq_i^S \bigcup_{v \in \mathcal{V}_2} G_\Pi(v)$. Indeed, consider some $v_2 \in \bigcup_{v \in \mathcal{V}_2} G_\Pi(v)$. This means that $v_2(\psi) \geq_i v(\phi)$ for some $v \in \mathcal{V}_2$ and every $\phi \lhd \psi \in \Pi$. Since $\mathcal{V}_1 \preceq_i^S \mathcal{V}_2$, there is some $v' \in \mathcal{V}_1$ s.t. $v' \leq_i v$. Thus, $v'(\psi) \leq_i v(\psi)$ for every $\phi \lhd \psi \in \Pi$ (Proposition 4). Thus, $v_2(\psi) \geq_i v'(\phi)$ for every $\phi \lhd \psi \in \Pi$ and $v_2 \in \bigcup_{v \in \mathcal{V}_2} G_\Pi(v)$. Since $\Gamma'_\Pi(\mathcal{V}_2) \subseteq \bigcup_{v \in \mathcal{V}_2} G_\Pi(v)$, we derive that $\bigcup_{v \in \mathcal{V}_1} G_\Pi(v) \preceq_i^S \Gamma'_\Pi(\mathcal{V}_2)$. In other words, for every $v_2 \in \Gamma'_\Pi(\mathcal{V}_2)$ there is some $v_1 \in \bigcup_{v \in \mathcal{V}_1} G_\Pi(v)$ s.t. $v_1 \leq_i v_2$. Since $\Gamma'_\Pi(\mathcal{V}_1) = \min_{\leq_i} \bigcup_{v \in \mathcal{V}_1} G_\Pi(v)$, it follows that $\Gamma'_\Pi(\mathcal{V}_1) \preceq_i^S \Gamma'_\Pi(\mathcal{V}_2)$.

We now show Γ'_Π admits a $\preceq_i^S$-minimal fixpoint. This fixpoint is constructed by applying Γ'_Π iteratively, starting with v_U (recall $v_\mathsf{U}(a) = \mathsf{U}$ for every $a \in \mathsf{At}$). Since $v_\mathsf{U} \leq_i v$ for any $v \in \mathcal{V}^4(\mathsf{At})$, $v_\mathsf{U} \preceq_i^S \Gamma_\Pi(v_\mathsf{U})$. By the

[2] Some semantics explicitly use the idea of a set of interpretations (Alcântara, Damásio, and Pereira 2005), whereas other semantics are phrased syntactically, resulting in a set of disjunctions (Baral, Lobo, and Minker 1992), which is clearly equivalent to a set of interpretations (see also (Seipel, Minker, and Ruiz 1997).

$\preceq_i^S$-monotonicity of Γ_Π, $\Gamma_\Pi^\alpha(v_{\mathsf{U}}) \preceq_i^S \Gamma_\Pi^\beta(v_{\mathsf{U}})$ for any ordinals α, β with $\alpha \leq \beta$. Since $\mathsf{At}(\Pi)$ is finite, this chain reaches an endpoint, i.e. for some ordinal γ, $\Gamma_\Pi^\gamma(v_{\mathsf{U}}) = \Gamma^{\gamma+1}\Pi(v_{\mathsf{U}})$. Thus, we have shown that $\Gamma_\Pi(v_{\mathsf{U}})$ admits a fixpoint. To show that this fixpoint is the $\preceq_i^S$-minimal fixpoint, consider some $\mathcal{V}' \subseteq \mathcal{V}^4(\mathsf{At})$ s.t. $\Gamma_\Pi(\mathcal{V}') = \mathcal{V}'$. Since $\mathcal{V}' = \Gamma_\Pi(\mathcal{V}') = \min_{\leq_i}(\bigcup_{v \in \mathcal{V}'} G_\Pi(v))$, $\mathcal{V}' \in \wp_{\leq_i}(\mathcal{V}(\mathsf{At}))$. Notice that $v_{\mathsf{U}} \preceq_i^S \mathcal{V}'$. Since Γ_Π is $\preceq_i^S$-monotonic, for any ordinal α, $\Gamma_\Pi^\alpha(v_{\mathsf{U}}) \preceq_i^S \Gamma^\alpha(\mathcal{V}')$. Since $\mathcal{V}^4$ is a fixpoint of Γ_Π, this means $\Gamma_\Pi^\alpha(v_{\mathsf{U}}) \preceq_i^S \mathcal{V}'$ for any ordinal α. In particular this holds for the ordinal γ for which $\Gamma_\Pi^\gamma(v_{\mathsf{U}}) = \Gamma^{\gamma+1}\Pi(v_{\mathsf{U}})$. With the anti-symmetry of $\preceq_i^S$, $\mathcal{V}' = \Gamma_\Pi^\gamma(v_{\mathsf{U}})$ or $\Gamma_\Pi^\gamma(v_{\mathsf{U}}) \prec_i^S \mathcal{V}'$. □

The grounded state is a generalization of the grounded interpretation for ADFs:

Proposition 13. For any ADF D, the grounded state coincides with $\{v\}$, where v is the grounded model of D.

Furthermore, the grounded state approximates any complete interpretation:

Proposition 14. For any cADF Π, where $\mathcal{V}'$ is the grounded state for Π and v is a complete interpretation of Π, we have that: $\mathcal{V}' \preceq_i^S \{v\}$.

We illustrate the construction of the grounded state with an example:

Example 13. Let $\Pi = \{p \vee q \lhd \top, s \lhd p, s \lhd q\}$ over the signature $\{p, q, s\}$. Then we can obtain the grounded state for Π by the following calculation:

- The first iteration is obtained as follows:
$$\Gamma'_\Pi(\{v_{\mathsf{U}}\}) = \{\mathsf{TUU}, \mathsf{UTU}\}.$$
- As a second step we calculate
$$\begin{aligned}\Gamma'_\Pi(\Gamma'_\Pi(v_{\mathsf{U}})) &= \min_{\leq_i}(\Gamma_\Pi(\mathsf{TUU}) \cup \Gamma_\Pi(\mathsf{UTU}))\\ &= \min_{\leq_i}(\{\mathsf{TUT}, \mathsf{UTT}\})\\ &= \{\mathsf{TUT}, \mathsf{UTT}\}.\end{aligned}$$
- As a third step, we calculate
$$\begin{aligned}\Gamma'_\Pi(\Gamma'_\Pi(\Gamma'_\Pi(v_{\mathsf{U}}))) &= \min_{\leq_i}(\Gamma_\Pi(\mathsf{TUT}) \cup \Gamma_\Pi(\mathsf{UTT}))\\ &= \{\mathsf{TUT}, \mathsf{UTT}\}\end{aligned}$$

Since in the third step a fixed point was reached, we see that the grounded state of Π is $\{\mathsf{TUT}, \mathsf{UTT}\}$. We see that the grounded state consists of two interpretations, which both make s true, and either make p or q true.

Remark 3. All semantics defined in this paper have been implemented in Java using the `Tweety`-library. The implementation can be found online.

7 Related Work

To the best of our knowledge, no generalizations of ADFs as we have suggested here have been proposed before. However, *epistemic graphs* (Hunter, Polberg, and Thimm 2020) can be regarded as an orthogonal approach to extend the expressivity of ADFs. There, general propositional formulas are interpreted through a probabilistic semantics (that is not related to ADF semantics), thus yielding an expressive probabilistic and argumentative formalism. Instead, we have a purely qualitative formalism that generalises the original ADF semantics directly.

ADFs have been generalized in other works, in particular as to allow for the handling of weights (Brewka et al. 2018; Bogaerts 2019). As our semantics, they allow for an extension of the set of truth-values $\{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ with other values. In fact, in (Brewka et al. 2018) an instantiation of weighted ADFs using Belnap's four-valued logic is discussed. However, in (Brewka et al. 2018) this results in five truth-values, since in weighted ADFs, the truth-values are always supplemented with an information-theoretic minimum U that is not part of the original set of truth-values. Furthermore, this instantiation uses Belnap's four-valued logic to evaluate complex formulas, which means that tautologies can be both assigned Belnap's inconsistent and incomplete truth-values (but never the external U-value). Finally, weighted ADFs have the same syntax as ADFs, and thus, the syntax of cADFs also generalize the syntax of weighted ADFs.

As a side effect of the semantics of cADFs, we obtain also a four-valued semantics of ADFs and argumentation frameworks. Four-valued semantics for abstract argumentation frameworks have been suggested in (Baroni, Giacomin, and Liao 2015) and studied in (Arieli 2012). In (Arieli 2012) argumentation labellings that map arguments to four truth values, in, out, none and both, are defined. Adjusting notation to our setting by letting T stand for in, F for out, U for none and I for both, we see that such argumentation labellings are nothing less and nothing more than four-valued interpretations over the set of arguments. However, using the translation of argumentation frameworks in ADFs from (Brewka et al. 2013), we do not get an equivalence between p-admissible labellings and admissible interpretations of the translated argumentation frameworks.

Example 14. Consider the argumentation framework $\mathsf{AF} = (\{A, B, C\}, (A, B), (C, B), (C, C), (B, B)\}$. Then the corresponding cADF is given by $\Pi = \{A \lhd \neg B \wedge \neg C; C \lhd \neg C; B \lhd \neg B\}$. It can be checked that FIU is p-admissible[3] for AF, but FIU is not admissible for Π, as $\Gamma_\Pi(\mathsf{FIU}) = \{\mathsf{UUI}\}$.

It remains a question for future work whether the translation of argumentation frameworks in cADFs can be adjusted to avoid this discrepancy.

The logic 4CL we designed as a generalization of the logic $\sqcap_i[v]^2$ underlying ADFs has not been suggested in the literature on many-valued logics, to the best of our knowledge. The semantics of 4CL bears some similarities to that of *generalized possibilistic logic* (Dubois 2012), where a pair of sets of possible worlds is used to represent the information given by a four-valued interpretation. However, the crucial difference is that $[v]^4$ might consist of more than two sets of possible worlds, and thus the logics behave quite differently. For example, in generalized possibilistic logic, there exists

[3] We refer to (Arieli 2012) for definitions of p-admissible labellings.

no model that assigns to p, q and $\neg p \vee \neg q$ a designated truth value, whereas, in 4CL, $v_{\mathsf{I}}(p) = v_{\mathsf{I}}(q) = v_{\mathsf{I}}(\neg p \vee \neg q) = \mathsf{I}$.

8 Conclusion

In this paper, we have defined and studied cADFs, which generalize ADFs and allow for indeterminism, over- and underspecfications. Semantics for cADFs are defined in terms of a Γ-function mapping four-valued interpretations to sets of four-valued interpretations. There remains still a lot of work to be done on cADFs. As a first next step, there are still some semantics that need to be generalized form ADFs to cADFs, in particular the stable semantics. Thereafter, we plan to study the computational complexity and realizability (in the style of (Pührer 2020)) of cADFs. On the basis of these steps, we will then have a clear view of which formalisms can be captured by cADFs. Among the most interesting candidates for such representational results, we have our eyes on disjunctive and propositional logic programming (Minker and Seipel 2002; Ferraris 2005) and logics for nonmonotonic conditionals (Kraus, Lehmann, and Magidor 1990).

Acknowledgements

The research reported here was supported by the Deutsche Forschungsgemeinschaft under grant KE 1413/11-1. We thank the reviewers for helpful comments on a previous version of this paper.

References

Alcântara, J.; Damásio, C. V.; and Pereira, L. M. 2005. A well-founded semantics with disjunction. In *International Conference on Logic Programming*, 341–355. Springer.

Arieli, O. 2012. Conflict-tolerant semantics for argumentation frameworks. In *European Workshop on Logics in Artificial Intelligence*, 28–40. Springer.

Baral, C.; Lobo, J.; and Minker, J. 1992. Generalized disjunctive well-founded semantics for logic programs. *Annals of Mathematics and Artificial Intelligence* 5(2):89–131.

Baroni, P.; Giacomin, M.; and Liao, B. 2015. I don't care, i don't know… i know too much! on incompleteness and undecidedness in abstract argumentation. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*. Springer. 265–280.

Belnap, N. D. 2019. How a computer should think. In *New Essays on Belnap-Dunn Logic*. Springer. 35–53.

Bogaerts, B. 2019. Weighted abstract dialectical frameworks through the lens of approximation fixpoint theory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2686–2693.

Brass, S., and Dix, J. 1995. Disjunctive semantics based upon partial and bottom-up evaluation. In *ICLP*, 199–213.

Brewka, G.; Strass, H.; Ellmauthaler, S.; Wallner, J. P.; and Woltran, S. 2013. Abstract dialectical frameworks revisited. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Brewka, G.; Strass, H.; Wallner, J. P.; and Woltran, S. 2018. Weighted abstract dialectical frameworks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 378–389. Springer.

Coste-Marquis, S.; Devred, C.; and Marquis, P. 2006. Constrained argumentation frameworks. *International Conference on Principles of Knowledge Representation and Reasoning* 6:112–122.

Dubois, D. 2012. Reasoning about ignorance and contradiction: many-valued logics versus epistemic logic. *Soft Computing* 16(11):1817–1831.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI* 77:321–358.

Ferraris, P. 2005. Answer sets for propositional theories. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 119–131. Springer.

Fitting, M. 2006. Bilattices are nice things. *Conference on Self-reference* 53–77.

Heyninck, J., and Arieli, O. 2021. Approximation fixpoint theory for non-deterministic operators and its application in disjunctive logic programming. In *International Conference on Principles of Knowledge Representation and Reasoning*.

Heyninck, J.; Kern-Isberner, G.; Skiba, K.; and Thimm, M. 2019. Interpreting conditionals in argumentative environments. In *NMR 2020 Workshop Notes*, 73.

Hunter, A.; Polberg, S.; and Thimm, M. 2020. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. *Artificial Intelligence* 281:103236.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44(1-2):167–207.

Minker, J., and Seipel, D. 2002. Disjunctive logic programming: A survey and assessment. In *Computational logic: logic programming and beyond*. Springer. 472–511.

Pelov, N., and Truszczynski, M. 2004. Semantics of disjunctive programs with monotone aggregates: an operator-based approach. In *Proceedings of NMR'04*, 327–334.

Polberg, S. 2016. Understanding the abstract dialectical framework. In *European Conference on Logics in Artificial Intelligence*, 430–446. Springer.

Pührer, J. 2020. Realizability of three-valued semantics for abstract dialectical frameworks. *Artificial Intelligence* 278:103198.

Seipel, D.; Minker, J.; and Ruiz, C. 1997. Model generation and state generation for disjunctive logic programs. *The Journal of Logic Programming* 32(1):49–69.

Smyth, M. B. 1976. Powerdomains. In *International Symposium on Mathematical Foundations of Computer Science*, 537–543. Springer.

On the Relation between Possibilistic Logic and Abstract Dialectical Frameworks

Jesse Heyninck[1,2] , **Matthias Thimm**[3] , **Gabriele Kern-Isberner**[1] , **Tjitze Rienstra**[3] , **Kenneth Skiba**[3]

[1]Technische Universität Dortmund, Dortmund, Germany
[2]University of Cape Town and CAIR, South-Africa
[3]University of Koblenz-Landau, Koblenz, Germany

Abstract

Abstract dialectical frameworks (in short, ADFs) are one of the most general and unifying approaches to formal argumentation. As the semantics of ADFs are based on three-valued interpretations, the question poses itself as to whether some and which monotonic three-valued logic underlies ADFs, in the sense that it allows to capture the main semantic concepts underlying ADFs. As an entry-point for such an investigation, we take the concept of *model* of an ADF, which was originally formulated on the basis of Kleene's three-valued logic. We show that an optimal concept of a model arises when instead of Kleene's three-valued logic, possibilistic logic is used. We then show that in fact, possibilistic logic is the most conservative three-valued logic that fulfils this property, and that possibilistic logic can faithfully encode all other semantical concepts for ADFs. Based on this result, we also make some observations on strong equivalence and introduce possibilistic ADFs.

1 Introduction

Formal argumentation is one of the major approaches to knowledge representation. In the seminal paper (Dung 1995), *abstract argumentation frameworks* were conceived of as directed graphs where nodes represent arguments and edges between these nodes represent attacks. So-called *argumentation semantics* determine which sets of arguments can be reasonably upheld together given such an argumentation graph. Various authors have remarked that other relations between arguments are worth consideration. For example, in (Cayrol and Lagasquie-Schiex 2005), *bipolar argumentation frameworks* are developed, where arguments can support as well as attack each other. The last decades saw a proliferation of such extensions of the original formalism of (Dung 1995), and it has often proven hard to compare the resulting different dialects of the argumentation formalisms. To cope with the resulting multiplicity, (Brewka and Woltran 2010; Brewka et al. 2013) introduced *abstract dialectical argumentation* that aims to unify these different dialects. Just like in (Dung 1995), *abstract dialectical frameworks* (in short, ADFs) are directed graphs. In contradistinction to abstract argumentation frameworks, however, in ADFs, edges between nodes do not necessarily represent attacks but can encode *any* relationship between arguments. Such a generality is achieved by associating an *acceptance condition* with each argument, which is a Boolean formula in terms of the parents of the argument that expresses the conditions under which an argument can be accepted. As such, ADFs are able to capture all the major extensions of abstract argumentation and offer a general framework for argumentation based inference.

The semantics of ADFs are based on three-valued interpretations assigning one of three truth values true (T), false (F), and undecided (U) to arguments. Even though in various papers on ADFs, Kleene's three-valued logic is mentioned (Brewka et al. 2013; Polberg, Wallner, and Woltran 2013; Linsbichler 2014), it is not so clear what the exact role of this logic is, or for that matter any other monotonic three-valued logic, in ADFs. In this paper, we make an in-depth investigation of which three-valued logics underlie abstract dialectical frameworks, i.e. which three-valued logics allow to straightforwardly encode all semantical concepts used in ADFs. The entry point of this investigation is the notion of a *model of an ADF*, which was mentioned in (Brewka et al. 2013) but barely considered afterwards. We show that, in contradistinction to a claim made by (Brewka et al. 2013), the notion of a model of an ADF as based on Kleene's three-valued logic is ill-conceived, in the sense that it does not form a generalization of the set of admissible interpretations. We then investigate on which logics a sound notion of model can be based, and show that no truth-functional three-valued logic using an involutive negation allows to formulate an adequate concept of model for an ADF. Furthermore, we show that possibilistic logic (Dubois and Prade 1998) is able to provide an adequate notion of model. In fact, this is the least informative logic to provide such a notion. Possibilistic logic can therefore be viewed as a monotonic base logic underlying ADFs. Based on this observation, we characterize strong equivalence of ADFs and we generalize the semantics of ADFs to allow for *possibility distributions* as generalized three-valued interpretations as a basic semantic unit for ADFs. We illustrate the fruitfulness of this generalization by allowing for possibilistic constraints on arguments.

Outline of this paper: We first state all the necessary preliminaries in Section 2 on propositional logic (Sec. 2.1), three-valued logics (Sec. 2.2), possibility theory (Sec.2.3) and ADFs (Sec. 2.4). In Section 3 we answer the question which logics underlie ADFs, by first recalling and generalizing the notion of model for an ADF (Sec. 3.1), then showing that possibilistic logic underlies ADFs in Section 3.2 and thereafter making a study of the relation between truth-functional three-valued logics and ADFs, starting with some general observations (Sec. 3.3) before moving to more specific results on three-valued logics using an involutive, paraconsistent and intuitionistic negation. Thereafter, we use the fact that possibilistic logic underlies ADFs to draw some consequences on strong equivalence for ADFs (Sec. 4) and generalize the semantics of ADFs to allow for possibility distributions instead of three-valued interpretations as a basic semantic unit, allowing for a generalization of ADFs we call *possibilistic ADFs* in Sec. 5. Related work is discussed in Sec. 6 and in Sec. 7 the paper is concluded.

2 Preliminaries

In this section the necessary preliminaries on propositional logic (Section 2.1), three-valued logics (Section 2.2), possibility theory (Section 2.3), and abstract dialectical argumentation (Section 2.4) are introduced.

2.1 Propositional logic

For a set At of atoms let $\mathcal{L}(\mathsf{At})$ be the corresponding propositional language constructed using the usual connectives $\wedge$ (*and*), $\vee$ (*or*), and $\neg$ (*negation*). A (classical) *interpretation* (also called *possible world*) ω for a propositional language $\mathcal{L}(\mathsf{At})$ is a function $\omega : \mathsf{At} \to \{\mathsf{T}, \mathsf{F}\}$. Let $\Omega(\mathsf{At})$ denote the set of all interpretations for At. $\mathsf{At}(\phi)$ is the set of all atoms used in a formula $\phi \in \mathcal{L}(\mathsf{At})$. We simply write Ω if the set of atoms is implicitly given. An interpretation ω *satisfies* (or is a *model* of) an atom $a \in \mathsf{At}$, denoted by $\omega \models a$, if and only if $\omega(a) = \mathsf{T}$. The satisfaction relation $\models$ is extended to formulas as usual.

As an abbreviation we sometimes identify an interpretation ω with its *complete conjunction*, i. e., if $a_1, \ldots, a_n \in \mathsf{At}$ are those atoms that are assigned T by ω and $a_{n+1}, \ldots, a_m \in \mathsf{At}$ are those propositions that are assigned F by ω we identify ω by $a_1 \ldots a_n \overline{a_{n+1}} \ldots \overline{a_m}$ (or any permutation of this).

For $\Phi \subseteq \mathcal{L}(\mathsf{At})$ we also define $\omega \models \Phi$ if and only if $\omega \models \phi$ for every $\phi \in \Phi$. Define the set of models $[X] = \{\omega \in \Omega(\mathsf{At}) \mid \omega \models X\}$ for every formula or set of formulas X. A formula or set of formulas X_1 *entails* another formula or set of formulas X_2, denoted by $X_1 \vdash_{\mathsf{PL}} X_2$, if $[X_1] \subseteq [X_2]$.

2.2 Three-valued logics

A 3-valued interpretation for a set of atoms At is a function $v : \mathsf{At} \to \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$, which assigns to each atom in At either the value T (true, accepted), F (false, rejected), or U (unknown). The set of all three-valued interpretations for a set of atoms At is denoted by $\mathcal{V}(\mathsf{At})$. A 3-valued interpretation v can be extended to arbitrary propositional formulas over At using various *logic systems* L. Therefore, we will, given an interpretation $v \in \mathcal{V}(\mathsf{At})$, denote the truth-value assigned by a logic system L to a formula ϕ as $v^{\mathsf{L}}(\phi)$.[1] Thus, a logic system L is defined as a function assigning a truth value to every formula-interpretation-pair. The (three-valued) models of a formula $\phi \in \mathcal{L}(\mathsf{At})$ for a logic system L are defined as $\mathcal{V}^{\mathsf{L}}(\phi) = \{v \in \mathcal{V}(\mathsf{At}) \mid v^{\mathsf{L}}(\phi) = \mathsf{T}\}$.[2] A consequence relation $\vdash_{\mathsf{L}} \subseteq \wp(\mathsf{L}(\mathsf{At})) \times \mathsf{L}(\mathsf{At}))$ can then be defined as usual by setting $\Gamma \vdash_{\mathsf{L}} \phi$ iff $\mathcal{V}^{\mathsf{L}}(\phi) \supseteq \bigcap_{\gamma \in \Gamma} \mathcal{V}^{\mathsf{L}}(\gamma)$. Thus, a logic system $\mathsf{L} : \mathcal{V}(\mathsf{At}) \times \mathcal{L}(\mathsf{At}) \to \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ gives rise to a consequence relation which is most commonly associated with a logic, and we shall therefore often refer to logic systems as simply *logics*.

A particular useful class of logics are truth-functional logics:

Definition 1. We say a three-valued logic L is *truth-functional* for an n-ary connective $*$, if for every $\phi_1, \ldots, \phi_n, \phi'_1, \ldots, \phi'_n \in \mathcal{L}(\mathsf{At})$, $v^{\mathsf{L}}(\phi_i) = v^{\mathsf{L}}(\phi'_i)$ for every $1 \leq i \leq n$ implies $v^{\mathsf{L}}(*(\phi_1, \ldots, \phi_n)) = v^{\mathsf{L}}(*(\phi'_1, \ldots, \phi'_n))$.

We also introduce a rather weak notion of relevance, which expresses that the truth-value of atoms not occurring in a formula ϕ should not have any impact on the truth-value assigned by L to that formula ϕ.

Definition 2. A logic L satisfies *relevance* iff for any $\phi \in \mathcal{L}(\mathsf{At})$ and $s \in \mathsf{At}$, if $s \notin \mathsf{At}(\phi)$ then for any $v_1, v_2 \in \mathcal{V}(\mathsf{At})$, $v_1(s') = v_2(s')$ for any $s' \in \mathsf{At} \setminus \{s\}$ implies $v_1^{\mathsf{L}}(\phi) = v_2^{\mathsf{L}}(\phi)$.

This notion of relevance is very similar to the property of *independence* (Kern-Isberner, Beierle, and Brewka 2020). Notice that any truth-functional logic satisfies relevance.

We assume two commonly-used orders $\leq_i$ and $\leq_{\mathsf{T}}$ over $\{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$. $\leq_i$ is obtained by making U the minimal element: $\mathsf{U} <_i \mathsf{T}$ and $\mathsf{U} <_i \mathsf{F}$ and this order is lifted pointwise as follows (given two valuations v, w over At): $v \leq_i w$ iff $v(s) \leq_i w(s)$ for every $s \in \mathsf{At}$. $\leq_{\mathsf{T}}$ is defined by $\mathsf{F} \leq_{\mathsf{T}} \mathsf{U} \leq_{\mathsf{T}} \mathsf{T}$ and can be lifted pointwise in a similar fashion.

It will sometimes prove useful to compare logics w.r.t. their *conservativeness*:

Definition 3. Given two logics L and L', L *is at least as conservative than* L' iff for every $\phi \in \mathcal{L}(\mathsf{At})$ and every $v \in \mathcal{V}(\mathsf{At})$, $v^{\mathsf{L}}(\phi) \leq_i v^{\mathsf{L}'}(\phi)$.

As an example, we consider Kleene's logic K.

Kleene's Logic K A 3-valued interpretation v can be extended to arbitrary propositional formulas over At via Kleene semantics (Kleene et al. 1952):

1. $v^{\mathsf{K}}(\neg\phi) = \mathsf{F}$ iff $v^{\mathsf{K}}(\phi) = \mathsf{T}$, $v^{\mathsf{K}}(\neg\phi) = \mathsf{T}$ iff $v^{\mathsf{K}}(\phi) = \mathsf{F}$, and $v^{\mathsf{K}}(\neg\phi) = \mathsf{U}$ iff $v^{\mathsf{K}}(\phi) = \mathsf{U}$;
2. $v^{\mathsf{K}}(\phi\wedge\psi) = \mathsf{T}$ iff $v^{\mathsf{K}}(\phi) = v^{\mathsf{K}}(\psi) = \mathsf{T}$, $v^{\mathsf{K}}(\phi\wedge\psi) = \mathsf{F}$ iff $v^{\mathsf{K}}(\phi) = \mathsf{F}$ or $v^{\mathsf{K}}(\psi) = \mathsf{F}$, and $v^{\mathsf{K}}(\phi \wedge \psi) = \mathsf{U}$ otherwise;
3. $v^{\mathsf{K}}(\phi\vee\psi) = \mathsf{T}$ iff $v^{\mathsf{K}}(\phi) = \mathsf{T}$ or $v^{\mathsf{K}}(\psi) = \mathsf{T}$, $v^{\mathsf{K}}(\phi\vee\psi) = \mathsf{F}$ iff $v^{\mathsf{K}}(\phi) = v^{\mathsf{K}}(\psi) = \mathsf{F}$, and $v^{\mathsf{K}}(\phi \vee \psi) = \mathsf{U}$ otherwise.

Proposition 1. Kleene's Logic K is truth-functional and satisfies semantic relevance.[3]

[1] Notice that $v^{\mathsf{L}}(\alpha) = v^{\mathsf{L}'}(\alpha)$ for any $\alpha \in \mathsf{At}$ and any two three-valued logics L and L'.

[2] Notice that we assume that T is the only designated value. In e.g. paraconsistent logics, also U is taking as a second designated value. However, we stick to the orthodoxy for ADFs and interpret the third truth-value U as "unknown" and therefore not designated.

[3] This follows immediately from the fact that Kleene's logic is *truth-compositional* as defined in e.g. (Chemla and Égré 2019).

2.3 Possibility theory and possibilistic logic

In this subsection, we introduce all necessary preliminaries from possibility theory and possibilistic logic. For more elaborate introductions to possibility theory, we refer the reader to (Dubois and Prade 1993).

Preliminaries from possibility theory Given a set of atoms At, a *possibility distribution* is a mapping $\pi : \Omega(\mathsf{At}) \to [0,1]$. We denote the set of possibility distributions over At by $\mathbf{P}(\mathsf{At})$. π is *normal* if there is some $\omega \in \Omega(\mathsf{At})$ s.t. $\pi(\omega) = 1$. A possibility distribution can be compared using the *principle of minimum specificity* (Dubois and Prade 1986):

Definition 4. Given two possibility distributions π and π', $\pi \leq_s \pi'$ iff $\pi(\omega) \leq \pi'(\omega)$ for every $\omega \in \Omega(\mathsf{At})$.

A possibility distribution induces two measures or degrees that say something about formulas, the *possibility degree* $\Pi_\pi : \mathcal{L}(\mathsf{At}) \to [0,1]$ and the *necessity degree* $\mathcal{N}_\pi : \mathcal{L}(\mathsf{At}) \to [0,1]$. They are defined as follows:

Definition 5. Given a possibility distribution π and a formula $\phi \in \mathcal{L}(\mathsf{At})$:

- $\Pi_\pi(\phi) = \sup\{\pi(\omega) \mid \omega \models \phi\}$.
- $\mathcal{N}_\pi(\phi) = 1 - \Pi_\pi(\neg\phi) = \inf\{1 - \pi(\omega) \mid \omega \models \neg\phi\}$.

Possibilistic logic In (Dubois and Prade 1998), a three-valued logic inspired by possibility theory is presented which is based on defining lower and upper bounds of the evaluation of a formula using a *possibility* and a *necessity measure*. In more detail, given a three-valued interpretation v over At, the set of two-valued interpretations extending a valuation v is defined as $[v]^2 = \{w \in \Omega(\mathsf{At}) \mid v \leq_i w\}$.[4]

Definition 6. Given $v \in \mathcal{V}(\mathsf{At})$, the *necessity measure* $\mathcal{N}_v$ and the *possibility measure* Π_v based on v are functions : $\mathcal{N}_v : \mathcal{L}(\mathsf{At}) \to \{\mathsf{T},\mathsf{F}\}$ and $\Pi_v : \mathcal{L}(\mathsf{At}) \to \{\mathsf{T},\mathsf{F}\}$

$$\Pi_v(\phi) = \begin{cases} \mathsf{T} & \text{iff } \omega \models \phi \text{ for some } \omega \in [v]^2 \\ \mathsf{F} & \text{otherwise} \end{cases}$$

$$\mathcal{N}_v(\phi) = \begin{cases} \mathsf{T} & \text{iff } \omega \models \phi \text{ for every } \omega \in [v]^2 \\ \mathsf{F} & \text{otherwise} \end{cases}$$

We can now derive a three-valued evaluation $v^{\mathsf{poss}} : \mathcal{L}(\mathsf{At}) \to \{\mathsf{T},\mathsf{F},\mathsf{U}\}$ by stating that:[5]

$$v^{\mathsf{poss}}(\phi) = \begin{cases} \mathsf{T} & \text{iff } \mathcal{N}_v(\phi) = \mathsf{T} \\ \mathsf{U} & \text{iff } \mathcal{N}_v(\phi) = \mathsf{F} \text{ and } \Pi_v(\phi) = \mathsf{T} \\ \mathsf{F} & \text{iff } \mathcal{N}_v(\phi) = \Pi_v(\phi) = \mathsf{F} \end{cases}$$

Example 1. Consider the interpretation v over $\{a,b\}$ with $v(a) = v(b) = \mathsf{U}$. Notice that $\mathcal{N}_v(a \vee \neg a) = \mathsf{T}$ and thus $v^{\mathsf{poss}}(a \vee \neg a) = \mathsf{T}$. However, $\mathcal{N}_v(a \vee b) = \mathcal{N}_v(\neg a) = \mathsf{F}$ and $\Pi_v(a \vee b) = \Pi_v(\neg a) = \mathsf{T}$. Thus, even though $v(a) = v^{\mathsf{poss}}(\neg a) = v(b) = \mathsf{U}$, $v^{\mathsf{poss}}(a \vee b) \neq v^{\mathsf{poss}}(a \vee \neg a)$.

Proposition 2. poss is not truth-functional but satisfies *relevance*.

Remark 1. It can be seen that the possibility and necessity measures given a three-valued interpretation v defined in Definition 6 are particular cases of possibility and necessity measures given a possibility distribution π. In more detail, given an interpretation v, set $\pi_v(\omega) = 1$ if $\omega \in [v]^2$ and $\pi_v(\omega) = 0$ otherwise. Then $\Pi_v(\phi) = \mathsf{T}[\mathsf{F}]$ iff $\Pi_{\pi(v)} = 1[0]$ and $\mathcal{N}_v(\phi) = \mathsf{T}[\mathsf{F}]$ iff $\mathcal{N}_{\pi(v)} = 1[0]$. We call the set of possibility distributions $\pi : \Omega(\mathsf{At}) \to \{0,1\}$ the set of *binary possibility distributions*. Clearly, the set of normal binary possibility distributions coincides with $\{\pi_v \mid v \in \mathcal{V}(\mathsf{At})\}$.

2.4 Abstract dialectical frameworks

We briefly recall some technical details on ADFs following loosely the notation from (Brewka et al. 2013). An ADF D is a tuple $D = (\mathsf{At}, L, C)$ where At is a set of statements, $L \subseteq \mathsf{At} \times \mathsf{At}$ is a set of links, and $C = \{C_s\}_{s \in \mathsf{At}}$ is a set of total functions $C_s : 2^{par_D(s)} \to \{\mathsf{T},\mathsf{F}\}$ for each $s \in \mathsf{At}$ with $par_D(s) = \{s' \in \mathsf{At} \mid (s',s) \in L\}$ (also called acceptance functions). An acceptance function C_s defines the cases when the statement s can be accepted (truth value T), depending on the acceptance status of its parents in D. By abuse of notation, we will often identify an acceptance function C_s by its equivalent *acceptance condition* which models the acceptable cases as a propositional formula. We denote by $\mathfrak{D}(\mathsf{At})$ the set of all ADFs which can be formulated on the basis of At.

Example 2. We consider the following ADF $D_1 = (\{a,b,c\}, L, C)$ with $L = \{(a,b),(b,a),(a,c),(b,c)\}$ and:
$C_a = \neg b \quad C_b = \neg a \quad C_c = \neg a \vee \neg b$
Informally, the acceptance conditions can be read as "a is accepted if b is not accepted", "b is accepted if a is not accepted" and "c is accepted if a is not accepted or b is not accepted".

An ADF $D = (\mathsf{At}, L, C)$ is interpreted through 3-valued interpretations $v \in \mathcal{V}(\mathsf{At})$. The topic of this paper is which logics can be used to extend v to complex formulas in way that is suited for ADFs. Given a set of valuations $V \subseteq \mathcal{V}$, $\sqcap_i V(s) := v(s)$ if for every $v' \in V$, $v(s) = v'(s)$ and $\sqcap_i V(s) = \mathsf{U}$ otherwise. The characteristic operator is defined by $\Gamma_D(v) : \mathsf{At} \to \{\mathsf{T},\mathsf{F},\mathsf{U}\}$ where $s \mapsto \sqcap_i\{w(C_s) \mid w \in [v]^2\}$. Thus, $\Gamma_D(v)$ assigns to s the truth-value that all two-valued extensions of v assign to the condition C_s of s, if they agree on C_s, and U otherwise.

Definition 7. Let $D = (\mathsf{At}, L, C)$ be an ADF with $v : \mathsf{At} \to \{\mathsf{T},\mathsf{F},\mathsf{U}\}$ an interpretation:

- v is a *2-valued model* iff $v \in \Omega(\mathsf{At})$ and $v(s) = v(C_s)$ for every $s \in \mathsf{At}$.
- v is *admissible* for D iff $v \leq_i \Gamma_D(v)$.
- v is *complete* for D iff $v = \Gamma_D(v)$.
- v is *preferred* for D iff v is $\leq_i$-maximal among the admissible interpretations for D.
- v is *grounded* for D iff v is $\leq_i$-minimal among the complete interpretations for D.

[4] In (Ciucci, Dubois, and Lawry 2014), instead of two-valued interpretations extending a valuation, the notion of *epistemic set* E_v is used, which defined as: $E_v = \{v' \in \Omega \mid v \leq_i v'\}$. It is clear that $E_v = [v]^2$ for any $v \in \mathcal{V}$.

[5] Notice that this enumeration of cases is exhaustive, as for any $v \in \mathcal{V}(\mathsf{At})$ and any $\phi \in \mathcal{L}(\mathsf{At})$, $\mathcal{N}_v(\phi) \leq_{\mathsf{T}} \Pi_v(\phi)$.

We denote by $\mathsf{2mod}(D)$, $\mathsf{admissible}(D)$, $\mathsf{complete}(D)$, $\mathsf{preferred}(D)$, respectively $\mathsf{grounded}(D)$ the sets of 2-valued models and admissible, complete, preferred, respectively grounded interpretations of D.

Example 3 (Example 2 continued)**.** The ADF of Example 2 has three complete models v_1, v_2, v_3 with:
$v_1(a) = \mathsf{T}$ $v_1(b) = \mathsf{F}$ $v_1(c) = \mathsf{T}$
$v_2(a) = \mathsf{F}$ $v_2(b) = \mathsf{T}$ $v_2(c) = \mathsf{T}$
$v_3(a) = \mathsf{U}$ $v_3(b) = \mathsf{U}$ $v_3(c) = \mathsf{U}$

v_3 is the grounded interpretation whereas v_1 and v_2 are preferred interpretations as well as 2-valued models.

3 Logics for ADFs

In this section, we ask the question of which three-valued logics qualify as a logic for ADFs. In particular, given a set of statements At, we will be interested in which logic $\mathcal{V}(\mathsf{At}) \times \mathcal{L}(\mathsf{At}) \rightarrow \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ can be reasonably said to underlie ADFs. We first recall the notion of a model for ADFs as introduced by (Brewka et al. 2013) and show it is flawed, after which we define models parametrized to a logic. In section 3.2, we show that models parametrized to the logic based on possibility-necessity pairs gives rise to a plausible notion of model. Finally, in section 3.3, we show that there are truth-functional logics that give rise to plausible notions of models, but they commit one to assign determinate truth-values to formulas to which poss assigns the undecided truth-value.

3.1 ADF-models

In (Brewka et al. 2013), *models* are defined as follows:

Definition 8. An interpretation v is a model of an ADF $D = (\mathsf{At}, L, C)$ iff $v(s) \neq \mathsf{U}$ implies $v(s) = v^{\mathsf{K}}(C_s)$ for every $s \in \mathsf{At}$.

In (Brewka et al. 2013) we find the following claim: "Note that admissible interpretations (as well as the special cases complete and preferred interpretations to be defined now) are actually three-valued models." The following example shows that this claim does not hold:

Example 4. $D = (\{a, b\}, L, C)$ with $C_a = b \vee \neg b$ and $C_b = b$. Consider the interpretation v with $v(a) = \mathsf{T}$ and $v(b) = \mathsf{U}$. Since $\sqcap_i [v]^2(b \vee \neg b) = \mathsf{T}$ and $\sqcap_i [v]^2(b) = \mathsf{U}$, v is complete. However, $v^{\mathsf{K}}(b \vee \neg b) = \mathsf{U}$ and thus $v(a) \neq v^{\mathsf{K}}(C_a)$, i.e. v is not a model.

One can notice that in (Brewka et al. 2013), Kleene's logic is only used in the definition of models. For all of the other semantics, no reference to Kleene's logic is made. Instead, the Γ_D-operator is used, which makes use of the completions $[v]^2$ of an interpretation v. Thus, models are the only concepts based on Kleene's logic in (Brewka et al. 2013).

We can now generalize the concept of a model by parameterizing it with the underlying logic L as follows:

Definition 9. Given a logic L s.t. $\mathsf{L} : \mathcal{V}(\mathsf{At}) \times \mathcal{L}(\mathsf{At}) \rightarrow \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ and an ADF D, the set of L-models of D is the set $\mathcal{M}^{\mathsf{L}}(D) = \{v \in \mathcal{V} \mid \text{for every } s \in \mathsf{At} \text{ if } v(s) \neq \mathsf{U} \text{ then } v(s) = v^{\mathsf{L}}(C_s)\}$.

A minimal condition on the set of models is that it includes all the admissible models:

Definition 10. A logic L is *admissible-preserving* if $\mathcal{M}^{\mathsf{L}}(D) \supseteq \mathsf{Admissible}(D)$.

Notice that any admissible-preserving logic L also guarantees that $\mathcal{M}^{\mathsf{L}}(D) \supseteq \mathsf{Sem}(D)$ for any $\mathsf{Sem} \in \{\mathsf{Preferred}, \mathsf{Grounded}, \mathsf{Complete}\}$ since for any Sem-interpretation v, v is admissible.

The following result is a central first insight in the class of admissible-preserving logics:

Lemma 1. *A logic L satisfying relevance is admissible-preserving iff $v^{\mathsf{L}}(\phi) \geq_i \sqcap_i [v]^2(\phi)$ for every $v \in \mathcal{V}(\mathsf{At})$ and every $\phi \in \mathcal{L}(\mathsf{At})$.*[6]

3.2 Possibilistic logic preserves admissibility

In this section, we show that possibilistic logic poss underlies ADFs. We first show the following crucial lemma, which show that for any interpretation, v^{poss} is identical to $\sqcap_i [v]^2$, the latter being a central technical notion in the semantics of ADFs.

Lemma 2. *For any $v \in \mathcal{V}(\mathsf{At})$ and any $\phi \in \mathcal{L}(\mathsf{At})$, $\sqcap_i [v]^2(\phi) = v^{\mathsf{poss}}(\phi)$.*

From this Lemma it follows that poss is admissible-preserving:

Proposition 3. *Possibilistic logic poss is admissible-preserving.*

Furthermore, interestingly enough, the set of models of an ADF under the logic poss collapses to the set of admissible interpretations:

Proposition 4. *For any ADF D, $\mathcal{M}^{\mathsf{poss}}(D) = \mathsf{Admissible}(D)$.*

Finally, we notice that the Γ_D-function, which is of central importance to the semantics of ADFs, can be easily captured in possibilistic logic. Indeed, for any ADF $D = (\mathsf{At}, L, C)$, $v \in \mathcal{V}(\mathsf{At})$ and $s \in \mathsf{At}$, $\Gamma_D(v)(s) = v^{\mathsf{poss}}(C_s)$ (this is immediate from Lemma 2). From this, it follows that the set of complete models of an ADF $D = (\mathsf{At}, L, C)$ coincides with the following set of interpretations: $\{v \in \mathcal{V}(s) \mid v(s) = v^{\mathsf{poss}}(C_s) \text{ for every } s \in \mathsf{At}\}$.

Remark 2. We draw some consequences from the results above for the case of *abstract argumentation frameworks* (Dung 1995). An abstract argumentation framework is a tuple $(\mathsf{Args}, \leadsto)$ where Args represents a set of arguments and $\leadsto \subseteq \mathsf{Args} \times \mathsf{Args}$ is an attack relation between arguments. We denote by $A^+ = \{B \in \mathsf{Args} : B \leadsto A\}$ the set of attackers of A. It it shown in (Brewka et al. 2013) that argumentation frameworks can be translated in ADFs as follows: given $(\mathsf{Args}, \leadsto)$, $D(\mathsf{Args}, \leadsto) = (\mathsf{Args}, \leadsto, C)$ where $C_A = \bigwedge_{B \in \mathsf{Args}: B \in A^+} \neg B$. Notice that for any $A \in \mathsf{Args}$, C_A is a conjunction of negated literals. For such formulas, Kleene's logic K and Poss coincide, i.e. $v^{\mathsf{K}}(\phi) = v^{\mathsf{Poss}}(\phi)$ for any ϕ built up solely from negated atoms using $\vee$ and $\wedge$ (Ciucci, Dubois, and Lawry 2014,

[6] In view of spatial restrictions, proofs have been left out, but can be found in an online appendix.

Prop. 4.5). It thus follows that for any argumentation framework $(\mathsf{Args}, \rightsquigarrow)$, v is complete iff $v(A) = v^{\mathsf{K}}(C_A)$ for every $A \in \mathsf{Args}$. Likewise, other classes of formulas for which (the non-truth-functional) poss is equivalent to (the truth-functional) K or to other logics, is useful for classes of ADFs, such as bipolar ADFs (Brewka and Woltran 2010; Diller et al. 2020) and ADFs corresponding to logic programs.

3.3 Truth-functional logics

We have shown in the previous section that possibilistic logic underlies ADFs. However, according to Proposition 2, possibilistic logic is not truth-functional. We might therefore ask whether there are some truth-functional three-valued logics that can be seen as a logic for ADFs. A first observation we make is that for any admissible-preserving three-valued logic (truth-functional or otherwise), either the logic coincides with poss or the logic assigns a determinate truth-value T or F to at least one formula ϕ (relative to at least one interpretation v) for which poss evaluates ϕ to U. In other words, poss is the most conservative logic that is admissible-preserving.

Proposition 5. For any admissible preserving logic L, if there is a $\phi \in \mathcal{L}(\mathsf{At})$ and a $v \in \mathcal{V}(\mathsf{At})$ s.t. $v^{\mathsf{L}}(\phi) \neq v^{\mathsf{poss}}(\phi)$, then L is strictly less conservative than poss.

In the rest of this section, we make some observations on what this means for truth-functional logics. To limit our study to a sensible class of three-valued truth-functional logics, we start by making some assumptions on the evaluation of connectives. Firstly, we will assume that any connective conforms with classical logic to determinate truth values, i.e. for any n-ary connective $*$, if $v \in \Omega(\mathsf{At})$, then $v^{\mathsf{L}}(*(\phi_1, \ldots, \phi_n)) = v^{\mathsf{PL}}(*(\phi_1, \ldots, \phi_n))$. Notice that for a truth-functional logic, this means that for every $v \in \mathcal{V}$, $v^{\mathsf{L}}(\phi_i) \in \{\mathsf{T}, \mathsf{F}\}$ for every $1 \leq i \leq n$, implies $v^{\mathsf{L}}(*(\phi_1, \ldots, \phi_n)) = v'^{\mathsf{PL}}(*(\phi_1, \ldots, \phi_n))$ where $v' \in \Omega(\mathsf{At})$ s.t. $v'(\phi_i) = v(\phi_i)$ for every $1 \leq i \leq n$. For conjunction, negation and disjunction this means that every logic has to conform to the following partial truth-tables:

$\wedge$	F	U	T
F	F		F
U			
T	F		T

$\vee$	F	U	T
F	F		T
U			
T	T		T

$\neg$	
F	T
U	
T	F

The full range of possibilities for filling out the ¬U-cell of the truth-table for negation results in three negations, known as the involutive $\neg^{\mathsf{i}}$, the paraconsistent $\neg^{\mathsf{p}}$ and the intuitionistic $\neg^{\mathsf{c}}$ (c stands for constructive). These negations have the following truth-tables:

$v(\phi)$	$v(\neg^{\mathsf{i}}\phi)$	$v(\neg^{\mathsf{p}}\phi)$	$v(\neg^{\mathsf{c}}\phi)$
T	F	F	F
U	U	T	F
F	T	T	T

We can show that for any truth-functional logic if the logic is admissible-preserving, it is *strictly* less conservative than poss. We notice that this can be shown without making any assumptions on the connectives other than conformity with classical logic.

Proposition 6. No truth-functional logic L at least as conservative as poss is admissible-preserving.

In passing, we notice that poss also uses an involutive negation, which also implies that $\neg$, in contradistinction to $\vee$ and $\wedge$, is a truth-functional connective in poss.

Fact 1. $\neg$ is a truth-functional, involutive negation under poss.

In the rest of this section, we will further look at which truth-functional logics are exactly admissible-preserving (even though they are strictly less conservative than poss). We shall follow (Ciucci and Dubois 2013) and assume some very basic properties of conjunction, namely (1) $\leq_{\mathsf{T}}$-monotonicity (i.e. if $\mathsf{X} \leq_{\mathsf{T}} \mathsf{Y}$ then $\mathsf{X} \wedge \mathsf{Z} \leq_{\mathsf{T}} \mathsf{X} \wedge \mathsf{Z}$ and $\mathsf{Z} \wedge \mathsf{X} \leq_{\mathsf{T}} \mathsf{Z} \wedge \mathsf{Y}$ for any $\mathsf{X}, \mathsf{Y}, \mathsf{Z} \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$) and (2) Symmetry (i.e. $\mathsf{U} \star \mathsf{T} = \mathsf{T} \star \mathsf{U}$). This results in the following partial truth-table:

$\wedge$	F	U	T
F	F	F	F
U	F		
T	F		T

In the rest of this section, we determine which truth-functional logics with a conjunction as defined above are admissible preserving, by systematically studying all options for the cells $\mathsf{U} \wedge \mathsf{U}$, $\mathsf{U} \wedge \mathsf{T}$ and $\mathsf{T} \wedge \mathsf{U}$.

Involutive negation We show that no truth-functional logic based on an involutive negation is admissible-preserving. Intuitively, the reason is that any such logic is strictly more conservative than poss. A particularly relevant example of this is a tautology like $a \vee \neg a$, which is evaluated to U by any truth-functional logic based on an involutive negation if $v(a) = \mathsf{U}$.

Proposition 7. There exists no truth-functional logic L with an involutive negation that is admissible-preserving.

Paraconsistent negation When we look at truth-functional logics using a paraconsistent negation (and a $\leq_{\mathsf{T}}$-monotonic conjunction conformant with classical logic), a logic can only be admissible-preserving if it makes use of the conjunction known as Bochvar's external conjunction (Bochvar and Bergmann 1981) and which we denote by $\wedge^{\mathsf{B}}$. As disjunction, we use $\vee^1$ (defined below). These connectives have the following truth-tables:

$\wedge^{\mathsf{B}}$	F	U	T
F	F	F	F
U	F	F	F
T	F	F	T

$\vee^1$	F	U	T
F	F	U	T
U	U	U	T
T	T	T	T

The main theorem of this section expresses that there exists a truth-functional three-valued logic using a paraconsistent negation that is admissible-preserving, but it is strictly less conservative than poss. Notice that the fact that this logic is strictly less conservative than poss follows immediately from Proposition 6: the main positive result here is that there exists a truth-functional three-valued logic using a paraconsistent negation that is admissible-preserving. Since the goal of Proposition 8 is to show merely that an admissible-preserving logic based on paraconsistent negation exists, no particular motivation for the choice of conjunction and disjunction is needed, besides the fact that it fulfils some basic properties like $\geq_{\mathsf{T}}$-monotonicity and symmetry (and similarly for Proposition 9).

Proposition 8. $\mathsf{L}^{\neg^{\mathsf{P}},\wedge^{\mathsf{B}},\vee^{1}}$ is admissible-preserving and strictly less conservative than poss.

Intuitionistic negation For an intuitionistic negation, we can show similarly to the previous section that there is a logic which is admissible-preserving (but again less conservative than poss). With regards to disjunction, note that conformity with v^{poss} requires that $v(\mathsf{U} \vee \mathsf{F}) = v(\mathsf{F} \vee \mathsf{U}) = \mathsf{T}$ to ensure that e.g. $v^{\mathsf{L}}(a \vee \neg a) = \mathsf{T}$ even when $v(a) = \mathsf{U}$. The other cells of the truth-table for disjunction can then be filled in using conformity to classical logic and left- and right-monotonicity. We shall use here the conjunction known as Sette's conjunction (Sette 1973). This is, in fact, not the only conjunction that can be used (even though $\wedge^{\mathsf{B}}$ would not result in an admissible-preserving logic).[7] The truth-tables for $\wedge^{\mathsf{S}}$ is written out below. We shall use for a disjunction $\vee^{2}$ as defined below:

$\wedge^{\mathsf{S}}$	F	U	T
F	F	F	F
U	F	T	T
T	F	T	T

$\vee^{2}$	F	U	T
F	F	T	T
U	T	T	T
T	T	T	T

We can now show the main result of this section:

Proposition 9. $\mathsf{L}^{\neg^{\mathsf{c}},\wedge^{\mathsf{S}},\vee^{2}}$ is admissible preserving and strictly less conservative than poss.

4 Strong equivalence

Strong equivalence (Lifschitz, Pearce, and Valverde 2001) is a notion of equivalence for non-monotonic formalisms which states that two knowledge bases (in this case, ADFs) are strongly equivalent if after the addition of any new information, the knowledge bases are equivalent (i.e. the semantics coincide). On the basis of our characterisation results in Section 3.2, one might expect to derive characterisations of strong equivalence for ADFs. After all, in Section 3.2 we have shown that possibilistic logic is a logic underlying abstract dialectical argumentation. Indeed, our results can be used to derive a characterisation of strong equivalence for ADFs. In more detail, we show that strong equivalence for ADFs coincides with pairwise equivalence of acceptance conditions under classical logic. Given our results from Section 3.2, this is not surprising, as equivalence under classical logic coincides with possibilistic logic:

Proposition 10. For any $\phi, \psi \in \mathcal{L}(\mathsf{At})$, $\mathcal{V}^{\mathsf{poss}}(\phi) = \mathcal{V}^{\mathsf{poss}}(\psi)$ iff ϕ and ψ are PL-equivalent (i.e. $[\phi] = [\psi]$).

We first elucidate the concept of strong equivalence for ADFs in more detail. Recall that a central concept in the definition of strong equivalence is *the addition of knowledge*. For many formalisms, addition of knowledge can be modelled using set-theoretic union. For ADFs, this is not feasible for several reasons. Firstly, simply combining two ADFs under set-theoretic union does, rather evidently, not result in a new ADF but rather in a set of ADFs. Secondly, one has to ensure that one models appropriately the combination of two ADFs with shared atoms. Consider e.g. two ADFs $D_1 = (\{a\}, L_1, C_a^1)$ and $D_2 = (\{a\}, L_2, C_a^2)$ with $C_a^1 = a$ and $C_a^2 = \neg a$. Clearly, the combination of ADFs has to be modelled on the basis of some logical operator combining C_a^1 and C_a^2 in a single new condition C_a. We specify a general model of addition of ADFs which allows for the combination of conditions using either disjunction or conjunction. Given a set of atoms At, an *and-or-assignment for* At is a mapping $\odot : \mathsf{At} \to \{\wedge, \vee\}$. Intuitively, an and-or-assignment specifies for every atom $s \in \mathsf{At}$ whether conditions for s will be combined using $\wedge$ or using $\vee$. Based on an and-or-assignment $\odot$, we can now define the combination of two ADFs using $\odot$:

Definition 11. [8] Let $D_1 = (\mathsf{At}_1, L_1, C_1)$ and $D_2 = (\mathsf{At}_2, L_2, C_2)$ be two ADFs and $\odot$ an and-or-assignment for At. Define $D_1 \Cup_{\odot} D_2 = (\mathsf{At}_1 \cup \mathsf{At}_2, L_1 \cup L_2, C^{\odot})$ with and $C^{\odot} = \{C_s^{\odot}\}_{s \in \mathsf{At}}$, where:

$$C_s^{\odot} = \begin{cases} C_s^1 \odot(s) C_s^2 & \text{if } s \in \mathsf{At}_1 \cap \mathsf{At}_2 \\ C_s^1 & \text{if } s \in \mathsf{At}_1 \setminus \mathsf{At}_2 \\ C_s^2 & \text{if } s \in \mathsf{At}_2 \setminus \mathsf{At}_1 \end{cases}$$

Example 5. Consider D as in Example 2, $D' = (\{a, b, d\}, L', C)$ with $C_a = b$, $C_b = d \wedge \neg a$ and $C_d = \neg a$, and $\odot(a) = \odot(b) = \wedge$ and $\odot(c) = \odot(d) = \vee$. Then $D_1 \Cup_{\odot} D_2 = (\{a, b, c, d\}, L_1 \cup L_2, C^{\odot})$ where:

$$\begin{array}{ll} C_a^{\odot} = \neg b \wedge b & C_b^{\odot} = \neg a \wedge d \wedge \neg a \\ C_c^{\odot} = \neg a \vee \neg b & C_d^{\odot} = \neg a \end{array}$$

We now define strong equivalence for ADFs as follows:

Definition 12. Two ADFs $D_1 = (\mathsf{At}, L_1, C_1)$ and $D_2 = (\mathsf{At}, L_2, C_2)$ are strongly equivalent under semantics Sem iff for any $D \in \mathfrak{D}(\mathsf{At})$ and any and-or-assignment $\odot$ for At, $\mathsf{Sem}(D_1 \Cup_{\odot} D) = \mathsf{Sem}(D_2 \Cup_{\odot} D)$.

For any of the admissible, complete, preferred and grounded semantics, pairwise equivalence of conditions under classical logic is a sufficient and necessary condition for strong equivalence:

Proposition 11. Let some $\mathsf{Sem} \in \{\mathsf{Admissible}, \mathsf{Complete}, \mathsf{Preferred}, \mathsf{Grounded}\}$ and two ADFs $D_1 = (\mathsf{At}, L_1, C_1)$ and $D_2 = (\mathsf{At}, L_2, C_2)$ be given. Then: for every $s \in \mathsf{At}$, $C_1^s \equiv_{\mathsf{PL}} C_2^s$ iff D_1 and D_2 are strongly equivalent under semantics Sem.

Interestingly enough, if we restrict the and-or-assignments allowed in combinations of ADFs, our result above does not hold anymore. In particular, for $\oplus \in \{\vee, \wedge\}$, we say that D_1 and D_2 are $\oplus$-strongly

[7] To see this, observe that then e.g. $v(a) = v(b) = \mathsf{U}$ would set $v^{\mathsf{L}}((a \wedge b) \vee (\neg a \wedge b) \vee (a \wedge \neg b) \vee (\neg a \wedge \neg b)) = \mathsf{F}$ even though $v^{\mathsf{poss}}((a \wedge b) \vee (\neg a \wedge b) \vee (a \wedge \neg b) \vee (\neg a \wedge \neg b)) = \mathsf{T}$, contradicting Lemma 1 and the assumption that L is admissible-preserving.

[8] Our notion of composition of ADFs is clearly a generalization of that of (Gaggl and Strass 2014).

equivalent if for any $D \in \mathfrak{D}(\mathsf{At})$ and any and-or-assignment $\odot$ for At for which $\odot(s) = \oplus$ for any $s \in \mathsf{At}$, $\mathsf{Sem}(D_1 \Cup_{\odot} D) = \mathsf{Sem}(D_2 \Cup_{\odot} D)$.

Proposition 12. Let some $\mathsf{Sem} \in \{\mathsf{Admissible}, \mathsf{Complete}, \mathsf{Preferred}, \mathsf{Grounded}\}$ and some $\oplus \in \{\vee, \wedge\}$ be given. Then there exist $\oplus$-strongly equivalent (under Sem) ADFs $D_1 = (\mathsf{At}, L_1, C_1)$ and $D_2 = (\mathsf{At}, L_2, C_2)$ for which for some $s \in \mathsf{At}$, $C_1^s \not\equiv_{\mathsf{PL}} C_2^s$.

Proof. We show the claim for $\odot = \wedge$. Consider the ADFs $D_1 = (\{a, b, c\}, L, C^1)$ and $D_2 = (\{a, b, c\}, L, C^2)$ with:

$$\begin{array}{rcrc} C_a^1 = & \bot & C_a^2 = & \bot \\ C_b^1 = & \bot & C_b^2 = & \bot \\ C_c^1 = & \neg a \wedge b \wedge c & C_c^2 = & a \wedge \neg b \wedge c \end{array}$$

Notice that $C_c^1 \not\equiv_{\mathsf{PL}} C_c^2$. We show that for any $D_3 = (\{a, b, c\}, L, C^3)$, $\mathsf{Admissible}(D_1 \otimes D_3) = \mathsf{Admissible}(D_2 \otimes D_3)$. Indeed, notice first that for any $\phi \in \mathcal{L}(\{a, b, c\})$, any $1 \leq i \leq 2$ and any $x \in \{a, b\}$, $\sqcap[v]^2(C_x^i \wedge \phi) = \mathsf{F}$. Thus, if $v \in \mathsf{Admissible}(D_1 \otimes D_3)$, $v(x) \leq_i \mathsf{F}$ for any $x \in \{a, b\}$. For any such v, $\sqcap[v]^2(\neg a \wedge b \wedge c \wedge \phi) \in \{\mathsf{U}, \mathsf{F}\}$ and $\sqcap[v]^2(a \wedge \neg b \wedge c \wedge \phi) \in \{\mathsf{U}, \mathsf{F}\}$. Thus, for $1 \leq i \leq 2$, if $v \in \mathsf{Admissible}(D_i \otimes D_3)$, $v(c) \leq_i \mathsf{F}$. Suppose now first that $v(c) = \mathsf{U}$. Then $v(c) \leq_i v(C_c^2 \wedge \phi)$ and thus $v \in \mathsf{Admissible}(D_1 \otimes D_3)$. If $v(c) = \mathsf{F}$, then clearly $\sqcap[v]^2(\neg a \wedge b \wedge c \wedge \phi) = \sqcap[v]^2(a \wedge \neg b \wedge c \wedge \phi) = \mathsf{F}$. Otherwise, $\sqcap[v]^2(\neg a \wedge b \wedge c \wedge \phi) \geq_i v(c)$ and $\sqcap[v]^2(a \wedge \neg b \wedge c \wedge \phi) \geq_i v(c)$. Thus, $v \in \mathsf{Admissible}(D_1 \otimes D_3)$ implies $v \in \mathsf{Admissible}(D_2 \otimes D_3)$. By symmetry we obtain $\mathsf{Admissible}(D_1 \otimes D_3) = \mathsf{Admissible}(D_2 \otimes D_3)$. The proof for other semantics is similar.

To show the claim for $\odot = \oplus$, a similar counter-example can be constructed. □

We leave the further investigation of such weaker notions of strong equivalence for future work.

5 ADFs from the perspective of possibility Theory

We now look further into the perspective offered by possibility theory on ADFs. In more detail, based on the strong connection established between ADFs and possibilistic logic (Sec. 3.2), we unpack the semantics of ADFs using concepts known from possibility theory. This will allow us to straightforwardly formulate generalizations of ADFs. We first show how all semantic concepts from abstract dialectical argumentation correspond to notions from possibility theory. Thereafter, we use these correspondences to define *possibilistic ADFs*.

5.1 ADFs interpreted in possibility theory

In this section we interpret the semantics of ADFs in terms of possibility theory, and generalize the semantics of ADFs to possibility distributions.

We start by looking closer at the information ordering. Recall that one interpretation v is less or equally informative than v' iff v' assigns the same determinate truth-value to every atom s for which v assigns a determinate truth-value. It turns out that this is equivalent to requiring that:

$$\mathcal{N}_v(s) \leq \mathcal{N}_{v'}(s) \text{ and } \Pi_v(s) \geq \Pi_{v'}(s) \text{ for every } s \in \mathsf{At}$$

or, equivalently:

$$\Pi_v(\overline{s}) \geq \Pi_{v'}(\overline{s}) \text{ and } \Pi_v(s) \geq \Pi_{v'}(s) \text{ for every } s \in \mathsf{At}$$

Fact 2. For any $v, v' \in \mathcal{V}$, $v \leq_i v'$ iff $\Pi_v(\overline{s}) \geq \Pi_{v'}(\overline{s})$ and $\Pi_v(s) \geq \Pi_{v'}(s)$ for every $s \in \mathsf{At}$.[9]

From this relation, we can derive that $\leq_s$ and $\leq_i$ are each-others converses when we look at three-valued interpretations (or equivalently, normal binary possibility distributions):

Proposition 13. For any interpretations $v, v' \in \mathcal{V}(\mathsf{At})$, $v \leq_i v'$ iff $\pi_{v'} \leq_s \pi_v$.

Based on Fact 2, we can define the information-ordering $\leq_i$ over the set of possibility distributions $\mathbf{P}(\mathsf{At})$ as follows: $\pi \leq_i \pi'$ iff $\Pi_\pi(\overline{s}) \geq \Pi_{\pi'}(\overline{s})$ and $\Pi_\pi(s) \geq \Pi_{\pi'}(s)$ for every $s \in \mathsf{At}$. In other words, more informative possibility distributions assign lower possibility measures to literals. This might seem at first counter-intuitive, when rephrased in terms of the dual necessity measures, the intuition becomes clearer:

$$\pi \leq_i \pi' \text{ iff } \mathcal{N}_\pi(\overline{s}) \leq \mathcal{N}_{\pi'}(\overline{s}) \text{ and } \mathcal{N}_\pi(s) \leq \mathcal{N}_{\pi'}(s)\ \forall s \in \mathsf{At}$$

Proposition 13 only generalizes to the setting of possibility distributions in one direction: indeed $\leq_i$ as defined over possibility distributions is a generalization of the reverse specificity-ordering:

Fact 3. For some possibility distributions $\pi, \pi' \in \mathbf{P}(\mathsf{At})$, $\pi \leq^s \pi'$ implies $\pi' \leq_i \pi$.

The following examples shows that the reverse direction of Proposition 13 does not generalize to the case of arbitrary normal possibility distributions:

Example 6. Consider the following possibility distributions $\pi, \pi' \in \mathbf{P}(\{a, b\})$:

ω	$\pi(\omega)$	$\pi'(\omega)$	ω	$\pi(\omega)$	$\pi'(\omega)$
ab	1	1	$a\overline{b}$	0.1	1
$\overline{a}b$	1	0.1	$\overline{a}\,\overline{b}$	1	1

Notice that $\Pi_\pi(s) = \Pi_{\pi'}(s)$ for any literal s and thus $\pi \leq_i \pi'$ and $\pi' \leq_i \pi$. However, π and π' are $\leq_s$ incomparable, as $\pi(a\overline{b}) \leq \pi'(a\overline{b})$ and $\pi(\overline{a}b) \leq \pi'(\overline{a}b)$. This shows that Proposition 13 does not generalize from $\mathcal{V}(\mathsf{At})$ to $\mathbf{P}(\mathsf{At})$.

We now characterize admissible and complete interpretations in terms of possibility and necessity measures. Admissible interpretations correspond to possibility distributions for which every node s has:

- a degree of necessity equal or less than the degree of necessity of the corresponding condition C_s; and
- a degree of possibility equal or higher than the degree of possibility of the corresponding condition C_s.

[9] Recall that $\leq_s$ is defined in Definition 4.

In other words, the interval formed by the degree of possibility and necessity of C_s is a sub-interval of the correspondent interval for s.

Completeness strengthens this by requiring the necessity degree, respectively the possibility degree, of a node to be equal to the corresponding degree of its condition.

Proposition 14. Given an ADF $D = (\mathsf{At}, L, C)$ and an interpretation $v \in \mathcal{V}(\mathsf{At})$:

1. v is admissible iff for every $s \in \mathsf{At}$, $\mathcal{N}_v(s) \leq \mathcal{N}_v(C_s)$ and $\Pi_v(s) \geq \Pi_v(C_s)$ (or, equivalently $\Pi_v(\neg s) \geq \Pi_v(\neg C_s)$ and $\Pi_v(s) \geq \Pi_v(C_s)$).
2. v is complete iff for every $s \in \mathsf{At}$, $\mathcal{N}_v(s) = \mathcal{N}_v(C_s)$ and $\Pi_v(s) = \Pi_v(C_s)$ (or, equivalently $\Pi_v(\neg s) = \Pi_v(\neg C_s)$ and $\Pi_v(s) = \Pi_v(C_s)$).

We can now straightforwardly generalize the ADF semantics to possibility distributions:

Definition 13. Given an ADF $D = (\mathsf{At}, L, C)$ and a normal possibility distribution $\pi \in \mathbf{P}(\mathsf{At})$:

- π is *admissible (for D)* iff $\Pi_\pi(\neg s) \geq \Pi_\pi(\neg C_s)$ and $\Pi_\pi(s) \geq \Pi_\pi(C_s)$ for every $s \in \mathsf{At}$.
- π is *complete (for D)* iff $\Pi_\pi(\neg s) = \Pi_\pi(\neg C_s)$ and $\Pi_\pi(s) = \Pi_\pi(C_s)$ for every $s \in \mathsf{At}$.
- π is *grounded (for D)* iff π is a $\leq_i$-minimal complete possibility distribution.
- π is *preferred (for D)* iff π is a $\leq_i$-maximal admissible possibility distribution.

We can show that these semantics satisfy the following basic argumentative properties for ADFs:

Proposition 15. Given an ADF $D = (\mathsf{At}, L, C)$: (1) there exists a unique grounded possibility distribution for π; (2) any preferred possibility distribution for π is complete.

The above proposition is shown by defining a function $\mathfrak{G}_D : \mathbf{P}(\mathsf{At}) \to \mathbf{P}(\mathsf{At})$ that returns, for a possibility distribution π, a new possibility distribution $\mathfrak{G}_D(\pi)$ s.t. for any $s \in \mathsf{At}$, $\Pi_{\mathfrak{G}_D}(\pi)(s) = \Pi_\pi(C_s)$ and $\Pi_{\mathfrak{G}_D}(\pi)(\overline{s}) = \Pi_\pi(\overline{C_s})$. To define such a $\mathfrak{G}_D$-function constructively, we need some preliminaries first. Given a set of formulas $\{\phi_1, \ldots, \phi_n\}$ and a possibility measure $\pi \in \mathbf{P}(\mathsf{At})$, we call the *possibility-vector of* $\{\phi_1, \ldots, \phi_n\}$ *given* π the vector $\langle \dot{\phi}_{i_1}, \ldots, \dot{\phi}_{i_1} \rangle$ s.t. for every $1 \leq i \leq n$, ϕ_i and $\overline{\phi_i}$ both occur exactly once in the vector and the vector is arranged w.r.t. ascending degree of possibility, i.e. for $j \leq k$ it holds that $\Pi_\pi(\dot{\phi}_{i_j}) \leq \Pi_\pi(\dot{\phi}_{i_k})$. We can now define the $\mathfrak{G}_D$-function as follows:

Definition 14. Let a possibility distribution $\pi \in \mathbf{P}(\mathsf{At})$, an ADF $D = (\mathsf{At}, L, C)$, and the possibility-vector $\langle \dot{C}_{s_{i_1}}, \ldots, \dot{C}_{s_{i_k}} \rangle$ of $\{C_{s_1}, \ldots, C_{s_n}\}$ given π be given. Then we define $\mathfrak{G}_D(\pi)$ as the possibility distribution s.t. $\mathfrak{G}_D(\pi)(\omega) = \sup_\pi([\dot{C}_{s_{i_j}}])$ for every $\omega \in [\dot{s}_{i_j}] \setminus \bigcup_{l=1}^{j-1} [\dot{s}_{i_l}]$ for every $1 \leq j \leq k$.[10]

Thus, $\mathfrak{G}_D(\pi)$ is constructed iteratively, starting with the literal $\dot{s}$ for which $\Pi_\pi(\dot{C}_s)$ is the lowest among all literals. For all worlds satisfying $\dot{s}$, we set $\mathfrak{G}_D(\pi)(\omega) = \Pi_\pi(\dot{C}_s)$. Then, we take the second element $\dot{s}'$ of the possibility-vector, and proceed similarly for all worlds satisfying $\dot{s}'$ but not satisfying $\dot{s}$. This process is repeated for all elements of the possibility-vector.

Example 7. Let $D_2 = (\{a, b, c\}, L, C)$ with:

$$C_a = \neg b \wedge \neg c \quad C_b = \neg a \quad C_c = c$$

and consider π_1 defined by:

ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$
abc	0.1	$ab\overline{c}$	0.2	$a\overline{b}c$	0.3	$a\overline{b}\overline{c}$	1.0
$\overline{a}bc$	0.3	$\overline{a}b\overline{c}$	0.2	$\overline{a}\overline{b}c$	0.1	$\overline{a}\overline{b}\overline{c}$	0.1

π gives rise to the following possibility measures for acceptance conditions and their negation:

ϕ	C_a	$\neg C_a$	C_b	$\neg C_b$	C_c	$\neg C_c$
$\Pi_\pi(\phi)$	1.0	0.3	0.3	1.0	0.3	1.0

This results in the following possibility-vector for D given π: $\langle \overline{C_a}, C_b, C_c, \overline{C_b}, C_a, \overline{C_c} \rangle$.

Since $\overline{C_a}$ occurs first in the possibility-vector, we set $\mathfrak{G}_D(\pi)(\overline{a}bc) = \mathfrak{G}_D(\pi)(\overline{a}b\overline{c}) = \mathfrak{G}_D(\pi)(\overline{a}\overline{b}c) = \mathfrak{G}_D(\pi)(\overline{a}\overline{b}\overline{c}) = 0.3$. Since $\Pi_\pi(C_b) = \Pi_\pi(C_c)$, we proceed similarly with all worlds that satisfy c or b, i.e. $\mathfrak{G}_D(\pi)(abc) = \mathfrak{G}_D(\pi)(ab\overline{c}) = \mathfrak{G}_D(\pi)(a\overline{b}c) = 0.3$.

Then, we proceed to the next element of the possibility-vector, $\overline{C_b}$, and, since $\Pi_\pi(\overline{C_b}) = 1.0$, we set $\mathfrak{G}_D(\pi)$ for every world that satisfies $\overline{b}$ but does not satisfy $\overline{a}$, c or b (i.e. every element of $[\overline{b}] \setminus ([\overline{a}] \cup [c] \cup [b])$) to 1.0. Thus, $\mathfrak{G}_D(\pi)(a\overline{b}\overline{c}) = 1.0$. Since every world in $\Omega(\mathsf{At})$ has been assigned a value, the construction of $\mathfrak{G}_D(\pi)$ is finished.

The $\mathfrak{G}_D$-function is a faithful generalization of the Γ_D-operator (in view of Remark 1):

Proposition 16. For any ADF D and any three-valued interpretation $v \in \mathcal{V}(\mathsf{At})$, $\Pi_{\Gamma_D(v)}(s) = \mathsf{T}[\mathsf{F}]$ iff $\Pi_{\mathfrak{G}_D(\pi_v)}(s) = 1[0]$ and $\mathcal{N}_{\Gamma_D(v)}(s) = \mathsf{T}[\mathsf{F}]$ iff $\mathcal{N}_{\mathfrak{G}_D(\pi_v)}(s) = 1[0]$.

Thus, the information order, as well as the semantics of ADFs can all be straightforwardly rephrased using possibility measures Π and necessity measures $\mathcal{N}$. On the basis of this interpretation, the semantics for ADFs were generalized from three-valued interpretations – which can be viewed as binary possibility distributions) – to arbitrary possibility distributions. In the next section, we use this generalization to define *possibilistic* ADFs.

5.2 Possibilistic ADFs

We now introduce possibilistic ADFs as a a quantitative extension of ADFs, which can assign a degree of plausibility to the acceptance of nodes. This allows, among others, the incorporation of possibilistic constraints on nodes and their acceptance condition.

Definition 15. An *ADF with possibilistic constraints* (pADF) is a tuple $\mathfrak{D} = (\mathsf{At}, L, C, \rho)$ where (At, L, C) is an ADF and $\rho : \mathsf{At} \to [0, 1]$.

[10] This construction has been implemented in Java using the `Tweety`-library. The implementation can be found online.

The intuitive interpretation of ρ_S is that they form an upper limit on the possibility of the nodes of an pADF.

Example 8. Consider the following pADF:

$$\mathfrak{D} = (\{a, b, c\}, L, \{C_a = \neg b \wedge \neg c, C_b = \neg a, C_c = c\},\\ \{\rho(a) = 1, \rho(b) = 0.8, \rho(c) = 0.4\})$$

Definition 16. Given a pADF $\mathfrak{D} = ((\mathsf{At}, L, C, \rho)$, a normal possibility distribution $\pi : S \to [0, 1]$ is:

- p-*permissible (for* $\mathfrak{D}$*)* iff $\Pi_\pi(s) \leq \rho(s)$ for every $s \in \mathsf{At}$.
- p-*admissible (for* $\mathfrak{D}$*)* iff it is admissible and p-permissible for $\mathfrak{D}$.
- p-*complete (for* $\mathfrak{D}$*)* iff it is complete and p-permissible for $\mathfrak{D}$.
- p-*grounded (for* $\mathfrak{D}$*)* if it is $\leq_i$-least specific p-complete interpretation for $\mathfrak{D}$.
- p-*preferred (for* $\mathfrak{D}$*)* if it is a $\leq_i$-maximal p-admissible interpretation for $\mathfrak{D}$.

Example 9. The following possibility distributions is p-grounded for the pADF $\mathfrak{D}$ from Example 8:

ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$	ω	$\pi_1(\omega)$
abc	0.4	$ab\overline{c}$	0.8	$a\overline{b}c$	0.4	$a\overline{b}\overline{c}$	1
$\overline{a}bc$	0.4	$\overline{a}b\overline{c}$	0.8	$\overline{a}\overline{b}c$	0.4	$\overline{a}\overline{b}\overline{c}$	0.8

The following distributions is p-preferred for $\mathfrak{D}$:

ω	$\pi_2(\omega)$	ω	$\pi_2(\omega)$	ω	$\pi_2(\omega)$	ω	$\pi_2(\omega)$
abc	0	$ab\overline{c}$	0	$a\overline{b}c$	0	$a\overline{b}\overline{c}$	1
$\overline{a}bc$	0	$\overline{a}b\overline{c}$	0	$\overline{a}\overline{b}c$	0	$\overline{a}\overline{b}\overline{c}$	0

Notice that the grounded possibility distribution for $D = (\{s, c\}, L, \{C_s = \neg c, C_c = \neg s\})$ is *not* p-complete for $\mathfrak{D}$. Indeed, the grounded extension for D is given by $\pi_3(\omega) = 1$ for every $\omega \in [\{a, b, c\}]$. To see that π_3 is not p-complete for $\mathfrak{D}$, it suffices to observe that $\Pi_{\pi_3}(b) = 1 > \rho(b) = 0.8$.

We remark here that there might not exist a unique p-grounded extension for a given pADF. Furthermore, there might be pADFs for which there do not exist even p-admissible extensions. For example, if we change $\rho(a) = 0.9$ in the pADF from Example 8 there does exist a normal p-admissible possibility distribution. A pADF for which no p-admissible extensions exist can be seen as faultily specified model. This is not unlike epistemic approaches to probablistic argumentation (Hunter and Thimm 2017), where certain requirements such as coherence w.r.t. an argumentation framework are required in order to ensure a good fit between a probability function and an argumentation framework (Hunter and Thimm 2017). We leave the investigation of such requirements for pADFs for future work.

6 Related work

In this paper, we have investigated three-valued monotonic logics underlying ADFs. To the best of our knowledge, this work is the first systematic such study, but there are some works which contain some similar results or questions. In (Baumann and Heinrich 2020), it is shown that there is no truth-functional three-valued logic L s.t. for every $v \in \mathcal{V}(\mathsf{At})$ and every $\phi \in \mathcal{L}(\mathsf{At})$, $v^{\mathsf{L}}(\phi) = \sqcap_i[v]^2(\phi)$. Lemma 1 is a generalization of this result. Our paper continues where (Baumann and Heinrich 2020) stopped, since we show which truth-functional logics are admissible-preserving, and that there is a non-truth-functional monotonic three-valued logic, poss for which $v^{\mathsf{poss}}(\phi) = \sqcap_i[v]^2(\phi)$ for every $v \in \mathcal{V}(\mathsf{At})$ and every $\phi \in \mathcal{L}(\mathsf{At})$. In (Heyninck and Kern-Isberner 2020) ADFs are translated in autoepistemic logic via epistemic models, which are related to possibilistic logic (Ciucci and Dubois 2012).

With respect to the possibilistic ADFs introduced in this paper, we make a comparison with *weighted ADFs* (Brewka et al. 2018). Weighted ADFs generalize ADFs by allowing interpretations which map nodes to elements of V_{U}, which is a complete partial order constructed on the basis of a chosen set V of values combined with the U-value, which forms the $\leq_i$-least element under the information order over V_{U}. This is a very general model of weighted argumentation, which possibilistic ADFs cannot lay claim to. On the other hand, in possibilistic ADFs, there is no need to postulate an additional value U, since it arises naturally from the possiblistic semantics as a discrepancy between the necessity measure $\mathcal{N}$ and the possibility measure Π. (Wu et al. 2016) defines *fuzzy argumentation frameworks*, where arguments and attacks are assigned a degree of belief. The central concept in this work is the concept of a *tolerable attack* which is an attack such that the belief in the attacked argument is not greater than the composition (according to an appropriate composition operator such as the Gödel t-norm) of the belief in the attacking argument and the belief in the attack. Argumentation semantics can then be defined using this concept of weakening attack. (Janssen, De Cock, and Vermeir 2008) uses a similar semantics. It can be seen that these semantics are dependent on the syntactical structure of argumentation frameworks consisting of arguments and attacks. Furthermore, it should be noticed that even though possiblistic logic is related to fuzzy logic, they are far from equivalent. Among the most poignant differences between these two formalisms in our setting is probably truth-functionality. For example, given the fuzzy degree of belief in two formulas ϕ_1 and ϕ_2, one can exactly determine the fuzzy degree of belief in $\phi_1 \wedge \phi_2$, whereas based on the possibility measure assigned to ϕ_1 and ϕ_2 according to π, one can merely determine an upper bound $min\{\Pi_\pi(\phi_1), \Pi_\pi(\phi_2)\}$ on $\Pi_\pi(\phi_1 \wedge \phi_2)$.

7 Conclusion

In this paper, we have investigated monotonic three-valued logics that underlie abstract dialectical argumentation. The central result is that possibilistic logic is closely related to abstract dialectical argumentation, as it is the most conservative admissible-preserving logic, and allows to straightforwardly codify all central semantical notions from abstract dialectial argumentation. We have also exhaustively investigated the ADF-related properties of truth-functional three-valued logics, showing that truth-functional logics using involutive negation are not admissible-preserving, whereas there exist admissible-preserving truth-functional logics using an intuitionistic or paraconsistent negation, but these are strictly less conservative than possibilistic logic. Furthermore, we have illustrated the fruitfulness of our results by (1) characterising strong equivalence and (2) proposing *possibilistic ADFs*, which allow for quantitative reasoning in ADFs in a way that faithfully generalizes (qualitative) reasoning in ADFs. We believe that the connection between possibilistic logic and possibility theory on the one hand, and (abstract) argumentation and ADFs on the other hand, will provide a useful tool for work argumentation, by providing opportunities for the application of results and insights from possibility theory in argumentation.

Acknowledgements

The research reported here was supported by the Deutsche Forschungsgemeinschaft under grant KE 1413/11-1. We thank Jonas Schumacher for helpful comments.

References

Baumann, R., and Heinrich, M. 2020. Timed abstract dialectical frameworks: A simple translation-based approach. *Computational Models of Argument: Proceedings of COMMA* 103 – 110.

Bochvar, D. A., and Bergmann, M. 1981. On a three-valued logical calculus and its application to the analysis of the paradoxes of the classical extended functional calculus. *History and Philosophy of Logic* 2(1-2):87–112.

Brewka, G., and Woltran, S. 2010. Abstract dialectical frameworks. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2013. Abstract dialectical frameworks revisited. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*.

Brewka, G.; Strass, H.; Wallner, J.; and Woltran, S. 2018. Weighted abstract dialectical frameworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 378–389. Springer.

Chemla, E., and Égré, P. 2019. Suzko's problem: Mixed consequence and compositionality. *The Review of Symbolic Logic* 12(4):736–767.

Ciucci, D., and Dubois, D. 2012. Three-valued logics for incomplete information and epistemic logic. In *European Workshop on Logics in Artificial Intelligence*, 147–159. Springer.

Ciucci, D., and Dubois, D. 2013. A map of dependencies among three-valued logics. *Information Sciences* 250:162–177.

Ciucci, D.; Dubois, D.; and Lawry, J. 2014. Borderline vs. unknown: comparing three-valued representations of imperfect information. *International Journal of Approximate Reasoning* 55(9):1866–1889.

Diller, M.; Keshavarzi Zafarghandi, A.; Linsbichler, T.; and Woltran, S. 2020. Investigating subclasses of abstract dialectical frameworks. *Argument & Computation* 11(1-2):191–219.

Dubois, D., and Prade, H. 1986. The principle of minimum specificity as a basis for evidential reasoning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 75–84.

Dubois, D., and Prade, H. 1993. Fuzzy sets and probability: misunderstandings, bridges and gaps. In *[Proceedings 1993] Second IEEE International Conference on Fuzzy Systems*, 1059–1068. IEEE.

Dubois, D., and Prade, H. 1998. Possibility theory: qualitative and quantitative aspects. In *Quantified representation of uncertainty and imprecision*. Springer. 169–226.

Dung, P. M. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence* 77(2):321–358.

Gaggl, S. A., and Strass, H. 2014. Decomposing abstract dialectical frameworks. In *Computational Models of Argument: Proceedings of COMMA*, 281–292.

Heyninck, J., and Kern-Isberner, G. 2020. An epistemic interpretation of abstract dialectical argumentation. In *Computational Models of Argument: Proceedings of COMMA 2020*, 227 – 238.

Hunter, A., and Thimm, M. 2017. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research* 59:565–611.

Janssen, J.; De Cock, M.; and Vermeir, D. 2008. Fuzzy argumentation frameworks. In *Information processing and management of uncertainty in knowledge-based systems*, 513–520.

Kern-Isberner, G.; Beierle, C.; and Brewka, G. 2020. Syntax splitting= relevance+ independence: New postulates for nonmonotonic reasoning from conditional belief bases. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 560–571.

Kleene, S. C.; De Bruijn, N.; de Groot, J.; and Zaanen, A. C. 1952. *Introduction to metamathematics*, volume 483. van Nostrand New York.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)* 2(4):526–541.

Linsbichler, T. 2014. Splitting abstract dialectical frameworks. In *Computational Models of Argument: Proceedings of COMMA*, 357–368.

Polberg, S.; Wallner, J. P.; and Woltran, S. 2013. Admissibility in the abstract dialectical framework. In *International Workshop on Computational Logic in Multi-Agent Systems*, 102–118. Springer.

Sette, A. M. 1973. On the propositional calculus p1. *Mathematica Japonicae*.

Wu, J.; Li, H.; Oren, N.; and Norman, T. J. 2016. Gödel fuzzy argumentation frameworks. In *Computational Models of Argument: Proceedings of COMMA*, 447–458.

Equivalence Results between *SETAF* and Attacking Abstract Dialectical Frameworks

João Alcântara[1] , **Samy Sá**[2]
[1,2]Universidade Federal do Ceará - Brazil
jnando@lia.ufc.br, samy@ufc.br

Abstract

Abstract Dialectical Frameworks (ADF) and Argumentation Frameworks with collective attacks, called *SETAF*, are two prominent extensions of Dung's Abstract Argumentation Frameworks that have attracted increasing interest in the last years. Previous studies have provided a translation from *SETAF* to ADF. In this work, we propose a new translation from a fragment of ADFs called Attacking Abstract Dialectical Frameworks (ADF^+s) to *SETAF* and prove various equivalences between their semantics, including the equivalence between their complete semantics, grounded semantics, preferred semantics, stable semantics and semi-stable/L-stable semantics. In addition, we show that ADF^+s without redundant links correspond precisely to *SETAF*. Indeed, we prove that the back and forth translations from ADF^+s to *SETAF*s correspond to bijective functions and each other's inverse provided ADF^+s have no redundant links.

1 Introduction

In recent years, *formal argumentation* has attracted an increasing attention among the Artificial Intelligence Community (Rahwan and Simari 2009). In formal argumentation, the reasoning process is characterized by constructing and evaluating arguments, which (roughly speaking) can be understood as sets of reasons for the validity of a claim. In perspective, while in Logic the reasoning process is focused on the inference within an argument (a proof), in argumentation the reasoning process contemplates the interactions between arguments. One particular interest of that approach is, given a possibly inconsistent logical theory, to pinpoint some of its consistent sub-theories.

Amongst the most influential works in this area, Dung's theory of Abstract Argumentation Frameworks (*AAF*s) (Dung 1995) presents arguments as abstract entities and computes semantics based on the conflicts between them. An argumentation semantics specifies criteria according to which sets of mutually acceptable arguments (framework extensions) are returned. Naturally, different criteria of acceptability will lead to different semantics, including Dung's original concepts of complete, stable, preferred and grounded extension-based semantics (Dung 1995) and semi-stable semantics (Verheij 1996; Caminada 2006b). It should be noticed that the semantics for *AAF*s can be equivalently expressed as argument labellings: in (Caminada 2006a; Caminada and Gabbay 2009) each argument is assigned a label, which can be either `in`, `out` or `undec`. Intuitively, an argument labeled `in` is explicitly accepted, while an argument labeled `out` is rejected, and one labeled `undec` is undecided, i.e., neither accepted not rejected. In this work, we will adhere to the labelling-based approach.

Despite their generality, not rarely *AAF*s have been criticised for being overly limiting as the only interaction between arguments is given by the attack relation. Indeed, one could argue that *AAF*s lack certain features which are common in almost every form of argumentation to be found in practice (Brewka and Woltran 2010). As overviewed in (Brewka, Polberg, and Woltran 2014), many proposals generalising *AAF*s can be found in the literature; it includes among others *AAF*s with support relations (Cayrol and Lagasquie-Schiex 2005; Cayrol and Lagasquie-Schiex 2013; Oren and Norman 2008; Polberg and Oren 2014), attacks on attacks (Modgil 2009) and *AAF*s with weights (Martınez, Garcıa, and Simari 2008; Dunne et al. 2011; Coste-Marquis et al. 2012). In the current work, we will focus on two generalisations of *AAF*s: Abstract Dialectical Frameworks (ADFs) (Brewka and Woltran 2010; Brewka et al. 2013) and frameworks with sets of attacking arguments (*SETAF*s) (Nielsen and Parsons 2006).

Abstract Dialectical Frameworks (ADFs) (Brewka and Woltran 2010; Brewka et al. 2013) are among the most comprehensive generalisations of *AAF*s by allowing the expression of arbitrary relationships among arguments. In an ADF, besides the attack relation, arguments may support each other, or a group of arguments may jointly attack another while each single member of the group is not strong enough to do so (Brewka et al. 2017). This additional expressiveness is obtained by associating to each node (argument) a two-valued acceptance condition which can be expressed as an arbitrary propositional formula. The intuition is that an argument is accepted if its associated acceptance condition is verified. In short, we can characterize ADFs as dependency graphs + acceptance conditions.

In (Nielsen and Parsons 2006) it was proposed an extension of Dung's Abstract Argumentation Frameworks(Dung 1995) (*AAF*s) to allow joint attacks on arguments. The resulting framework is called *SETAF*. Whereas in *AAF*s only (individual) arguments can attack another argument, in *SETAF*s sets of arguments can also attack arguments. A translation from *SETAF* to ADF (see (Polberg 2016)) can be obtained by representing each attack from a (finite) set B of arguments to an argument a as the propositional formula $\neg(\bigwedge_{b \in B} b)$ to express the acceptance conditions of a. It has been proved in (Linsbichler, Pührer, and Strass 2016) *SETAF* is strictly less expressive than ADF. A question naturally arises: which fragment of ADF corresponds exactly to *SETAF*?

It is clear ADFs are basic argumentation frameworks with links to many argumentative approaches; they are general enough to express the notions of support, collective attacks, and even more sophisticated relations. However, this comprehensive expressivity comes with a price to pay: a higher computational complexity. Thus, it is natural to investigate subclasses of ADFs to look for a balance between expressivity and complexity as well as for connections with other generalisations of *AAF*s (see (Diller et al. 2020)). According to (Alcântara, Sá, and Acosta-Guadarrama 2019), one of these subclasses, dubbed Attacking Abstract Dialectical Frameworks (ADF^+s), is of particular interest, for the complexity of many reasoning tasks on ADF^+s may likely have the same complexity as standard Dung's *AAF*s (Alcântara, Sá, and Acosta-Guadarrama 2019). With that in mind, we will provide a translation from ADF^+s to *SETAF*s and will prove various equivalences between their semantics, including the equivalence between their complete semantics, grounded semantics, preferred semantics, stable semantics and semi-stable/L-stable semantics. Further, we will show our translation and the translation from *SETAF* to ADF in (Polberg 2016) are bijective functions and each other's inverse provided ADF^+ has no redundant links. Therefore, it is licit to say that *SETAF* corresponds exactly to ADF^+ without redundant links (and vice versa).

The significance of our work is twofold. First, looking inwardly into the formal argumentation field, we conduct a precise characterisation of *SETAF* in ADF^+ without redundant links (and vice versa); our results establish an equivalence between them and contribute to improve our understanding on the connections between the many approaches based on formal argumentation. Second, looking outwardly and more broadly, this paper contributes to an active line of research at the frontier of formal argumentation, which studies the correspondence of argumentation semantics and other semantics for non-monotonic reasoning formalisms; amongst other implications, this potentially allows us to import proof procedures and implementations from formal argumentation to these formalisms (and vice versa).

In short, we will show that *SETAF* and ADF^+ are essentially the same. This conclusion leads to the following practical implications: applications of ADF^+ and ADF (as in (Al-Abdulkarim, Atkinson, and Bench-Capon 2016; Cabrio and Villata 2016; Neugebauer 2017; Pührer 2017)) can be remodelled in the context of *SETAF*. Further, results proven about either formalism carry over to the other. For instance, in (Alcântara, Sá, and Acosta-Guadarrama 2019), Normal Logic Programs have been translated into ADF^+s and some equivalence results have been proved. Due to our findings, the same translation and results can be straightforwardly adapted to *SETAF*s. On the other hand, results obtained for *SETAF* in works such as (Dvořák, Fandinno, and Woltran 2019; Flouris and Bikakis 2019) can be taken for granted in ADF^+.

The paper proceeds as follows. First, we recall the necessary background on ADFs and *SETAF*, including the semantics we will investigate as argument labellings. Next, we consider the Attacking Abstract Dialectical Frameworks (ADF^+s), a fragment of ADFs in which the unique relation involving arguments is the attack relation. In the subsequent section, we show a translation from ADF^+s to *SETAF* and prove the equivalence between their complete models, grounded models, preferred models, stable models and semi-stable/L-stable models (Subsection 4.1), and in Subsection 4.2 we recall a translation from *SETAF* to ADF^+ and show both translations are bijective functions and each other's inverse concerning ADF^+ with no redundant links. Finally, we round off with a discussion of the obtained results and pointers for future works.

2 Background

2.1 Abstract Dialectical Frameworks

Abstract Dialectical Frameworks (ADFs) have been designed in (Brewka and Woltran 2010; Brewka et al. 2013) to treat arguments (called statements there) as abstract and atomic entities. One can see it as a directed graph whose nodes represent statements susceptible to evaluation. The links between nodes represent dependencies: the status (accepted or not accepted) of a node s only depends on the status of its parents ($par(s)$), i.e., the nodes with a direct link to s. We will restrict ourselves to finite ADFs:

Definition 1 (Abstract Dialectical Frameworks)**.** *(Brewka and Woltran 2010) An abstract dialectical framework is a tuple $D = (S, L, C)$ where*

- *S is a finite set of statements (positions, nodes);*
- *$L \subseteq S \times S$ is a set of links, and for each $s \in S$, $par(s) = \{t \in S \mid (t, s) \in L\}$;*
- *$C = \{C_s \mid s \in S\}$ is a set of total functions $C_s : 2^{par(s)} \rightarrow \{\mathbf{t}, \mathbf{f}\}$, one for each statement s. C_s is called the acceptance condition of s.*

The function C_s is intended to determine the acceptance status of a statement s, which only depends on the status of its parent nodes $par(s)$. Intuitively, s will be accepted if there exists $R \subseteq par(s)$ such that $C_s(R) = \mathbf{t}$, which means every statement in R is accepted while each statement in $par(s) - R$ is not. Besides the above, the acceptance conditions in C for an ADF $D = (S, L, C)$ can as well be represented in the following two ways:

- Any function $C_s \in C$ can be represented by the set of subsets of $par(s)$ leading to acceptance, i.e., $C^{\mathbf{t}} = \{C_s^{\mathbf{t}} \mid s \in S\}$, where $C_s^{\mathbf{t}} = \{R \subseteq par(s) \mid C_s(R) = \mathbf{t}\}$. We will indicate this alternative by denoting an ADF as $(S, L, C^{\mathbf{t}})$.
- Any function $C_s \in C$ can also be represented as a classical two-valued propositional formula φ_s over the vocabulary

$par(s)$ as follows:

$$\varphi_s \equiv \bigvee_{R \in C_s^{\mathbf{t}}} \left(\bigwedge_{a \in R} a \wedge \bigwedge_{b \in par(s) - R} \neg b \right). \qquad (1)$$

If $C_s(\varnothing) = \mathbf{t}$ and $par(s) = \varnothing$, we obtain $\varphi_s \equiv \mathbf{t}$. If there is no $R \subset par(s)$ such that $C_s(R) = \mathbf{t}$, then $\varphi_s \equiv \mathbf{f}$. By C^φ we mean the set $\{\varphi_s \mid s \in S\}$. We will indicate this alternative by denoting an ADF as (S, L, C^φ). We also emphasize any propositional formula φ_s equivalent (in the classical two-valued sense) to the formula in Equation (1) can be employed to represent C_s.

When referring to an ADF as (S, L, C^φ), we will assume the acceptance formulas implicitly specify the parents a node depends on. Then, the set L of links between statements can be ignored, and the ADF can be represented as (S, C^φ), where L gets recovered by $(t, s) \in L$ iff t appears in φ_s. In order to define the different semantics for ADFs over the set of statements S, we will resort to the notion of (3-valued) interpretations:

Definition 2 (Interpretations and Models)**.** *(Brewka and Woltran 2010) Let $D = (S, C^\varphi)$ be an ADF. A 3-valued interpretation (or simply interpretation) over S is a mapping $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ that assigns to each statement a truth value amongst true ($\mathbf{t}$), false ($\mathbf{f}$) and unknown ($\mathbf{u}$). Interpretations will be extended to assign values to formulas over statements according to Kleene's strong 3-valued logic (Kleene et al. 1952): negation switches $\mathbf{t}$ and $\mathbf{f}$, and leaves $\mathbf{u}$ unchanged; a conjunction is $\mathbf{t}$ if both conjuncts are $\mathbf{t}$, it is $\mathbf{f}$ if some conjunct is $\mathbf{f}$ and it is $\mathbf{u}$ otherwise; disjunction is dual. A 3-valued interpretation v is a model of D if for all $s \in S$ we have $v(s) \neq \mathbf{u}$ implies $v(s) = v(\varphi_s)$.*

Sometimes we will refer to an interpretation v over S as a set $V = \{s \mid s \in S \textit{ and } v(s) = \mathbf{t}\} \cup \{\neg s \mid s \in S \textit{ and } v(s) = \mathbf{f}\}$. Obviously, if neither $s \in V$ nor $\neg s \in V$, then $v(s) = \mathbf{u}$.

Furthermore, the three truth values are partially ordered by $\leq_i$ according to their information content: $\mathbf{u} <_i \mathbf{t}$, $\mathbf{u} <_i \mathbf{f}$, no other pair is in $<_i$, and $\leq_i$ is the reflexive transitive closure of $<_i$. The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ forms a complete meet-semilattice[1] with the meet operation $\sqcap$. This meet can be read as consensus and assigns $\mathbf{t} \sqcap \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap \mathbf{f} = \mathbf{f}$, and returns $\mathbf{u}$ otherwise.

The information ordering $\leq_i$ extends as usual to interpretations v_1, v_2 over S such that $v_1 \leq_i v_2$ iff $v_1(s) \leq_i v_2(s)$ for all $s \in S$. The set of all 3-valued interpretations over S forms a complete meet-semilattice with respect to $\leq_i$. The consensus meet operation $\sqcap$ of this semilattice is given by $(v_1 \sqcap v_2)(s) = v_1(s) \sqcap v_2(s)$ for all $s \in S$. The least element of this semilattice is the interpretation v such that $v(s) = \mathbf{u}$ for each $s \in S$.

In (Brewka et al. 2013), the semantics for ADFs were defined via an operator Γ_D:

Definition 3 (Γ_D Operator)**.** *(Brewka et al. 2013) Let $D = (S, L, C^\varphi)$ be an ADF and v be a 3-valued interpretation over S. We have*

$$\Gamma_D(v)(s) = \bigsqcap \{w(\varphi_s) \mid w \in [v]_2\},$$

in which $[v]_2 = \{w \mid v \leq_i w$ and for each $s \in S$, it holds $w(s) \in \{\mathbf{t}, \mathbf{f}\}\}$.

Each element in $[v]_2$ is a 2-valued interpretation extending v. The elements of $[v]_2$ form an $\leq_i$-antichain with greatest lower bound $v = \bigsqcap[v]_2$. For each $s \in S$, Γ_D returns the consensus truth value for φ_s, where the consensus takes into account all possible 2-valued interpretations w extending v. If v is 2-valued, we get $[v]_2 = \{v\}$. In this case, $\Gamma_D(v)(s) = v(\varphi_s)$ and v is a 2-valued model for D iff $\Gamma_D(v) = v$. As $[v]_2$ has only 2-valued interpretations, if φ_s^1 is equivalent to φ_s^2 in the classical two-valued sense, it is clear

$$\bigsqcap \{w(\varphi_s^1) \mid w \in [v]_2\} = \bigsqcap \{w(\varphi_s^2) \mid w \in [v]_2\}.$$

That means when defining Γ_D operator, it does not matter the acceptance formula we choose as far as it is equivalent in the classical 2-valued sense. In addition, Γ_D operator can as well be employed to characterize complete interpretations:

Definition 4 (Complete Interpretations)**.** *(Brewka et al. 2013) Let $D = (S, L, C^\varphi)$ be an ADF and v be a 3-valued interpretation over S. We state v is a* complete *interpretation of D iff $v = \Gamma_D(v)$.*

As shown in (Brewka and Woltran 2010), Γ_D operator is $\leq_i$-monotonic. Then a $\leq$-least fixpoint of Γ_D is always guaranteed to exists for every ADF D. Note complete interpretations of D are also models of D. For this reason, they are also called complete models. The notion of reduct borrowed from logic programming (Gelfond and Lifschitz 1988) is reformulated to deal with ADFs:

Definition 5 (Reduct)**.** *(Brewka et al. 2013) Let $D = (S, L, C^\varphi)$ be an ADF and v be a 2-valued model of D. The reduct of D with v is given by the ADF, $D^v = (E_v, L^v, C^v)$, in which $E_v = \{s \in S \mid v(s) = \mathbf{t}\}$, $L^v = L \cap (E_v \times E_v)$, and $C^v = \{\varphi_s^v \mid s \in E_v$ and $\varphi_s^v = \varphi_s[b/\mathbf{f} : v(b) = \mathbf{f}]\}$; i.e., in each acceptance formula, φ_s^v, we replace in φ_s every statement $b \in S$ by $\mathbf{f}$ if $v(b) = \mathbf{f}$.*

We can now define some of the main semantics for an ADF as follows:

Definition 6 (Semantics)**.** *(Brewka et al. 2013) Let $D = (S, L, C^\varphi)$ be an ADF, and v a model of D. We state that*

- *v is a* grounded model *of D iff v is the $\leq_i$-least complete model of D.*
- *v is a* preferred model *of D iff v is a $\leq_i$-maximal complete model of D.*
- *v is a* stable model *of D iff v is a 2-valued model of D such that v is the grounded model of $D^v = (E_v, L^v, C^v)$.*

We proceed by displaying an example to illustrate these semantics:

Example 1. *Consider the ADF, $D = (S, C^\varphi)$, given by*

$$a[\neg b] \qquad b[\neg a] \qquad c[a \wedge \neg b \wedge \neg c],$$

[1] A complete meet-semilattice is such that every non-empty finite subset has a greatest lower bound, the meet; and every nonempty directed subset has a least upper bound. A subset is directed if any two of its elements have an upper bound in the set.

where $S = \{a, b, c\}$, and the acceptance formula of each $s \in S$ is written in square brackets on the right of s. For the semantics of D, we have

- *Complete models: $\varnothing, \{a, \neg b\}$ and $\{b, \neg a, \neg c\}$*
- *Grounded models: $\varnothing$*
- *Preferred models: $\{a, \neg b\}$ and $\{b, \neg a, \neg c\}$*
- *Stable models: $\{b, \neg a, \neg c\}$*

Notice some ADFs have no stable models. For instance, in an ADF whose unique statement is $a[\neg a]$, there is no stable model. Furthermore, an ADF can have more than one stable model as in the ADF represented by $a[\neg b]$ and $b[\neg a]$, which has $\{a, \neg b\}$ and $\{b, \neg a\}$ for its stable models. In contrast, the grounded model is unique for each ADF (see (Brewka and Woltran 2010; Brewka et al. 2013)).

2.2 SETAF

In (Nielsen and Parsons 2006) it was proposed an extension of Dung's Abstract Argumentation Frameworks (Dung 1995) (*AAF*s) to allow joint attacks on arguments. The resulting framework, called *SETAF*, is displayed below:

Definition 7 (*SETAF*)**.** *(Nielsen and Parsons 2006) A framework with sets of attacking arguments (SETAF) is a pair $S = (A, R)$, in which A is a finite set of arguments and $R \subseteq 2^A \times A$ is an attack relation such that if $(B, a) \in R$, there is no $B' \subset B$ such $(B', a) \in R$, i.e., B is a minimal set (w.r.t. $\subseteq$) attacking a.*

In an *AAF*, only individual arguments can attack arguments. In a *SETAF* the novelty is that sets of two or more arguments can also attack arguments. This means a *SETAF*s (A, R), in which for each $(B, a) \in R$ it holds $|B| = 1$, amount to (standard Dung) *AAF*s. Besides, unlike the original definition of *SETAF*, and in order to satisfy our current purpose in this paper, Definition 7 allows the empty set to attack an argument $a \in A$, i.e., it is possible to have $(\varnothing, a) \in R$.

As in (Linsbichler, Pührer, and Strass 2016), we define the 3-valued counterparts based on labellings of the semantics introduced in (Nielsen and Parsons 2006) by following the conventions introduced in (Caminada and Gabbay 2009).

Definition 8 (Labellings)**.** *Let $S = (A, R)$ be a SETAF. A SETAF labelling is a function $Lab : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}^2$. A SETAF labelling is called* admissible *iff for each $a \in A$*

- *If $Lab(a) = \mathbf{t}$, then for each $B \subseteq A$ such that $(B, a) \in R$, it holds $\exists b \in B$ such that $Lab(b) = \mathbf{f}$.*
- *If $Lab(a) = \mathbf{f}$, then there exists $B \subseteq A$ such that $(B, a) \in R$ and $\forall b \in B$ it holds $Lab(b) = \mathbf{t}$.*

A SETAF labelling is called complete *iff it is admissible and for each $a \in A$,*

- *If $Lab(a) = \mathbf{u}$ there exists $B \subseteq A$ such that $(B, a) \in R$ and $\forall b \in B$ it holds $Lab(b) \neq \mathbf{f}$ and for each $B \subseteq A$ such that $(B, a) \in R$, it holds $\exists b \in B$ such that $Lab(b) \neq \mathbf{t}$.*

We write $\mathtt{in}(Lab)$ for $\{a \in A \mid Lab(a) = \mathbf{t}\}$, $\mathtt{out}(Lab)$ for $\{a \in A \mid Lab(a) = \mathbf{f}\}$ and $\mathtt{undec}(Lab)$ for $\{a \in A \mid Lab(a) = \mathbf{u}\}$. As a *SETAF* labelling essentially defines a partition among the arguments, we sometimes write Lab as a triple $(\mathtt{in}(Lab), \mathtt{out}(Lab), \mathtt{undec}(Lab))$. We can now describe the *SETAF* semantics studied in this paper:

Definition 9 (Semantics)**.** *Let $S = (A, R)$ be a SETAF. A complete SETAF labelling Lab is called*

- grounded *iff $\mathtt{in}(Lab)$ is minimal (w.r.t. $\subseteq$) among all complete SETAF labellings of S.*
- preferred *iff $\mathtt{in}(Lab)$ is maximal (w.r.t. $\subseteq$) among all complete SETAF labellings of S.*
- stable *iff $\mathtt{undec}(Lab) = \varnothing$.*
- semi-stable *iff $\mathtt{undec}(Lab)$ is minimal (w.r.t. $\subseteq$) among all complete SETAF labellings of S.*

Let us consider the following example:

Example 2. *Consider the SETAF $S = (A, R)$ below:*

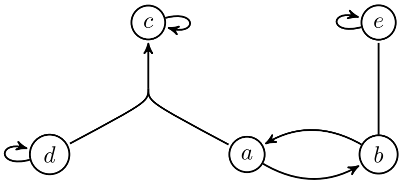


Figure 1: A *SETAF S*

Concerning the semantics of S, we have

- *Complete labellings: $Lab_1 = (\varnothing, \varnothing, \{a, b, c, d, e,\})$, $Lab_2 = (\{a\}, \{b\}, \{c, d, e\})$ and $Lab_3 = (\{b\}, \{a, e\}, \{c, d\})$;*
- *Grounded labellings: $Lab_1 = (\varnothing, \varnothing, \{a, b, c, d, e\})$;*
- *Preferred labellings: $Lab_2 = (\{a\}, \{b\}, \{c, d, e\})$ and $Lab_3 = (\{b\}, \{a, e\}, \{c, d\})\}$;*
- *Stable labellings: none;*
- *Semi-stable labellings:$Lab_3 = (\{b\}, \{a, e\}, \{c, d\})$.*

In the next section, we will focus on a fragment of ADFs, dubbed Attacking Abstract Dialectical Frameworks (ADF^+s), and in the sequel we will show that ADF^+s are enough to capture any of the *SETAF* semantics based on complete models we mentioned above.

3 Attacking Abstract Dialectical Frameworks

Now we consider the Attacking Abstract Dialectical Frameworks (ADF^+s), a fragment of ADFs in which the unique relation involving statements is the attack relation. First we should recall the notions of supporting and attacking links:

Definition 10 (Supporting and Attacking Links (Brewka and Woltran 2010))**.** *Let $D = (S, L, C)$ be an ADF. A link $(r, s) \in L$ is*

supporting in D *iff for no $R \subseteq par(s)$ we have $C_s(R) = \mathbf{t}$ and $C_s(R \cup \{r\}) = \mathbf{f}$.*

[2] In (Caminada and Gabbay 2009), in, out and undec were used instead of $\mathbf{t}, \mathbf{f}$ and $\mathbf{u}$ respectively.

attacking in D *iff for no* $R \subseteq par(s)$ *we have* $C_s(R) = \mathbf{f}$ *and* $C_s(R \cup \{r\}) = \mathbf{t}$.

A link (r, s) is *redundant* if it is both attacking and supporting. Redundant links can be deleted from an ADF as they mean no real dependencies (Brewka and Woltran 2010). Again in (Brewka and Woltran 2010), the authors introduced the Bipolar Abstract Dialectical Frameworks ($BADF$), a subclass of ADFs in which every link is either supporting or attacking. Now we even regard a subclass of $BADFs$ in which only attacking links are admitted:

Definition 11 (ADF^+)**.** *An Attacking Abstract Dialectical Framework, denoted by* ADF^+*, is an* ADF (S, L, C) *such that every* $(r, s) \in L$ *is an attacking link, i.e., for every* $s \in S$*, if* $C_s(M) = \mathbf{t}$*, then for every* $M' \subseteq M$*, we have* $C_s(M') = \mathbf{t}$.

In an ADF^+ (S, L, C), for each $s \in S$, its acceptance formula φ_s can be simplified:

Theorem 1. *(Alcântara, Sá, and Acosta-Guadarrama 2019) Let* $D = (S, L, C^{\mathbf{t}})$ *be an* ADF^+ *and, for each* $s \in S$*, let* $C_s^{max} = \{R \in C_s^{\mathbf{t}} \mid$ *there is no* $R' \in C_s^{\mathbf{t}}$ *such that* $R \subset R'\}$*. Then, for every* $s \in S$*,*

$$\varphi_s \equiv \bigvee_{R \in C_s^{max}} \bigwedge_{b \in par(s) - R} \neg b.$$

Hence, in ADF^+s, every acceptance formula corresponds to a propositional formula in the disjunctive normal form, where each disjunct is a conjunction of negative atoms. Notice replacing an acceptance formula by a two-valued equivalent one does not change complete semantics, and we are not interested in the three-valued models of the ADF^+. The importance of these formulas will be evident below. Before, however, note ADF^+ does not prohibit redundant links. For instance, consider the ADF $D = (S, L, C)$, in which $S = \{a, b, c\}$, $L = \{(b, a), (c, a)\}$ and $C_a^{\mathbf{t}} = \{\{b\}, \varnothing\}$ and $C_b^{\mathbf{t}} = C_c^{\mathbf{t}} = \{\varnothing\}$. We know D is an ADF^+ as both (b, a) and (c, a) are attacking links. In addition, (b, a) is a redundant link as it is also supporting. Redundant links can be easily identified in ADF^+s:

Theorem 2. *Let* $D = (S, L, C^{\mathbf{t}})$ *be an* ADF^+*. Then* D *has no redundant links in* L *iff*

$$L = \{(r, s) \mid \exists R \in C_s^{max} \text{ such that } r \notin R\}.$$

Proof.

($\Rightarrow$) Suppose $D = (S, L, C^{\mathbf{t}})$ has no redundant link. By absurd, assume there exists $(r, s) \in L$ such that for any $R \in C_s^{max}$, we have $r \in R$. As $(r, s) \in L$ is not redundant, it cannot be a supporting link. Then there exists $R' \subseteq par(s)$ such that $C_s(R') = \mathbf{t}$ and $C_s(R' \cup \{r\}) = \mathbf{f}$.

Given that $r \in R$ for any $R \in C_s^{max}$, there exists $R'' \in C_s^{max}$ such that $R' \cup \{r\} \subseteq R''$ and $C_s(R'') = \mathbf{t}$. But then, as any link in L is attacking, we obtain $C_s(R' \cup \{r\}) = \mathbf{t}$. An absurd.

($\Leftarrow$) Let $L = \{(r, s) \mid \exists R \in C_s^{max}$ *such that* $r \notin R\}$. By absurd, assume $(r, s) \in L$ is a redundant link. Then, in particular, it is a supporting link, i.e., for every $R \subseteq par(s)$, we have if $R \in C_s^{\mathbf{t}}$, then $(R \cup \{r\}) \in C_s^{\mathbf{t}}$.

Given that $(r, s) \in L$, we know there exists $R \in C_s^{max}$ such that $r \notin R$. This means $R \in C_s^{\mathbf{t}}$. But then we obtain $(R \cup \{r\}) \in C_s^{\mathbf{t}}$. It is an absurd as $R \in C_s^{max}$.

□

This means in order to eliminate redundant links in an ADF^+ $D = (S, L, C)$, it suffices to ignore links $(r, s) \in L$ such that for each $R \in C_s^{max}$ it holds $r \in R$, i.e., the resulting ADF^+ is $D = (S, L', C')$, in which

- $L' = \{(r, s) \in L \mid \exists R \in C_s^{max}$ *such that* $r \notin R\}$
- $C' = \{C'_s \mid s \in S\}$ is a set of total functions $C'_s : 2^{par'(s)} \to \{\mathbf{t}, \mathbf{f}\}$ s.t. $C'_s(R) = C_s(R)$ and $par'(s) = \{r \in S \mid (r, s) \in L'\}$.

Example 3. *Let us recall the* ADF^+ $D = (S, L, C)$ *above in which* $S = \{a, b, c\}$*,* $L = \{(b, a), (c, a)\}$ *and* $C_a^{\mathbf{t}} = \{\{b\}, \varnothing\}$ *and* $C_b^{\mathbf{t}} = C_c^{\mathbf{t}} = \{\varnothing\}$*. From Theorem 2 we know there exists a redundant link in* L *as the unique* $R = \{b\} \in C_a^{max}$ *is not empty. By following the procedure above, we obtain the* ADF^+ $D = (S, L', C')$*, in which* $L' = \{(c, a)\}$ *and* $C_a^{\mathbf{t}} = C_b^{\mathbf{t}} = C_c^{\mathbf{t}} = \{\varnothing\}$*. As expected, the redundant link* (b, a) *was excluded from* $C_a^{\mathbf{t}}$.

Theorem 2 is an important result for our work. We will later employ it to show *SETAF* corresponds to ADF^+ without redundant links.

In Subsection 2.1, we explained how the Γ_D operator is employed to define the semantics for ADF. However, according to (Alcântara, Sá, and Acosta-Guadarrama 2019), when restricted to ADF^+s, it assumes a simpler version:

Theorem 3. *(Alcântara, Sá, and Acosta-Guadarrama 2019) Let* $D = (S, L, C^{\varphi})$ *be an* ADF^+*,* v *be a 3-valued interpretation over* S*, and for each* $s \in S$*, let* φ_s *be the formula*

$$\bigvee_{R \in C_s^{max}} \bigwedge_{b \in par(s) - R} \neg b$$

depicted in Theorem 1. It holds that:

For every $s \in S$, $\Gamma_D(v)(s) = v(\varphi_s)$.

Still in (Alcântara, Sá, and Acosta-Guadarrama 2019), the authors showed that besides being noticeably simpler when restricted to ADF^+, this new characterisation of Γ_D leads to lower complexity of reasoning: while in an ADF, the problem of verifying whether a given interpretation is complete is proved to be DP-complete (Brewka et al. 2013); in ADF^+, owing to this characterisation of Γ_D, this problem can get solved in polynomial time. Hence, the complexity of many reasoning tasks on ADF^+s may likely have the same complexity as standard Dung's *AAF*s (Dung 1995). Another consequence from Theorem 3 is the stable models of an ADF^+ D may be characterised as the two-valued complete models of D:

Theorem 4. *(Alcântara, Sá, and Acosta-Guadarrama 2019) Let* $D = (S, L, C^{\varphi})$ *be an* ADF^+*. Then* v *is a stable model of* D *iff* v *is a 2-valued complete model of* D.

They also defined a new semantics for ADF^+: L-stable.

Definition 12 (L-stable)**.** *(Alcântara, Sá, and Acosta-Guadarrama 2019) Let* $D = (S, L, C^{\varphi})$ *be an* ADF^+*, and* v *be a 3-valued interpretation of* D*. We say that* v *is an L-stable model of* D *iff* v *is a complete model with minimal* $\mathtt{undec}(v) = \{s \in S \mid v(s) = \mathbf{u}\}$ *(w.r.t. set inclusion) among all complete models of* D.

We highlight that the L-stable Models semantics is defined for every ADF^+. Further, the L-stable models of an ADF^+ D will coincide with its stable models whenever D has at least one stable model. In fact, we can understand a stable model v as an L-stable model in which $\texttt{undec}(v) = \varnothing$. Therefore, the existence of a single stable model in D suffices for every L-stable model of D to be also stable.

Example 4. *Consider the ADF^+ $D = (S, C^\varphi)$ given by*

$$\begin{array}{lll} a[\neg b] & b[\neg a] & c[(\neg c \land \neg a) \lor (\neg c \land \neg d)] \\ d[\neg d] & e[\neg e \land \neg b], & \end{array}$$

where $S = \{a, b, c, d, e\}$, and the acceptance formula for each statement $s \in S$ is written in square brackets on the right of s. As for the semantics of D, we have

- *Complete models: $\{a, \neg b\}$, $\{b, \neg a, \neg e\}$ and $\varnothing$*
- *Grounded model: $\varnothing$*
- *Preferred models: $\{a, \neg b\}$ and $\{b, \neg a, \neg e\}$;*
- *Stable models: none;*
- *L-stable models: $\{b, \neg a, \neg e\}$.*

Thus none of these semantics for ADF^+ are equivalent to each other. However, in the sequel, we will show some equivalences between *SETAF*s semantics and ADF^+ semantics. In fact, the main objective of this work is to show each semantics for *SETAF*s presented in Subsection 2.2 has an equivalent one for ADF^+.

4 Equivalence Between ADF^+ and *SETAF*

We will present a translation from ADF^+ to *SETAF* that is able to account for a whole range of equivalences between their semantics. This includes to prove the equivalence between their complete models, grounded models, preferred models, stable models and semi-stable/L-stable models. Next we will recall a translation from *SETAF* to ADF^+ showed in (Polberg 2016) and will prove both translations correspond to bijections and are each other's inverse on appropriate domains.

4.1 From ADF^+ to *SETAF*

Now we will show how to translate ADF^+ to *SETAF*:

Definition 13. *Let $D = (A, L, C^{\mathbf{t}})$ be an ADF^+. The SETAF associated with D is $\mathfrak{S}(D) = (A, R)$, in which $R = \{(B, a) \mid a \in A$ and B is a minimal subset of $par(a)$ such that $B \notin C^{\mathbf{t}}_a\}$.*

The following example illustrates the above concepts.

Example 5. *Consider the ADF^+ $D = (A, L, C^{\mathbf{t}})$ where $A = \{a, b, c, d, e, f\}$, $L = \{(b, a), (a, b), (a, c), (c, c), (d, c), (e, d), (d, e), (b, f), (f, f)\}$, and*

- $C^{\mathbf{t}}_a = C^{\mathbf{t}}_b = C^{\mathbf{t}}_d = C^{\mathbf{t}}_e = C^{\mathbf{t}}_f = \{\varnothing\}$
- $C^{\mathbf{t}}_c = \{\{a\}, \{d\}, \varnothing\}$

As formulae, the acceptance conditions for the statements in D are given by

$$\begin{array}{lll} a[\neg b] & b[\neg a] & c[(\neg c \land \neg d) \lor (\neg c \land \neg a)] \\ d[\neg e] & e[\neg d] & f[\neg f \land \neg b] \end{array}$$

From Definition 13, we obtain the SETAF $\mathfrak{S}(D) = (A, R)$ in which $R = \{(\{b\}, a), (\{a\}, b), (\{c, d\}, c), (\{c, a\}, c), (\{e\}, d), (\{d\}, e), (\{b\}, f), (\{f\}, f)\}$. We depict $\mathfrak{S}(D)$ below:

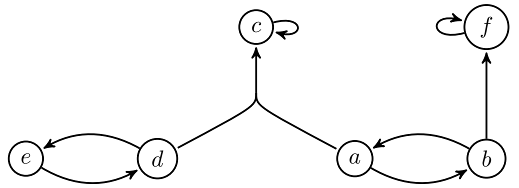


Figure 2: *SETAF* framework $\mathfrak{S}(D)$ corresponding to ADF^+ D.

Now we can prove one of the main results of this paper: that the Complete Models of an ADF^+ correspond to the Complete Labellings of its associated *SETAF*.

Theorem 5. *Let $D = (A, L, C^{\mathbf{t}})$ be an ADF^+ and $\mathfrak{S}(D) = (A, R)$ be the corresponding SETAF. Then v is a complete model of D iff v is a complete labelling of $\mathfrak{S}(D)$.*

Proof. Let $D = (A, L, C^{\mathbf{t}})$ be an ADF^+ and $\mathfrak{S}(D) = (A, R)$ be the corresponding *SETAF*. Let v be a 3-valued interpretation. We will prove v is a complete model of D iff v is a complete labelling of $\mathfrak{S}(D)$:

(⇒) Assume v is a complete model of D, i.e., $v = \Gamma_D(v)$ according to Definition 4. Then from Theorems 1 and 3, it holds for every $a \in A$,

$$\Gamma_D(v)(a) = v(\varphi_a) = v(\bigvee_{M \in C^{max}_a} \bigwedge_{b \in par(a) - M} \neg b) = v(a).$$

Hence,

- If $v(a) = \mathbf{t}$, then there exists $M \in C^{max}_a$ such that for each $b \in par(a) - M$, we have $v(b) = \mathbf{f}$. Thus for each $B \subseteq par(a)$, either $B \in C^{\mathbf{t}}_a$ or there exists $b \in B$ such that $v(b) = \mathbf{f}$. This means for each $(B, a) \in R$, there exists $b \in B$ such that $v(b) = \mathbf{f}$.
- If $v(a) = \mathbf{f}$, then for every $M \in C^{max}_a$ there exists $b \in par(a) - M$ such that $v(b) = \mathbf{t}$. Let $B \subseteq par(a)$ be a minimal set (w.r.t. inclusion order) such that $B = \{b \in par(a) \mid M \in C^{max}_a,\ b \in par(a) - M \textit{ and } v(b) = \mathbf{t}\}$. Clearly B is a minimal subset (w.r.t. inclusion order) of $par(a)$ such that $B \notin C^{\mathbf{t}}_a$. This means there exists $(B, a) \in R$ such that for each $b \in B$, we have $v(b) = \mathbf{t}$.
- If $v(a) = \mathbf{u}$, then for each $M \in C^{max}_a$ there exists $b \in par(a) - M$ such that $v(b) \neq \mathbf{f}$ and there exists $M \in C^{max}_a$ such that for each $b \in par(a) - M$, we have $v(b) \neq \mathbf{t}$. Now let $B' \subseteq par(a)$ be a minimal set (w.r.t. inclusion order) such that $B' = \{b \in par(a) \mid M \in C^{max}_a, b \in par(a) - M \textit{ and } v(b) \neq \mathbf{f}\}$. Then, for each $B'' \subseteq par(a)$, either $B'' \in C^{\mathbf{t}}_a$ or there exists $b \in B''$ such that $v(b) \neq \mathbf{t}$. This means there exists a minimal set (w.r.t. inclusion order) $B \subseteq par(a)$ such that $B \notin C^{\mathbf{t}}_a$ and for each $b \in B$ $v(b) \neq \mathbf{f}$ and for each $B \subseteq par(a)$ such that $B \notin C^{\mathbf{t}}_a$, there exists $b \in B$ such that $v(b) \neq \mathbf{t}$, i.e., there exists $(B, a) \in R$ such that for

each $b \in B$ $v(b) \neq \mathbf{f}$ and for each $(B, a) \in R$, there exists $b \in B$ such that $v(b) \neq \mathbf{t}$

Consequently, v is a complete labelling of $\mathfrak{S}(D)$.

($\Leftarrow$) Assume v is a complete labelling of $\mathfrak{S}(D)$, i.e.,

- If $v(a) = \mathbf{t}$, then for every $(B, a) \in R$, there exists $b \in B$ such that $v(b) = \mathbf{f}$. Let $N = \{b \mid (B, a) \in R, b \in B \textit{ and } v(b) = \mathbf{f}\}$, and $M = par(a) - N$. By absurd, suppose $M \notin C_a^{\mathbf{t}}$. Then, there exists $(B, a) \in R$ such that $B \subseteq M$. In this case, there exists $b \in par(a)$ such that $b \in N$ and $b \in M = par(a) - N$, an absurd! Thus, $M \in C_a^{\mathbf{t}}$. This means there exists $M' \in C_a^{max}$ such that $M \subseteq M'$ and for every $b \in par(a) - M'$, it holds $v(b) = \mathbf{f}$, i.e.,

$$\Gamma_D(v)(a) = v(\varphi_a) = v(\bigvee_{M \in C_a^{max}} \bigwedge_{b \in par(a) - M} \neg b) = \mathbf{t}$$

- If $v(a) = \mathbf{f}$, then there exists $(B, a) \in R$ such that for each $b \in B$, we have $v(b) = \mathbf{t}$. By absurd, suppose there exists $M \in C_a^{max}$ such that $B \cap (par(a) - M) = \varnothing$. In this case, $B \subseteq M$. It is an absurd as $B \notin C_a^{\mathbf{t}}$. Hence, for every $M \in C_a^{max}$, $B \cap (par(a) - M) \neq \varnothing$. This means for every $M \in C_a^{max}$, there exists $b \in pr(a) - M$ such that $v(b) = \mathbf{t}$ i.e.,

$$\Gamma_D(v)(a) = v(\varphi_a) = v(\bigvee_{M \in C_a^{max}} \bigwedge_{b \in par(a) - M} \neg b) = \mathbf{f}$$

- If $v(a) = \mathbf{u}$, then there exists $(B, a) \in R$ such that for each $b \in B$, we have $v(b) \neq \mathbf{f}$ and for each $(B, a) \in R$, there exists $b \in B$ such that $v(b) \neq \mathbf{t}$. Then
  - We already know that $B \cap (par(a) - M) \neq \varnothing$ for every $M \in C_a^{max}$. This means that for every $M \in C_a^{max}$, there exists $b \in pr(a) - M$ such that $v(b) \neq \mathbf{f}$.
  - Let $N = \{b \mid (B, a) \in R, b \in B \textit{ and } v(b) \neq \mathbf{t}\}$, and $M = par(a) - N$. By absurd, suppose $M \notin C_a^{\mathbf{t}}$. Then, there exists $(B, a) \in R$ such that $B \subseteq M$. In this case, there exists $b \in par(a)$ such that $b \in N$ and $b \in M = par(a) - N$, an absurd! Thus, $M \in C_a^{\mathbf{t}}$. Therefore, there exists $M' \in C_a^{max}$ such that $M \subseteq M'$ and for every $b \in par(a) - M'$, it holds $v(b) \neq \mathbf{t}$.

  This means

$$\Gamma_D(v)(a) = v(\varphi_a) = v(\bigvee_{M \in C_a^{max}} \bigwedge_{b \in par(a) - M} \neg b) = \mathbf{u}$$

□

From the equivalence shown in Theorem 5, the following results are obtained immediately:

Theorem 6. *Let $D = (A, L, C^{\mathbf{t}})$ be an ADF^+ and $\mathfrak{S}(D) = (A, R)$ be the corresponding SETAF. We have*

- *v is a grounded model of D iff v is a grounded model of $\mathfrak{S}(D)$.*
- *v is a preferred model of P iff v is a preferred model of $\mathfrak{S}(D)$.*
- *v is a stable model of P iff v is a stable model of $\mathfrak{S}(D)$.*
- *v is an L-stable model of P iff v is a semi-stable model of $\mathfrak{S}(D)$.*

Recalling Example 5, we obtain that, as expected, the ADF^+ D and its corresponding *SETAF* $\mathfrak{S}(D)$ share the same semantics:

- Complete models:

$$\left\{ \begin{array}{ll} (\varnothing, \varnothing, \{a,b,c,d,e,f\}), & (\{a\}, \{b\}, \{c,d,e,f\}), \\ (\{b\}, \{a,f\}, \{c,d,e\}), & (\{d\}, \{e\}, \{a,b,d,f\}), \\ (\{e\}, \{d\}, \{a,b,d,f\}), & (\{a,d\}, \{b,c,e\}, \{f\}), \\ (\{a,e\}, \{b,d\}, \{c,f\}), & (\{b,d\}, \{a,e,f\}, \{c\}), \\ (\{b,e\}, \{a,d,g\}, \{c\}) & \end{array} \right\}$$

- Grounded model: $\{(\varnothing, \varnothing, \{a,b,c,d,e,f\})\}$;
- Preferred models:

$$\left\{ \begin{array}{ll} (\{a,d\}, \{b,c,e\}, \{f\}) & (\{a,e\}, \{b,d\}, \{c,f\}), \\ (\{b,d\}, \{a,e,f\}, \{c\}), & (\{b,e\}, \{a,d,g\}, \{c\}) \end{array} \right\}$$

- Stable model: $\varnothing$;
- Semi-stable/L-stable models:

$$\left\{ \begin{array}{ll} (\{a,d\}, \{b,c,e\}, \{f\}) & (\{b,d\}, \{a,e,f\}, \{c\}), \\ (\{b,e\}, \{a,d,g\}, \{c\}) & \end{array} \right\}$$

From Theorems 5 and 6, we see the ADF^+ D and the corresponding *SETAF* $\mathfrak{S}(D)$ produce the same semantics. This result sheds light on the connections between ADF^+s and *SETAF*s. Theorem 6 ensures the translation from ADF^+ to *SETAF* in Definition 13 is robust enough to guarantee at least the equivalence between any semantics based on complete models. Indeed, we will show that the relation between ADF^+ without redundant links and *SETAF*s is deeper than what we observe at the level of semantics. We will do so by recalling a translation from *SETAF* to ADF^+ originally presented in (Polberg 2016) and showing that both translations discussed in our work are bijective functions and each other's inverse.

4.2 From *SETAF* to ADF^+

Now we will show a translation from *SETAF* to ADF^+:

Definition 14. *(Polberg 2016) Let $S = (A, R)$ be a SETAF. The ADF corresponding to S is $\mathfrak{D}(S) = (A, L, C)$, in which $L = \{(b, a) \mid b \in B \textit{ for some } (B, a) \in R\}$ and $C = \{C_a \mid a \in A\}$, s.t. each $C_a : 2^{par(a)} \to \{\mathbf{t}, \mathbf{f}\}$ is created in the following way:*

$$C_a(B) = \left\{ \begin{array}{ll} \mathbf{f} & \textit{if } \exists (X, a) \in R \textit{ such that } X \subseteq B \\ \mathbf{t} & \textit{otherwise} \end{array} \right.$$

We can prove the resulting $\mathfrak{D}(S) = (A, L, C)$ is indeed an ADF^+ for which L has no redundant links:

Theorem 7. *Let $S = (A, R)$ be a SETAF and $\mathfrak{D}(S) = (A, L, C)$ be its corresponding ADF. Then $\mathfrak{D}(S)$ is an ADF^+ with no redundant links.*

Proof. By absurd, suppose $\mathfrak{D}(S)$ is not an ADF^+. This means there exists a link $(b, a) \in L$ for which exists some $B \subseteq par(a)$ where $C_a(B) = \mathbf{f}$ and $C_a(B \cup \{b\}) = \mathbf{t}$ (Definition 10). As $C_a(B) = \mathbf{f}$, from Definition 14, we obtain $\exists (X, a) \in R$ such that $X \subseteq B$. Then we can say $\exists (X, a) \in R$

such that $X \subseteq B \cup \{b\}$. But then $C_a(B \cup \{b\}) = \mathbf{f}$. An absurd!

Again, by absurd, assume there exists a redundant link $(b,a) \in L$. According to Theorem 2, $b \in B$ for every $B \in C_a^{max}$. As $(b,a) \in L$, we know from Definition 14 there exists $(B',a) \in R$ such that $b \in B'$. This also means there exists no $B'' \subset B'$ such that $(B'',a) \in R$ (Definition 7). Then, from Definition 14, we obtain $C^{\mathbf{t}}(B') = \mathbf{f}$ and $C^{\mathbf{t}}(B' - \{b\}) = \mathbf{t}$. But then, there exists $B \in C_a^{max}$, in which $b \notin B$. It is an absurd! Thus, there is no redundant link $(b,a) \in L$. □

Example 6. *Consider the SETAF $\mathfrak{S}(D) = (A,R)$ depicted in Figure 2. According to Definition 14, we will obtain the ADF $\mathfrak{D}(\mathfrak{S}(D)) = (A,L,C)$, in which $A = \{a,b,c,d,e,f\}$, $L = \{(b,a),(a,b), (a,c), (c,c), (d,c), (e,d),(d,e),(b,f),(f,f)\}$, and*

- $C_a^{\mathbf{t}} = C_b^{\mathbf{t}} = C_d^{\mathbf{t}} = C_e^{\mathbf{t}} = C_f^{\mathbf{t}} = \{\{\}\}$
- $C_c^{\mathbf{t}} = \{\{a\},\{d\},\{\}\}$

As one can check from Examples 5 and 6, $\mathfrak{D}(\mathfrak{S}(D)) = D$. This is not a mere coincidence! As we will prove next, the functions $\mathfrak{S}$ and $\mathfrak{D}$ are bijective and each other's inverse, provided D is an ADF^+ with no redundant links.

Theorem 8. *Let $D = (A,L,C^{\mathbf{t}})$ be an ADF^+ without redundant links and $\mathfrak{S}(D) = (A,R)$ be the corresponding SETAF, then $\mathfrak{D}(\mathfrak{S}(D)) = D$.*

Proof. According to Definition 14, $\mathfrak{D}(\mathfrak{S}(D)) = (A,L',C')$, has $L' = \{(b,a) \mid b \in B \textit{ for some } (B,a) \in R\}$, $C' = \{C'_a \mid a \in A\}$, and every $C'_a : 2^{par(a)} \to \{\mathbf{t},\mathbf{f}\}$ is created in such a way that:

$$C'_a(B) = \begin{cases} \mathbf{f} & \textit{if } \exists (X,a) \in R \textit{ such that } X \subseteq B \\ \mathbf{t} & \textit{otherwise} \end{cases}$$

We will show $L = L'$: given that L has no redundant links, from Theorem 2, we can ensure that $L = \{(b,a) \mid \exists M \in C_a^{max} \textit{ such that } b \notin M\}$. Hence

- $(b,a) \in L \Rightarrow \exists M \in C_a^{max}$ *such that* $b \notin M \Rightarrow$ there exists a minimal (w.r.t. $\subseteq$) $B \subseteq M \cup \{b\}$ such that $C_a(B) = \mathbf{f}$ and $b \in B \Rightarrow \exists B \subseteq A$ such that $(B,a) \in R$ and $b \in B \Rightarrow (b,a) \in L'$.
- $(b,a) \in L' \Rightarrow \exists B \subseteq A$ such that $(B,a) \in R$ and $b \in B \Rightarrow$ there exists a minimal set (w.r.t. $\subseteq$) $B \subseteq par(a)$ such that $C_a(B) = \mathbf{f}$ and $b \in B \Rightarrow$ there exists a maximal set (w.r.t. $\subseteq$) $M \supseteq (B - \{b\})$ such that $C_a(M) = \mathbf{t}$ and $b \notin M \Rightarrow \exists M \in C_a^{max}$ *such that* $b \notin M \Rightarrow (b,a) \in L$.

We will show $C = C'$; then for each $a \in A$, for each $B \in par(a)$,

- $C_a(B) = \mathbf{t} \Rightarrow$ for each $B' \subseteq B$, we have $C_a(B') = \mathbf{t} \Rightarrow$ for each $B' \subseteq B$, we have $(B',a) \notin R \Rightarrow C'_a(B) = \mathbf{t}$;
- $C_a(B) = \mathbf{f} \Rightarrow$ there exists a minimal $B' \subseteq B$ such that $C_a(B') = \mathbf{f} \Rightarrow (B',a) \in R$ and $B' \subseteq B \Rightarrow C'_a(B) = \mathbf{f}$.

□

Theorem 9. *Let $S = (A,R)$ be a SETAF and $\mathfrak{D}(S) = (A,L,C)$ be it's corresponding ADF^+, then $\mathfrak{S}(\mathfrak{D}(S)) = S$.*

Proof. From Definition 13, we have that $\mathfrak{S}(\mathfrak{D}(S)) = (A,R')$, where $R' = \{(B,a) \mid a \in A \textit{ and } B \textit{ is a minimal subset of } par(a) \textit{ such that } B \notin C_a^{\mathbf{t}}\}$. We will show that $R = R'$:

- ($R \subseteq R'$) Suppose $(B,a) \in R \Rightarrow B \subseteq par(a)$, $C_a(B) = \mathbf{f}$ and $\forall B' \subset B$, we have $C_a(B') = \mathbf{t}$, $\Rightarrow B$ is a minimal subset of $par(a)$ such that $B \notin C_a^{\mathbf{t}} \Rightarrow (B,a) \in R'$
- ($R' \subseteq R$) Suppose $(B,a) \in R' \Rightarrow B$ is a minimal subset of $par(a)$ such that $B \notin C_a^{\mathbf{t}} \Rightarrow B \subseteq par(a)$, $C_a(B) = \mathbf{f}$ and $\forall B' \subset B$, we have $C_a(B') = \mathbf{t}$. By absurd, suppose $(B,a) \notin R$. In this case, as $C_a(B) = \mathbf{f}$, there exists $B' \subset B$ such that $(B',a) \in R$. But then $C_a(B') = \mathbf{f}$. This is an absurd as $C_a(B') = \mathbf{t}$ for each $B' \subset B$. Thus, $(B,a) \in R$.

□

The above results guarantee the back and forth translations between ADF^+ and *SETAF* are one to one related provided ADF^+ has no redundant links. Clarifying this relationship is important as these formalisms are quite prominent. It also allow us to conceive any attack to any argument a in a *SETAF* as part of the acceptance condition of a in an ADF^+ (and vice versa). For instance, considering our translations, one can check the attack $(\varnothing,a)$ in a *SETAF* corresponds to $a[\mathbf{f}]$ in an ADF^+, whose meaning indicates it is not possible to accept a. Hence, if one wants to restore the usual definition of *SETAF* (A,R) with its attack relation R defined as $R \subseteq (2^A - \varnothing) \times A$ while preserving the one to one correspondence with ADF^+, one will have to prohibit statements with $\mathbf{f}$ as their acceptance formula. What is more, if we allow $(B,a) \in R$ and $(B',a) \in R$ in a *SETAF* $S = (A,R)$ such that $B \subset B'$, according to Definition 14 and Theorem 7, there will be redundant links in the resulting ADF^+ $\mathfrak{D}(S)$. By establishing this connection, we can say redundant links are worthless to ADF^+s in the same extent as attacks (B,a) in a *SETAF* in which B is not a minimal set (w.r.t. $\subseteq$).

5 Conclusions and Future Works

This work has exploited the connections between a fragment of Abstract Dialectical Frameworks (ADFs), called Attacking Abstract Dialectical Frameworks (ADF^+s), and an extension of Dung's Abstract Argumentation Frameworks(Dung 1995), called *SETAF*, that allows joint attacks on arguments. We have provided a translation from ADF^+s to *SETAF*s and proved various equivalences between their semantics, including the equivalence between their complete semantics, grounded semantics, preferred semantics, stable semantics and semi-stable/L-stable semantics. Furthermore, we defined a translation from ADF^+ to *SETAF* and showed that our translation and the translation from *SETAF* to ADF in (Polberg 2016) are bijective functions and each other's inverse provided ADF^+ has no redundant link. Consequently, we proved that a fragment of ADFs, namely Attacking Abstract Dialectical Frameworks (ADF^+s) without redundant links, correspond exactly to *SETAF*. Not only their semantic models correspond to one another, they actually coincide precisely.

The results showed in this paper not only guarantee the equivalence between the aforementioned semantics for ADF^+ and *SETAF*, but also ADF^+ and *SETAF* are one to one related assuming ADF^+ has no redundant links. In particular, these results allow us to make connections between attacks in *SETAF* with acceptance conditions in ADF^+ (and vice versa), and to identify easily redundant links in a ADF^+. Besides improving our understanding on the connections between *SETAF* and ADF^+, this paper contributes to an active line of research at the frontier of formal argumentation, which studies the correspondence of argumentation semantics and other semantics for non-monotonic reasoning formalisms; amongst other implications, this potentially allows us to import proof procedures and implementations from formal argumentation to these formalisms and vice-versa.

Some connections between Abstract Dialectical Frameworks and Normal Logic Programs have already been established (Brewka and Woltran 2010; Strass 2013; Alcântara, Sá, and Acosta-Guadarrama 2019). Given the results unveiled in the current paper, we also intend to exploit the connections between Logic Programming semantics and *SETAF*s semantics. Clarifying this relationship is important as these formalisms are quite prominent in related, but somewhat different areas, namely declarative problem solving and argumentation.

References

Al-Abdulkarim, L.; Atkinson, K.; and Bench-Capon, T. 2016. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artificial Intelligence and Law* 24(1):1–49.

Alcântara, J.; Sá, S.; and Acosta-Guadarrama, J. 2019. On the equivalence between abstract dialectical frameworks and logic programs. *arXiv preprint arXiv:1907.09548*.

Brewka, G., and Woltran, S. 2010. Abstract dialectical frameworks. In *Twelfth International Conf. on the Principles of Knowledge Representation and Reasoning*, 102–111. AAAI Press.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2013. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 803–809. AAAI Press.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J.; and Woltran, S. 2017. Abstract dialectical frameworks. an overview. *IFCoLog Journal of Logics and Their Applications* 4(8):2263–2317.

Brewka, G.; Polberg, S.; and Woltran, S. 2014. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems* 29(1):30–38.

Cabrio, E., and Villata, S. 2016. Abstract dialectical frameworks for text exploration. In *International Conference on Agents and Artificial Intelligence*, volume 2, 85–95. SciTePress.

Caminada, M. W., and Gabbay, D. M. 2009. A logical account of formal argumentation. *Studia Logica* 93(2-3):109.

Caminada, M. 2006a. On the issue of reinstatement in argumentation. *Logics in artificial intelligence* 111–123.

Caminada, M. 2006b. Semi-stable semantics. *1st International Conference on Computational Models of Argument (COMMA)* 144:121–130.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 378–389. Springer.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2013. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning* 54(7):pp–876.

Coste-Marquis, S.; Konieczny, S.; Marquis, P.; and Ouali, M. A. 2012. Weighted attacks in argumentation frameworks. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 593–597. AAAI Press,.

Diller, M.; Keshavarzi Zafarghandi, A.; Linsbichler, T.; and Woltran, S. 2020. Investigating subclasses of abstract dialectical frameworks. *Argument & Computation* 11(1-2):191–219.

Dung, P. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Dunne, P. E.; Hunter, A.; McBurney, P.; Parsons, S.; and Wooldridge, M. 2011. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence* 175(2):457–486.

Dvořák, W.; Fandinno, J.; and Woltran, S. 2019. On the expressive power of collective attacks. *Argument & Computation* 10(2):191–230.

Flouris, G., and Bikakis, A. 2019. A comprehensive study of argumentation frameworks with sets of attacking arguments. *International Journal of Approximate Reasoning* 109:55–86.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. of the 5th International Conference on Logic Programming (ICLP)*, volume 88, 1070–1080.

Kleene, S. C.; de Bruijn, N.; de Groot, J.; and Zaanen, A. C. 1952. *Introduction to metamathematics*, volume 483. van Nostrand New York.

Linsbichler, T.; Pührer, J.; and Strass, H. 2016. A uniform account of realizability in abstract argumentation. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 252–260. IOS Press.

Martınez, D. C.; Garcıa, A. J.; and Simari, G. R. 2008. An abstract argumentation framework with varied-strength attacks. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, 135–144.

Modgil, S. 2009. Reasoning about preferences in argumentation frameworks. *Artificial intelligence* 173(9-10):901–934.

Neugebauer, D. 2017. Generating defeasible knowledge bases from real-world argumentations using d-bas. In *Proceedings of the 1st Workshop on Advances in Argumentation in Artificial Intelligence*, 105–110.

Nielsen, S. H., and Parsons, S. 2006. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *International Workshop on Argumentation in Multi-Agent Systems*, 54–73. Springer.

Oren, N., and Norman, T. J. 2008. Semantics for evidence-based argumentation. In Besnard, P.; Doutre, S.; and Hunter, A., eds., *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA)*, volume 172, 276–284. IOS Press.

Polberg, S., and Oren, N. 2014. Revisiting support in abstract argumentation systems. In *COMMA*, 369–376.

Polberg, S. 2016. Understanding the abstract dialectical framework. In *European Conference on Logics in Artificial Intelligence*, 430–446. Springer.

Pührer, J. 2017. Argueapply: A mobile app for argumentation. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 250–262. Springer.

Rahwan, I., and Simari, G. R. 2009. *Argumentation in artificial intelligence*, volume 47. Springer.

Strass, H. 2013. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence* 205:39–70.

Verheij, B. 1996. Two approaches to dialectical argumentation: admissible sets and argumentation stages. *Proc. NAIC* 96:357–368.

Working Memory for Assessment Under Inconsistency

Pierre Bisquert[1,2] , **Florence Dupin de Saint-Cyr**[3]
[1]IATE, Univ Montpellier, INRAE, Institut Agro, Montpellier, France
[2]LIRMM, Inria, Univ Montpellier, CNRS, Montpellier, France
[3]IRIT, Toulouse University
pierre.bisquert@inrae.fr, florence.bannay@irit.fr

Abstract

This paper proposes a new way of handling the inconsistency of a knowledge base when answering to a query about the validity of a formula. The idea is inspired by human behavior in front of inconsistency, namely, try to never encounter it. For this purpose, we encode a kind of compartmentalization of the working memory. More precisely, given a query and a potentially inconsistent knowledge base, called long term memory, our system only loads in working memory the consistent knowledge which is the most related to the query. We position this system with regard to a major reference in the field, Brewka's preferred subtheories, and study its efficiency by providing complexity and experimental results.

keywords: Inconsistency handling, SAT, Bounded rationality, Preferred subtheories

1 Introduction

How do humans reason in the presence of contradictory information? Some psychologists (De Neys and Everaerts 2008) answer that they inhibit counter-examples to come to their mind. Translating this phenomena inside a framework that uses the distinction done by (Baddeley and Hitch 1974) between long-term (*LTM*) and working memory (*WM*) where *WM* is conceived as the "activated" part of the long-term memory (*LTM*), (Barrouillet and Camos 2007) studies how memory activation is produced or inhibited. Findings of (De Neys, Schaeken, and d'Ydewalle 2005) corroborate that *WM*-resources are used for retrieval and inhibition of stored counter-examples and for avoiding conflicts with the logical validity of a reasoning problem (De Neys, Schaeken, and d'Ydewalle 2005). Inspired by the way inconsistency seems to be handled by human beings we propose a model that tries to reason with two datasets: a first potentially inconsistent one representing the *LTM* and a second consistent one for representing the *WM*.

Handling of inconsistent knowledge bases is a thoroughly studied subject in computer science, and in particular in the domains of Databases (with Repairs (Greco et al. 2003), Consistent Query Answering (Chomicki 2006)) and Knowledge Representation and Reasoning (SAT, repairs, revision, argumentation), see chapters (Amgoud et al. 2020; Dubois et al. 2020). In general, such an inconsistency is addressed by either repairing directly the knowledge base, which typically lead to loss of information (Doder and Vesic 2015; Bertossi 2006)), or by considering that the query is entailed if it is in the intersection of all the minimal repairs (or maximal maxi-consistent subbases). This last approach, while avoiding the loss of information, is hampered by the computational price required to compute all the repairs or subbases. Another family of approaches for inconsistency handling, called paraconsistent logic, is outside the scope of this paper since these approaches either rely on extra-information (e.g. possibilistic logic (Dubois and Prade 2014)) or are based on non-classical logical axioms (see the surveys (Avron, Arieli, and Zamansky 2018; Carnielli and Coniglio 2016)).

In this paper, we use ideas from psychology as an inspiration to provide a system able to handle inconsistent knowledge base without needing to compute all the maximally consistent subbases. Indeed, the notion of *WM* allows for the computation of one "repair" in the sense that only relevant and consistent pieces of information from the knowledge base are activated, effectively compartmentalizing the inconsistency. The paper will present a way of recursively selecting which pieces should be activated based on some heuristic, and how it will affect the reasoning, in complexity and in execution time, notably when the *WM* is restricted in size to account for bounded rationality (Simon 1955).

We are in particular interested in the impact of successive querying in this kind of context where results might differ due to the way the size-limited subbase is built. In order to experiment our approach, we will place ourselves in the context of SAT. Indeed, SAT is a deeply studied domain with powerful solvers (see the competition (Järvisalo et al. 2012)). On that topic, it should be noted that within the SAT domain, handling inconsistency can be done thanks to (weighted) MAXSAT (Li and Manya 2009). While this is a relevant approach, it needs either the removal cost of each clause, which needs to be elicited, or to consider that every clause is as relevant to the query as any other.

In this work, we would like to study another, somehow close, way where the selection of clauses is based on their closeness to a given query.

The paper is organized as follows: we start by recalling inconsistency handling classical approaches, then we describe the Working Memory approach. In the last section we demonstrate theoretical results about this approach and study its complexity, then we describe the experiments that were conducted on several benchmarks. We conclude by a discussion about MAXSAT and about the non-monotonic properties of the inference relation based on the Working Memory approach.

2 Background on selection-based inconsistency handling

Notations: we consider a propositional logical language $\mathscr{L}$ containing formulas denoted by lower case Greek letters, based on a vocabulary $\mathscr{V}$ of variables denoted by Latin lower case letters. Negation, conjunction, disjunction, material implication, contradiction and classical inference are denoted respectively by $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\bot$ and $\vdash$. A CNF formula is a conjunction of clauses, a clause is a disjunction of literals, a literal is a variable or its negation. Abusing notation, clauses are assimilated to sets of variables when using the two set operators $\in$ and $\cap$ (for membership and intersection). Lists of elements are represented with square brackets, and :: is the operator s.t. e::L is the new list formed by the element e followed by the elements of L.

In this paper we are going to use finite knowledge bases defined as follows:

Definition 1 (Knowledge base)**.** *A knowledge base is a finite set of formulas of $\mathscr{L}$, considered as the conjunction of its elements.*

One of the best known approach to cope with inconsistency is the one of Rescher and Manor (Rescher and Manor 1970): it is based on the computation of the set of maximal consistent subsets of the belief base, then a formula is accepted as a consequence of the base if it can be classically inferred from every maximal consistent subset (or MSS for maximum satisfiable subset). This idea has been refined in the preferred subtheory approach of (Brewka 1989) where the knowledge base is divided into several subsets according to a given reliability level. We will see in Section 3.2 that our approach actually allows to dynamically compute this reliability levels with respect to a given query.

Definition 2 (Preferred subtheory (Brewka 1989))**.** *Given a tuple $T = (T_1, \ldots, T_N)$ of sets of formulas*[1] *of $\mathscr{L}$, $S = S_1 \cup \cdots \cup S_N$ is a* preferred subtheory *of T* iff *for all k, $(1 < k < N)$ $S_1 \cup \cdots \cup S_k$ is a maximal consistent subset of $T_1 \cup \cdots \cup T_k$.*

In the words of Brewka: "in order to obtain a preferred subtheory of T we have to start with any maximal consistent subset of T_1, add as many formulas from T_2 as consistently can be added (in any possible way), and continue this process for $T_3, \ldots, T_N$".

Computing preferred subtheories requires some kind of inconsistency checking; in this paper, we will make use of the notion of minimal unsatisfiable subsets[2].

Definition 3 (MUS and MSS (Liffiton and Sakallah 2008))**.** *A subset S of clauses of a base B is a minimal unsatisfiable subset (MUS) if S is inconsistent and for all $c \in S$, $S \setminus \{c\}$ is consistent.*

A subset $S \subseteq B$ is a maximal satisfiable subset MSS if S is consistent and for all $c \in B \backslash S$, $S \cup c$ is inconsistent.

Example 1. *Let us consider the following knowledge base LTM (in CNF form):*

$$LTM = \overbrace{(a \vee \neg d)}^{c_1} \wedge \overbrace{\neg a}^{c_2} \wedge \overbrace{(\neg a \vee b \vee d)}^{c_3} \wedge \overbrace{\neg b}^{c_4} \wedge \\ \overbrace{(\neg a \vee c)}^{c_5} \wedge \overbrace{\neg c}^{c_6} \wedge \overbrace{d}^{c_7}$$

There are two MUSes of LTM: $\{c_1, c_2, c_7\}$ and $\{c_1, c_5, c_6, c_7\}$. There are four MSSes of LTM: $\{c_1, c_2, c_3, c_4, c_5, c_6\}$, $\{c_1, c_3, c_4, c_5, c_7\}$, $\{c_1, c_3, c_4, c_6, c_7\}$ and $\{c_2, c_3, c_4, c_5, c_6, c_7\}$.

When the user has information about the formulas that are more important/sure (called "preferred" in (Brewka 1989)) then the selection can be done among the preferred subtheories (which according to (Brewka 1989) are maximal consistent subbases of the knowledge base in which the most important formulas are primarily chosen). Nebel in (Nebel 1991) also proposes to use a syntax based approach that he calls "epistemic relevance", which is a complete preorder on all the formulas that are consequences of the beliefs. This relevance/preference information may come from the confidence given into the different sources of the belief base, it is then considered as exogeneous extra information about the beliefs. Another kind of approach takes profit from the syntax of the belief base to discover the strength of each belief, this meta information is then endogeneous with respect to the belief base: for instance System Z is able to rank automatically the beliefs based on their specificity (Pearl 1990).

When a user wants to conserve the belief base without forcing consistency, two ways can be adopted: either reason on ALL the most interesting subbases (for instance, reason on Brewska's preferred subtheories (Brewka 1989), providing that rankings on beliefs are available, or simply maximal consistent subbases when no extra-information is available) or select only ONE preferred consistent subbase and reason

[1] Preferred subtheories are originally defined of a first-order language; in this paper, we will restrict this defining by using a propositional language.

[2] Please note that our approach is agnostic on that point and might use different inconsistency checking mechanisms. That being said, while computing MUS is computationally expensive, they need to be computed just once, which would not potentially be the case with other inconsistency checking methods where the computation would be needed with each newly considered clauses.

with it (Benferhat, Dubois, and Prade 1995; Benferhat et al. 1993). In both cases, the whole initial base is preserved, but the reasoning process is made on one (or several) of its consistent part(s) (called "repair(s)" in Query Answering community (Greco et al. 2003; Chomicki 2006)).

Other, somewhat different, approaches for reasoning under inconsistency exist; for instance, modifying the beliefs by adding premises in order to specify better the context in which some rules should not be fired (Doder and Vesic 2015; Dupin De Saint Cyr, Duval, and Loiseau 2001), somehow coming back to the old idea of circumscription (McCarthy 1986).

3 Working Memory

In this section we are going to present a way to assess a CNF formula called the query, denoted φ, w.r.t. a potentially inconsistent finite CNF knowledge base called *LTM* (for long term memory). This is done by checking the consistency of the formula w.r.t. a subset of the *LTM* called *WM* (for working memory).

More precisely, we propose a process that: 1) links the *LTM* clauses together based on the number of common literals (Section 3.1), 2) uses these links to build a *WM* that is relevant for the query φ to assess, where relevance is understood in terms of common variables (Section 3.2) and 3) actually checks the status of the query (or queries, Section 3.3).

3.1 Preprocessing on LTM

This first step of preprocessing on LTM extracts the information required to build a consistent and relevant *WM*. More precisely, it consists in creating a dictionary *AssocCl* that associates to each clause c the list of clauses that contains at least one common variable with c together with the number of common variables:

$$\begin{aligned} AssocCl(c) = [(c', nbV) \quad | \quad & c' \in LTM \setminus \{c\}, \\ & nbV = |c \cap c'| \text{ s.t.} \\ & nbV > 0] \end{aligned}$$

This is done by Algorithm 1 which also computes the maximum number of common variables *maxCom* between any pair of clauses and the dictionary $Var2Cl$ which maps each variable v to the set $Var2Cl(v)$ of clauses in which it appears:

$$Var2Cl(v) = \{c \quad | \quad c \in LTM \text{ s.t. } v \in c\}$$

Please note that these association tables do not need to be recomputed for each query (but they can be updated).

Example 2. *In Example 1, $Var2Cl(a) = \{c_1, c_2, c_3, c_5\}$ and table AssocCl is: $AssocCl(c_1) = [(c_2, 1), (c_3, 2), (c_5, 1), (c_7, 1)]$, $AssocCl(c_2) = [c_1, 1), (c_3, 1), (c_5, 1)]$, $AssocCl(c_3) = [(c_1, 2), (c_2, 1), (c_4, 1), (c_5, 1), (c_7, 1)]$, $AssocCl(c_4) = [(c_3, 1)]$, $AssocCl(c_5) = [(c_1, 1), (c_2, 1), (c_3, 1), (c_6, 1)]$, $AssocCl(c_6) = [(c_5, 1)]$, $AssocCl(c_7) = [(c_1, 1), (c_3, 1)]$ and $maxCom = 2$.*

```
Algorithm 1: LTM_preprocessing(LTM)
Input:  LTM in Dimacs CNF format
Output: Var2Cl: dict. of clauses associated with
        var; maxCom: max nb of common vars in
        2 clauses; AssocCl: dictionary of
        associated clauses
Var2Cl ← empty dictionary
for each clause c in LTM do for each v in c do
Var2Cl($v$) ←Var2Cl($v$) ∪ {$c$}
maxCom ← 0; AssocCL ← empty dictionary
for each ($c_1$,$c_2$) in $LTM^2$ s.t. $c_1 \cap c_2 \geq 1$ do
    AssocCl($c_1$) ← ($c_2$,$c_1 \cap c_2$) :: AssocCl($c_1$)
    if $maxCom < c_1 \cap c_2$ then
        maxCom ← $c_1 \cap c_2$
return (Var2Cl, maxCom, AssocCl)
```

```
Algorithm 2: Query_preprocessing($\varphi$,LTM)
Input:  $\varphi$: a query in CNF format; LTM: set of
        formulas in CNF format
Output: QAssocV: dict. of common var of each
        LTM clause w.r.t. $\varphi$
QAssocV ← empty dictionary of clauses
for each clause c in LTM do
    for each clause i in $\varphi$ do
        comV ← $i \cap c$  // common vars between i and c
        if $comV \neq \emptyset$ then
            QAssocV($c$) ← QAssocV($c$) ∪ comV
return (QAssocV)
```

3.2 Building a consistent *WM* for assessing φ

Given a query φ and a LTM, Algorithm 2 builds the dictionary $QAssocV$ that associates to each clause c of the LTM the set $QAssocV(c)$ of variables that c has in common with φ:

$$QAssocV(c) = c \cap \varphi$$

This dictionary will be used (in Algorithm 3) to start feeding the *WM* with clauses directly linked to the query, since as detailed below, the score of a clause is the sum of the number of common variables, this number is normalized by dividing it by the maximum number obtained for an entry of the dictionary.

Example 3. *Let us consider the following formula containing two clauses: $\varphi = (b \vee \neg c) \wedge d$. $QAssocV(c_1) = QAssocV(c_7) = \{d\}$ meaning that c_1 (and also c_7) has the variable d in common with φ. $QAssocV(c_3) = \{b, d\}$, $QAssocV(c_4) = \{b\}$, $QAssocV(c_5) = QAssocV(c_6) = \{c\}$.*

We propose a best-first-search algorithm (Algorithm 3) for filling the Working Memory with the clauses related to a query φ. Technically, the idea is to select the "closest" clauses w.r.t. to φ, with the closeness notion defined inductively as follows: first the clauses that have a maximum number of common variables with φ are inserted in a maximal binary heap[3] with a *percentage of*

[3] A maximal binary heap is represented by a tabular

Algorithm 3: WM_download

```
Input:   QAssocV: dict. of vars assoc. with LTM
         clauses; AssocCl: dict. of associated
         clauses in LTM; MUS: set of MUS of
         LTM; $m$: capacity size of $WM$; maxCom:
         max common vars between 2 clauses
Output:  new $WM$
/* Create a max binary heap with clauses associated
   with their score w.r.t. $\varphi$ */
Queue ← empty maximal binary heap
maxScore ← 0
for each key c in QAssocV do
    score(c) ← |QAssocV(c)|
    if maxScore < score(c) then maxScore ←
      score(c)
for each key c in QAssocV do
  Add(Queue,(c,score(c)/maxScore))
WM ← empty set; InconsSeen ← empty set
/* Best first search algorithm */
while Queue not empty and $|WM| < m$ do
    (key,rel) ← Remove(Queue)      // Removing the
      max element of the heap
    if $key \notin WM$ then
        if $\exists mus \in MUS, mus \subseteq WM \cup \{key\}$ then
          // key is inconsistent with WM
            $InconsSeen \leftarrow InconsSeen \cup \{key\}$
        else // key is consistent with WM
            $WM \leftarrow WM \cup \{key\}$
            keyAdjacents ← AssocCl(key)
            for each pair $(c, s)$ in keyAdjacents do
                if $c \notin WM$ and $c \notin InconsSeen$ then
                    Add(Queue,
                      $(c, rel \times s/maxCom)$)
return (WM)
```

relevance to the query (the number of common variables normalized with the maximal score *maxScore*, the biggest number of associated clauses). A clause c with highest relevance is then taken out of the heap and added to the *WM* (if not inconsistent); its closest clauses (according to AssocCl) c' are added to the heap with a *percentage of relevance to the query* equal to the relevance of c multiplied by its degree of relevance with c' (number of common variables with c' divided by $maxCom$ the maximum of common variables between two clauses in *LTM*). Using percentage and normalized score ensures that the "farther" from the query a clause is, the lower its relevance to the query will be.

More fundamentally, Algorithm 3 presents a way to compute a relevance score which plays a similar role as Brewka's reliability level, with the significant difference of being dynamically computed. Indeed, first, the set of clauses with the highest relevance to the query, i.e. with the highest number of common variables, is selected to form the first stratum of the knowledge base LTM_1; from this stratum is extracted a set of maximally consistent clauses WM_1. A new relevance score is then computed for the remaining clauses of the *LTM* based on their relevance with the clauses in WM_1, essentially computing a transitive relevance to the query, and forming LTM_2. This process continues recursively until no new stratum can be formed, either because the maximal size of the *WM* has been reached or because there is no relevant clause anymore.

More formally, the selection of the relevant clauses requires the notion of consistent sets of clauses that are maximal for inclusion with the condition that they have a size under a given bound s, defined as follows:

Definition 4 (Max-consistent for inclusion under s)**.** *S is a max-consistent subset of K for inclusion under $s \in \mathbb{N}$ (S mci_s K) iff $S \subseteq K$ and S is consistent and $|S| \leq s$ and there is no S' consistent s.t. $S' \subseteq K$ and $|S'| \leq s$, $S' \supset S$.*

Algorithm 3 recursively collects clauses that are less and less (transitively) relevant to the query, yielding a consistent set *WM*, more formally defined as follows:

Definition 5 (Working memory w.r.t. a query)**.** *Given a knowledge base LTM and a formula φ (called the query), a working memory $WM(LTM, \varphi, m)$ associated with LTM and φ given a maximum size m of the working memory is recursively defined as follows:*

$$score_1(c) \quad = \quad |\bigcup_{c' \in \varphi}(c' \cap c)|, \qquad c \in LTM$$

$$LTM_1 \quad = \quad \text{argmax}_{\substack{c \in LTM \\ and\ score_1(c) \neq 0}} \qquad score_1(c)$$

$$WM_1 \quad mci_m \quad LTM_1$$

Given $WM_1, \ldots, WM_k$, and $LTM_1, \ldots, LTM_k$ and $m(k) = m - \sum_{i=1}^{k} |W_i|$:

- *If $m(k) > 0$ and $WM_k \neq \emptyset$ then*

  $$score_{k+1}(c) = \max_{c' \in WM_k} (|c' \cap c| \times score_k(c')/maxCom)$$

  $$where\ maxCom = max_{c,c' \in LTM}|c \cap c'|.$$

  $$LTM_{k+1} = \text{argmax}_{\substack{c \in LTM \setminus (LTM_1 \cdots LTM_k) \\ and\ score_{k+1}(c) \neq 0}} score_{k+1}(c)$$

  $$WM_{k+1} \quad mci_{m(k)} \quad LTM_{k+1}$$

- *Else $WM(LTM, \varphi, m) = \bigcup_{j=1}^{k} WM_j$*

As it can be seen in Definitions 4 and 5 with the notion of max-consistent subset for inclusion under s (mci_s), consistency must be maintained when building the *WM*. In Algorithm 3, we use MUS to ensure consistency, more precisely, each time a new clause is added to the *WM* we check whether a MUS is not a subset of the *WM*. The MUS of the LTM are precomputed offline thanks to a standard algorithm (we have chosen to use the system CAMUS (Liffiton and Sakallah 2008)). Note that another technique would be to use an incremental SAT solver like GlucoseInc (Audemard, Lagniez, and Simon 2013), or even just check for consistency each

where the root is at index 1, the left child of any node i is at index $2i$ and the right child is at index $2i + 1$.

time a new clause is added to the *WM*. The comparison, in terms of complexity and computation time, of these different consistency handling techniques is left for future work.

Example 4. *Let us consider that we want to build a WM of size $m = 5$ extracted from the LTM of Example 1 for the query $\varphi = (b \vee \neg c) \wedge d$. The scores of the clauses indexed in QAssocV are: $score(c_1) = score(c_4) = score(c_5) = score(c_6) = score(c_7) = 1$ and $score(c3) = 2$ (maxScore), yielding to an initial Queue consisting in the maximal binary heap $[(c_3, 1)$; $(c_1, 0.5)$; $(c_4, 0.5)$; $(c_5, 0.5)$; $(c_6, 0.5)$; $(c_7, 0.5)]$. Algorithm 3 starts building a WM by removing the maximal element (the clause that is the most related to the query) of the heap, namely c_3, and adding it to WM. After this step, $WM = \{c_3\}$ and $Queue = [(c_7, 0.5)$; $(c_1, 0.5)$; $(c_4, 0.5)$; $(c_5, 0.5)$; $(c_6, 0.5)]$*[4]*.*

The clauses associated to c_3 are already stored in $AssocCl(c_3) = \{(c_1, 2), (c_2, 1), (c_4, 1), (c_5, 1), (c_7, 1)\}$, each of them is added to the Queue with a weight equal to the value of c_3 (which equals 1) times their weight divided by maxCom (which equals 2), yielding the new Queue: $[(c_1, 1)$; $(c_1, 0.5)$; $(c_7, 0.5)$; $(c_5, 0.5)$; $(c_6, 0.5)$; $(c_4, 0.5)$; $(c_2, 0.5)$; $(c_4, 0.5)$; $(c_5, 0.5)$; $(c_7, 0.5)]$. Note that the queue may contain several occurrences of the same element. Then c_1 is removed from Queue and added to the WM, afterwards c_4 then c_7 and c_5. Finally when c_2 is at the top of the heap, it cannot be added since inconsistent with WM idem for c_6. At the end $WM = \{c_1, c_3, c_4, c_5, c_7\}$ (which is actually a max consistent subset of B, it is not necessarily the case that a whole MSS is obtained). Note that due to equalities other WM are obtainable (more precisely, every subset of size 5 of any MSS except the subsets of $\{c_2, c_3, c_4, c_5, c_6, c_7\}$ (since c_1 should be present due to its high number of common variables with the query).

3.3 *WM* loading with overflow for new queries

Once a *WM* has been built, it is possible to evaluate the query. In particular, we will say that a query φ is accepted when $WM \cup \{\varphi\} \not\models \bot$. Note that we used Sat4J (Le Berre and Parrain 2010) to check satisfiability, but any other SAT solver could be used.

Example 5. *Given $WM = \{c_1, c_3, c_4, c_5, c_7\}$ with $\varphi = (b \vee \neg c) \wedge d$, we get $WM \cup \{\varphi\} \models \bot$.*

When a new query φ' arrives, the previously loaded clauses might be irrelevant, i.e. there might be no association between the query and clauses in the *WM* ($AssocCl \cap QAssocV = \emptyset$), which prompts for the loading of other clauses. Then two cases might arise according to the room left in the current *WM* (where $room = m - |WM|$, i.e. capacity size of the *WM* minus current occupation) and to the number of clauses needed to answer query φ' ($needed = |WM(LTM, \varphi', m)|$):

- either the *WM* still has room to store the newly needed clauses: $needed \leq room$, in that case the process is the same as before,
- or the *WM* lacks room: $needed > room$ then a set of old clauses (of size $needed - room$) is discarded from the *WM*.

4 Characterization about efficiency in time and accuracy

In this section, we will study some properties of the inference relation induced by our framework and assess it experimentally.

4.1 Theoretical results

Before getting into the details of the inference relation, we need to define the notion of cluster.

Definition 6 (Clusters)**.** *Given a set of clauses LTM, a* cluster *of the LTM is a set of clauses composing a connected component of the graph whose vertices are the clauses and the edges are relating two clauses with at least one common variable; clusters(LTM) is the set of clusters of LTM.*

Now, we are in the position to define the inference relation with regards to the *WM*.

Definition 7 (*LTM* inference)**.** *Given a LTM and a capacity size m of the WM,*

$$\alpha \mid\!\sim^{m}_{LTM} \beta \text{ is defined by } WM(LTM, \alpha, m) \vdash \alpha \rightarrow \beta$$

where $WM(LTM, \alpha, m)$ is a working memory in the sense of Definition 5 and $\vdash$ is classical logic inference.

The following proposition establishes that detecting inconsistency of the *LTM* with the query φ by selecting a consistent subbase with no limit of size is the same as doing it with a size equal to the size of the maximal cluster of the *LTM*. The proposition holds when the query φ is related to only one cluster of the *LTM*, in other words when φ concerns only one domain of knowledge (i.e., associated to only one vocabulary).

Proposition 1. *Let $mc = \max_{C \in clusters(LTM)} |C|$, if $mc \leq m$ (where m is the capacity of the WM), and $\alpha \in \mathscr{L}$ s.t. there is only one cluster $C \in clusters(LTM)$ where for all clause i in α, for all cluster C' in $clusters(LTM) \setminus C$ and for all clause $c \in C'$, $i \cap c = \emptyset$:*

$$\alpha \mid\!\sim^{\infty}_{LTM} \beta \quad \textit{if and only if} \quad \alpha \mid\!\sim^{mc}_{LTM} \beta$$

Proof. Due to Definition 7, the proof is based on the definition of $WM(LTM, \alpha, m)$ which returns a consistent sub-base *WM* such that by construction all its clauses belong to the same cluster of the *LTM* (since in WM_1 the clauses have at least one common variables with α, then WM_2 is a set of clauses that have at least one common variables with WM_1 and so on). Moreover m being big enough to contain any cluster of the *LTM*, downloading is limited to at most mc formulas, hence $WM(LTM, \alpha, \infty) = WM(LTM, \alpha, mc)$. □

[4] Removing the max element of a heap replaces it with the last leave of the tree and percolate it down to its right place, here c_7 go to the top and stays there.

The following propositions guarantees that the rejection of a query by using Working Memory is in accordance with the result that could be obtained by selecting a maximal consistent subbase of the *LTM*.

Proposition 2. *For all $m > 0$, if $\alpha \mid\!\sim^{m}_{LTM} \beta$ then there is a maximal consistent subbase B of the LTM s.t. $B \vdash \alpha \rightarrow \beta$*

Proof. By construction, for any m, $WM(LTM, \varphi, m)$ is a consistent subbase, thus there is maxi-consistent subbase of the *LTM* that contains it. Hence the result due to the monotonicity of $\vdash$. □

In the following, we compute the worst case time complexity denoted $T_{\max}$ associated to the processing of k consecutive queries, where processing a query means to check its consistency w.r.t. the current *WM*. The complexity of this processing is expressed w.r.t. the number of variables n and the number of clauses in the LTM denoted m_L, the capacity of *WM* in clauses is denoted m and the number of queries k of m_q clauses. When the process is done on the LTM only, the complexity is denoted $T_{\max}(LTM(n, m_L, k, m_q))$ while the one of the process done with a *WM* given the *LTM* is denoted $T_{\max}(\mathrm{WM}(n, m_L, m, k, m_q))$.

As recalled in (Pătraşcu and Williams 2010) there is a sequence of papers that have provided algorithms for CNF_SAT with $2^{n-o(n)}.poly(m)$ runtime, where n is the number of variables and m is the number of clauses and $poly(m)$ is a polynomial function of m. The current best (Calabro, Impagliazzo, and Paturi 2006) is a deterministic algorithm that runs in $2^{n(1-\frac{1}{O(\log(m/n))})}poly(m)$ time, as shown by (Dantsin and Hirsch 2009). Applying these results in our context, the following remark shows the worst case computational complexity of checking k queries of m_q clauses in the *LTM* (please note that the expression is simpler when we assume that the number of queries and their size is negligible in front of the size of the *LTM*).

Remark 1. *If $k.m_q \ll m$ then $T_{\max}(LTM(n, m_L, k, m_q)) \in \Theta(k \times poly(m_L) \times 2^{n\alpha(n,m_L)})$ where $\alpha(n, m_L) = 1 - \frac{1}{O(\log(m_L/n))}$.*

This is due to the fact that $T_{\max}(LTM(n, m_L, k, m_q)) = \sum_{i=1}^{k}(T_{\max}(SAT(n, m_L + i.m_q)))$.

Due to Algorithms 1, 2 and 3, assessing k queries of m_q clauses via the *WM* has the following worst case computational complexity.

Proposition 3. *If $m_q \ll m$ then $T_{\max}(WM(n, m_L, m, k, m_q)) \in \Theta(n.m_L^2 + k.m_L.m^2 + k.poly(m) \times 2^{n.\alpha(n,m)})$*

Proof. $T_{\max}(WM(n, m_L, m, k, m_q)) =$
$T_{\max}(LTMprep(n, m_L))$ +

$$k \times \left(\begin{array}{l} T_{\max}(\mathrm{Qprep}(n, m_L, m_q)) + \\ T_{\max}(WMdl(n, m_L, m, m_q)) + \\ m \times (T_{\max}(pop + push)) \\ +T_{\max}(SAT(n, m + m_q)) \end{array} \right)$$

where push and pop are the operations that respectively add and delete an element from a fifo (here they are used to delete and add clauses to the *WM* since in the worse case m clauses have to replace all the clauses that were in the *WM* before, these operations can be implemented in $\Theta(1)$), *LTMprep*, Qprep, *WMdl* are the respective abbreviations for the Algorithms *LTM*_preprocessing (Algo 1), Query_preprocessing (Algo 2) and *WM*_download (Algo 3).
Moreover $T_{max}(LTMprep(n, m_L)) \in \Theta(n.m_L^2)$, since it computes the AssocCl dictionnary of the m_L clauses. $T_{max}(Qprep(n, m_L, m_q)) \in \Theta(m_L.m_q.n)$ considering that the intersection of two clauses is done in linear time of the number of variables (the clause literals being ordered) and this intersection being done between all the m_L clauses of the *LTM* and all the m_q clauses of the query. $T_{max}(WMdl(n, m_L, m, m_q)) \in \Theta(m_q.m_L + m.(m + m.m_L + m\log(m)))$ (since the while is done at worst m times and runs one membership test (in $\Theta(m)$), one inclusion test to the MUS list (in $\Theta(m.m_L)$ assuming that the clauses in the MUSes are ordered), one `Remove` and at worst m `Adds` to a heap of capacity size m which are both in $\Theta(\log m)$). After simplification, we get $T_{max}(WMdl(n,\ m_L,\ m,\ m_q) \in \Theta(m^2.m_L)$. Wrapping it up yields $T_{\max}(WM(n, m_L, m, k, m_q)) \in \Theta(n.m_L^2 + k(m_L.m_q.n + m^2.m_L + poly(m + m_q) \times 2^n)))$, we finally obtain the result. □

Function QRandGener(LTM,$maxV$,$maxCl$)

```
Require: RandNum(M): returns number in interval [1, M] and Sample(S,n): returns n random
         elements from S and Vars(S): returns the variables of clauses set S
Input:   LTM: set of formulas in CNF format;
         maxV: maximum size of a query clause;
         maxCl: maximum number of clauses in query
Output:  $\varphi$: query in CNF
counterCl $\leftarrow$ 0; $\varphi \leftarrow$ empty list of clauses
for counterCl in [1,RandNum(maxCl)] do
    clause $\leftarrow$ empty list of literals
    vars $\leftarrow$ Sample(Vars(LTM),RandNum(maxV))
    for each v in vars do
        newLit $\leftarrow$ RandChoice($\{\overline{v}, v\}$);
        clause $\leftarrow$ newLit :: clause
    $\varphi \leftarrow$ clause :: $\varphi$
return ($\varphi$)
```

Hence if we ignore the first preprocessing of the *LTM*, comparing Rem. 1 and Prop. 3 leads to a theoretical gain in time in the worst case provided that $m \ll m_L$, *i.e.*, when the size of the *WM* is small w.r.t. the size of the LTM. This is confirmed by the following empirical results.

4.2 Empirical results

In order to assess the empirical interest of our approach, we implemented[5] the approach and placed ourselves in the situations where the base is receiving several consecutive queries. These queries are randomly generated from the clauses in the base (see Function QRandGener). Intuitively, the function creates a random number of clauses (bounded by a parameter *maxCl*) that are populated by a random number of variables (bounded by a parameter *maxV*) that appear in some clauses of the base (or a specific cluster); each of these variables is then randomly set positive or negative. Please note that it is possible to generate queries on a specific cluster of the base by replacing Vars(*LTM*) line 5 by Vars(*Cluster*); clusters are computed by transitive closure of the neighborhood relation between clauses (where neighbor means to have common variables, see Definition 6).

In order to assess our approach, we observed its results under different maximal sizes for the WM. As a baseline, a WM representing the most relevant maxi-consistent subbase is computed (*WM MaxiCons*); this WM is created by associating the query with all the clauses from the base (assigning 0 in case no variables are shared) and by using the regular *WM_download* algorithm. We then run the experiment with *WM* having respectively the size of the *LTM* (*WM* LTMSize), the size of the cluster of clauses of maximum size (*WM* MaxCluster) and the size of the average size of all the clusters of clauses (*WM* AverageCluster).

Table 1 summarizes the results for different bases built on two Dimacs files coming from the SAT benchmark Blocks World[6]: *mdi* and *ami* are respectively Medium.cnf cut off after 150 lines and Anomaly.cnf cut off after 50 lines, both of them made inconsistent by negating their first clause; *mdiX* and *amiX* are the files obtained by repeating the *mdi* and *ami* X times (literals are renamed to avoid redundancy); *uma* is the union of *mdi2* and *ami4*. Hence, *adi60*, *mdi20* and *um6* are all composed of 3000 clauses.

The notation qXY indicates that the generated query has a maximum number of clauses of X and a maximum number of variables per clause of Y. (1st) and (5th) indicates respectively that the row corresponds to the first or the last queries of five successive queries.

Based on the results on the table, we can make the following remarks. Bases that have several connected components benefit from the approach when the query concerns a limited amount of these components, since the solver will be executed on a much smaller base which will reduce the execution time while maintaining accuracy. We argue however that it is fair to assume that a general base will have a tendency to cluster its formulas, where each cluster represents some sort of "context" or "domain".

Iterative querying on the same cluster allows to reduce of lot of the overhead caused by the approach since the *WM* does not need to be recomputed. On the other hand, iterative querying on the whole base forces the re-computation of the *WM* quite often, which implies longer execution times. In that case, the accuracy may decrease since the working memory may be overwhelmed by the number of subjects that must be covered at the same time; it is interesting to note that this behavior is somewhat reminiscent of human behavior, for instance when a person, by mixing different subjects and domains, makes it impossible to apprehend the full extent of her statement.

Finally, as expected, let us note that the choice of a suitable *WM* capacity is a matter of compromise between time and correctness: a bigger size will give more correct results but will take more time to compute whereas a smaller size may be less correct but faster. That being said, as Table 1 shows, selecting a size equal to the average of the clusters demonstrates noticeable decrease in time while maintaining good results.

5 Conclusion

In this paper, we presented an approach to handle inconsistency in a knowledge base by using a notion of associations between clauses based on common variables. These associations are used to extract one consistent subbase. We showed that this approach has interesting results both in terms of complexity and execution time.

It should be noted that our approach, by eliciting only one subbase, may give results that are different from some classical approaches that consider all the consistent subbases. Moreover, depending on the history of the knowledge base, i.e. the sequence of queries that happened beforehand, different consistent subbases can be chosen. In addition, we can remark that our approach behaves in a different way than MAXSAT (which also selects only one subbase). This can be observed in the following small example:

Example 6. *Let us consider a KB LTM′ in CNF form:*

$$LTM' = \begin{array}{l} \overbrace{(a \vee b \vee c)}^{c_1} \wedge \overbrace{\neg a \vee \neg d}^{c_2} \wedge \overbrace{(a \vee \neg c \vee \neg d)}^{c_3} \wedge \\ \overbrace{\neg b \vee c}^{c_4} \wedge \overbrace{(\neg d \vee \neg e)}^{c_5} \wedge \overbrace{a \vee e}^{c_6} \wedge \overbrace{c \vee e}^{c_7} \wedge \\ \overbrace{(\neg c \vee d)}^{c_8} \wedge \overbrace{(\neg a \vee c)}^{c_9} \wedge \overbrace{(\neg e)}^{c_{10}} \end{array}$$

Let $\varphi = (c \vee d) \wedge (\neg b)$. In order to minimize the number of deleted clauses, MAXSAT would find a solution by removing only the clause c_8, and φ would be accepted. On the contrary, our approach would create the WM by selecting all the clauses but c_9 and c_{10} and reject the query. In that example, it seems more desirable to exclude c_9 and c_{10} since they are less related to the query than c_8.

[5] Using Python 3.9.2, Sat4J 2.3.5 and CAMUS 1.0.7.

[6] https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/PLANNING/BlocksWorld/descr.html

Query type	Filename	*WM MaxiCons* (Tbuild, Tsat)	*WM LTMSize* (Tbuild, Tsat)	*WM MaxCluster* (Tbuild,Tsat)	*WM AverageCluster* (Tbuild,Tsat)
q11	adi60 (1st)	(1299.02,17.8)	(3.3,6.1)	100% (2.43,6.1)	100% (2.32,6.1)
	adi60 (5th)	(2.03,14.3)	(1.72,5.8)	100% (1.16,5.9)	100% (1.22,5.8)
	mdi20 (1st)	(1299.02,17.8)	(7.84,7.2)	100% (8.68,7.1)	100% (8.25,7.3)
	mdi20 (5th)	(1.51,17.5)	(1.32,7.2)	97% (1.36,7.1)	97% (1.41,7.1)
	uma6 (1st)	(1181.35,16.4)	(4.02,6.6)	100% (5.03,6.6)	100% (3.03,6.5)
	uma6 (5th)	(1.29,14.9)	(1.08,6.5)	100% (1.33,6.4)	97% (1.51,6.3)
q33 on same cluster	adi60 (1st)	(1034.53,15.8)	(3.54,6.0)	100% (2.85,6.0)	100% (2.53,6.0)
	adi60 (5th)	(2.72,15.5)	(1.6,5.8)	98% (1.36,5.9)	98% (1.25,5.9)
	mdi20 (1st)	(1304.62,18.2)	(8.08,7.2)	100% (7.86,7.3)	100% (8.44,7.2)
	mdi20 (5th)	(2.17,17.3)	(1.4,7.0)	99% (1.32,7.1)	99% (1.58,7.2)
	uma6 (1st)	(1166.63,16.6)	(6.14,6.7)	100% (5.15,6.6)	99% (2.81,6.5)
	uma6 (5th)	(2.37,16.1)	(1.31,6.2)	100% (1.25,6.2)	92% (1.58,6.3)
q33 on different clusters	adi60 (1st)	(1025.18,15.3)	(14.36,7.2)	83% (3.02,6.6)	83% (2.89,6.5)
	adi60 (5th)	(2.53,14.9)	(5.74,9.4)	85% (3.1,6.6)	85% (2.73,6.6)
	mdi20 (1st)	(1285.47,18.1)	(57.22,9.1)	89% (6.19,7.6)	89% (6.79,7.5)
	mdi20 (5th)	(2.1,17.5)	(13.74,11.8)	89% (5.56,7.8)	89% (5.86,7.6)
	uma6 (1st)	(1174.62,16.9)	(36.65,8.6)	90% (7.02,7.4)	86% (4.66,7.0)
	uma6 (5th)	(2.22,16.5)	(9.15,11.5)	92% (5.68,7.6)	94% (4.17,7.0)

Table 1: Agreement ratio between differently sized *WM* for 100 runs. Percentages in the cells correspond respectively to agreement ratio between the *WM LTMSize* and, respectively, *WM MaxCluster* and *WM AverageCluster*. Tbuild and Tsat represent respectively the time (in ms) to build the *WM* and to execute the sat solver.

This first preliminary study opens several research avenues:

Finer WM building: extending the relevance between clauses by being able to compare them semantically instead of just counting their common variables (see e.g. (Bisquert et al. 2017) where associated formulas are built on the results of a serious game) could overcome the drawbacks related to the syntax dependency of the non-monotonic inference relation $\mid\!\sim^{\varphi}_{LTM}$. It is important to note that, with the current syntactic definition of relevance, different sets of clauses may be relevant to two equivalent clauses (e.g. by disjunctively adding superfluous literals): for instance, consider the clauses c and $a \vee \neg a \vee c$. On that note, the reader can check that $\mid\!\sim^{m}_{LTM}$ satisfies some classical properties of non-monotonic inference relations of (Kraus, Lehmann, and Magidor 1990) like reflexivity ($\alpha \mid\!\sim^{m}_{LTM} \alpha$) and right weakening (if $\vdash \alpha \rightarrow \beta$ and $\gamma \mid\!\sim^{m}_{LTM} \alpha$ then $\gamma \mid\!\sim^{m}_{LTM} \beta$). However, left logical equivalence, cut or cautious monotony[7] are not guaranteed since two equivalent formulas may imply different WM downloading. Other, more semantical, definitions of relevance between clauses may allow for the satisfaction of more non-monotonic properties, for instance the semantical dependence built on the notion of forgetting (Lang, Liberatore, and Marquis 2003).

WM & LTM updating: an interesting study would focus on the evolution of the knowledge with the arrival of different queries, i.e. under which conditions the formula of the query might be accepted and stored in the *WM*. Moreover, considering a capacity limited *WM* implies that some *WM* clauses might be discarded to make room for others clauses when a query is irrelevant to the current *WM*. These currently unnecessary clauses might still be relevant for later incoming queries and purely losing them might be detrimental in the long run. One perspective is hence to study in detail which clauses should be unloaded from *WM* and stored in the *LTM* and, in order to avoid too much redundancy, how those clauses could be compacted in the *LTM*. Updating the *LTM* prompts then the computation of a new association table to account for the new pieces of information, which may be done efficiently by using the old AssocCl together with QAssocV. In this context, an incremental algorithm has to be created in order to update the MUSes associated to the updated LTM.

Different inconsistency handling: Our approach han-

[7] The reader can refer to (Lagasquie-Schiex 1995) for a well-organized overview of the main classical non-monotonic inference relations and their properties (in French) or to (Cayrol and Lagasquie-Schiex 1995; Cayrol, Lagasquie-Schiex, and Schiex 1998) for its English counterparts.

dles inconsistency based on the idea that inconsistent pieces of information lead to concealing some other formulas, meaning that depending on the context some knowledge will be ignored. Introducing uncertainty on the formula, for instance with penalty logic (Dupin de Saint-Cyr, Lang, and Schiex 1994), would ensure that every piece of information is taken into consideration, albeit with different "strength".

Acknowledgment

The paper has benefited from pertinent remarks made by anonymous reviewers of ECSQARU'2021, and also from very interesting points raised by anonymous reviewers of NMR'2021. We thank them all for their very useful work.

References

Amgoud, L.; Besnard, P.; Cayrol, C.; Chatalic, P.; and Lagasquie-Schiex, M.-C. 2020. Argumentation and inconsistency-tolerant reasoning. In Marquis, P.; Papini, O.; and Prade, H., eds., *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*. Springer Nature. 415–440.

Audemard, G.; Lagniez, J.-M.; and Simon, L. 2013. Improving glucose for incremental sat solving with assumptions: Application to MUS extraction. In *International conference on theory and applications of satisfiability testing*, 309–317. Springer.

Avron, A.; Arieli, O.; and Zamansky, A. 2018. *Theory of effective propositional paraconsistent logics*. College Publications.

Baddeley, A. D., and Hitch, G. 1974. Working memory. In *Psychology of learning and motivation*, volume 8. Elsevier. 47–89.

Barrouillet, P., and Camos, V. 2007. The time-based resource-sharing model of working memory. *Working memory: State of the science* 85.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In *IJCAI*, volume 93, 640–645.

Benferhat, S.; Dubois, D.; and Prade, H. 1995. How to infer from inconsisent beliefs without revising? In *IJCAI*, volume 95, 1449–1455. Citeseer.

Bertossi, L. 2006. Consistent query answering in databases. *ACM Sigmod Record* 35(2):68–76.

Bisquert, P.; Croitoru, M.; Dupin de Saint-Cyr, F.; and Hecham, A. 2017. Formalizing cognitive acceptance of arguments: Durum wheat selection interdisciplinary study. *Minds and Machine* 27(1):233–252.

Brewka, G. 1989. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI*, volume 89, 1043–1048. Citeseer.

Calabro, C.; Impagliazzo, R.; and Paturi, R. 2006. A duality between clause width and clause density for sat. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, 7–pp. IEEE.

Carnielli, W. A., and Coniglio, M. E. 2016. *Paraconsistent logic: Consistency, contradiction and negation*, volume 40. Springer.

Cayrol, C., and Lagasquie-Schiex, M. 1995. Nonmonotonic syntax-based entailment: A classification of consequence relations. In Froidevaux, C., and Kohlas, J., eds., *Symbolic and Quantitative Approaches to Reasoning and Uncertainty, European Conference, ECSQARU'95, Fribourg, Switzerland, July 3-5, 1995, Proceedings*, volume 946 of *Lecture Notes in Computer Science*, 107–114. Springer.

Cayrol, C.; Lagasquie-Schiex, M.; and Schiex, T. 1998. Nonmonotonic reasoning: From complexity to algorithms. *Ann. Math. Artif. Intell.* 22(3-4):207–236.

Chomicki, J. 2006. Consistent query answering: Opportunities and limitations. In *17th International Workshop on Database and Expert Systems Applications (DEXA'06)*, 527–531. IEEE.

Dantsin, E., and Hirsch, E. A. 2009. Worst-case upper bounds. *Handbook of Satisfiability* 185(403-424):9.

De Neys, W., and Everaerts, D. 2008. Developmental trends in everyday conditional reasoning: The retrieval and inhibition interplay. *Journal of Experimental Child Psychology* 100(4):252–263.

De Neys, W.; Schaeken, W.; and d'Ydewalle, G. 2005. Working memory and everyday conditional reasoning: Retrieval and inhibition of stored counterexamples. *Thinking & Reasoning* 11(4):349–381.

Doder, D., and Vesic, S. 2015. How to decrease and resolve inconsistency of a knowledge base?. In *ICAART (2)*, 27–37.

Dubois, D., and Prade, H. 2014. Possibilistic logic - an overview. In Siekmann, J. H., ed., *Computational Logic*, volume 9 of *Handbook of the History of Logic*. Elsevier. 283–342.

Dubois, D.; Everaere, P.; Konieczny, S.; and Papini, O. 2020. Main issues in belief revision, belief merging and information fusion. In Marquis, P.; Papini, O.; and Prade, H., eds., *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*. Springer Nature. 441–485.

Dupin De Saint Cyr, F.; Duval, B.; and Loiseau, S. 2001. A priori revision. In Benferhat, S., and Besnard, P., eds., *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2001), Toulouse, 19/09/01-21/09/01*, number 2143 in LNAI, 488–497. http://www.springerlink.com: Springer.

Dupin de Saint-Cyr, F.; Lang, J.; and Schiex, T. 1994. Penalty logic and its link with dempster-shafer theory. In *Uncertainty Proceedings 1994*. Elsevier. 204–211.

Greco, S.; Sirangelo, C.; Trubitsyna, I.; and Zumpano, E. 2003. Preferred repairs for inconsistent databases. In

Seventh International Database Engineering and Applications Symposium, 2003. Proceedings., 202–211. IEEE.

Järvisalo, M.; Le Berre, D.; Roussel, O.; and Simon, L. 2012. The international sat solver competitions. *Ai Magazine* 33(1):89–92.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44(1-2):167–207.

Lagasquie-Schiex, M. 1995. *Contribution à l'étude des relations d'inférence non-monotone combinant inférence classique et préférences. (A Contribution to the study of non-monotonic inference relationships combining classical inference and preferences).* Ph.D. Dissertation, Paul Sabatier University, Toulouse, France.

Lang, J.; Liberatore, P.; and Marquis, P. 2003. Propositional independence: Formula-variable independence and forgetting. *J. Artif. Intell. Res.* 18:391–443.

Le Berre, D., and Parrain, A. 2010. The sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation* 7(2-3):59–64.

Li, C. M., and Manya, F. 2009. Maxsat, hard and soft constraints. In *Handbook of satisfiability*. IOS Press. 613–631.

Liffiton, M. H., and Sakallah, K. A. 2008. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning* 40(1):1–33.

McCarthy, J. 1986. Applications of circumscription to formalizing common-sense knowledge. *Artificial intelligence* 28(1):89–116.

Nebel, B. 1991. Belief revision and default reasoning: Syntax-based approaches. *KR* 91:417–428.

Pătraşcu, M., and Williams, R. 2010. On the possibility of faster sat algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, 1065–1075. SIAM.

Pearl, J. 1990. System z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, 121–135.

Rescher, N., and Manor, R. 1970. On inference from inconsistent premises. *Theory and Decision* 1:179–219.

Simon, H. A. 1955. A behavioral model of rational choice. *The quarterly journal of economics* 69(1):99–118.

Algorithms for Inconsistency Measurement using Answer Set Programming

Isabelle Kuhlmann and Matthias Thimm
University of Koblenz-Landau
Universitätsstraße 1
56072 Koblenz, Germany
{iskuhlmann, thimm}@uni-koblenz.de

Abstract

We present algorithms based on answer set programming (ASP) encodings for solving the problem of determining inconsistency degrees in propositional knowledge bases. For that, we consider the contension inconsistency measure, the forgetting-based inconsistency measure, and the hitting set inconsistency measure. Our experimental evaluation shows that all three algorithms significantly surpass the state of the art.

1 Introduction

A major challenge in symbolic approaches to AI is the handling of *inconsistent* information. The field of *Inconsistency Measurement*—see the seminal work (Grant 1978) and the book (Grant and Martinez 2018)—provides an *analytical* perspective on the issue of inconsistency in formal knowledge representation formalisms. Its aim is to quantitatively assess the *severity* of inconsistency in order to both guide automatic reasoning mechanisms and to help human modellers in identifying issues and compare different alternative formalizations. For example, inconsistency measures have been used to estimate reliability of agents in multi-agent systems (Cholvy, Perrussel, and Thevenin 2017), to analyze inconsistencies in news reports (Hunter 2006), to support collaborative software requirements specifications (Martinez, Arias, and Vilas 2004), to allow for inconsistency-tolerant reasoning in probabilistic logic (Potyka and Thimm 2017), and to monitor and maintain quality in database settings (Bertossi 2018).

Previous research on the computational complexity of inconsistency measures (Thimm and Wallner 2019) showed that evaluating them is computationally hard in general. However, as the list of application areas above shows, there is a need to practical working solutions. In this paper, we address this need by leveraging existing problem solving paradigms to develop effective algorithmic solutions to some prominent inconsistency measures. More precisely, we consider the contension inconsistency measure (Grant and Hunter 2011), the hitting set inconsistency measure (Thimm 2016), and the forgetting-based inconsistency measure (Besnard 2016) (we will give their formal definitions in Section 2). Natural decision problems pertaining to those measures are hard for the first level of the polynomial hierarchy, but still easier compared to many other measures (Thimm and Wallner 2019). We therefore believe that these measures are most likely suitable for real-world applications, due to the existence of general problem solving paradigms able to solve problems of this complexity in comparably short time. Here, we use Answer Set Programming (ASP) (Gelfond and Lifschitz 1991; Gelfond and Leone 2002; Gebser et al. 2012) for this purpose, a non-monotonic logic programming language that has been proven successful to solve problems in many other areas such as formal argumentation (Dvorák et al. 2020) and automated planning (Erdem et al. 2013), see also (Erdem, Gelfond, and Leone 2016). We selected the contension, forgetting-based, and hitting set inconsistency measure, as they are conceptually more similar to each other than to the other measures which Thimm and Wallner identified to be on the first level of the polynomial hierarchy (Thimm and Wallner 2019). Part of our ongoing research is, however, to investigate the other measures on this level, for instance the distance-based inconsistency measures proposed by Grant and Hunter (2017).

In summary, the contributions of this paper are as follows:

1. We introduce algorithms based on answer set encodings for determining the inconsistency value wrt. the contension inconsistency measure, the forgetting-based inconsistency measure, and the hitting set inconsistency measure (Section 3).

2. We present our findings of an experimental evaluation of these algorithms, where we compare their runtime with the runtime of existing baseline implementations (Section 4).

In Section 2 we give an overview on the necessary preliminaries, in particular about inconsistency measurement and answer set programming. We conclude with a discussion of our findings and possible future work in Section 5.

A short paper introducing a preliminary version of the encoding for the contension inconsistency measure has been published before (Kuhlmann and Thimm 2020). In this paper, we present an improved encoding of that measure and novel encodings for the other two measures.

2 Preliminaries

Let At be a fixed set of propositional atoms and let $\mathcal{L}(\mathsf{At})$ be the corresponding propositional language constructed with the usual connectives $\wedge$ (*conjunction*), $\vee$ (*disjunction*), and $\neg$ (*negation*). A *knowledge base* $\mathcal{K}$ is a finite set of formulas $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$. Moreover, we define $\mathbb{K}$ as the set of all knowledge bases. Further, $\mathsf{At}(X)$ denotes the set of atoms appearing in a formula (or set of formulas) X.

Interpretations give semantics to propositional languages. An interpretation i on At is a function $i : \mathsf{At} \rightarrow \{\mathit{true}, \mathit{false}\}$. We define $\mathsf{Int}(\mathsf{At})$ as the set of all interpretations wrt. At. An interpretation i *satisfies* an atom $x \in \mathsf{At}$, denoted $i \models x$, iff $i(x) = \mathit{true}$. This concept is extended to formulas in the usual manner. If an interpretation i satisfies a formula ϕ, it is called a *model* of ϕ, respectively.

Let $\Phi \subseteq \mathcal{L}(\mathsf{At})$ be a set of formulas. We define $i \models \Phi$ iff $i \models \phi$ for all $\phi \in \Phi$. A formula (or set of formulas) X_1 *entails* another formula (or set of formulas) X_2, indicated as $X_1 \models X_2$ if $i \models X_1$ implies $i \models X_2$ for every interpretation i. If there exists no interpretation i with $i \models X$, we denote this as $X \models \bot$, and X is called *inconsistent*.

2.1 Inconsistency Measurement

In general, an *inconsistency measure* $\mathcal{I}$ is a function $\mathcal{I} : \mathbb{K} \rightarrow \mathbb{R}^{\infty}_{\geq 0}$ (Thimm 2019). The intuition behind such inconsistency measures is that a higher value indicates a more severe inconsistency than a lower one. The minimal value 0 is supposed to model the absence of inconsistency, i.e., consistency.

The Contension Inconsistency Measure The *contension inconsistency measure* (Grant and Hunter 2011) is based on Priest's three-valued logic (Priest 1979). In addition to *true* and *false*, this logic introduces a third truth value denoted *both (true and false)* or *paradoxical*. In the remainder of this work we will also refer to these truth values as T, F, and B, respectively. The truth tables of this logic are presented in Table 1. A corresponding three-valued interpretation i^3 is a function that assigns one of the three truth values to each atom in a knowledge base $\mathcal{K}$:

$$i^3 : \mathsf{At}(\mathcal{K}) \mapsto \{\mathit{true}, \mathit{both}, \mathit{false}\}$$

Such an interpretation is called a *model* if each formula $\phi \in \mathcal{K}$ evaluates to either *true* or *both*. The set of all models wrt. $\mathcal{K}$ is defined as

$$\mathsf{Models}(\mathcal{K}) = \{i^3 \mid \forall \phi \in \mathcal{K}, i^3(\phi) = T \text{ or } i^3(\phi) = B\}$$

Further, we can divide the domain of an interpretation i^3 into two sets. One contains those atoms that are assigned a classical truth value (T, F), the other one contains those that are assigned truth value B, i. e., those which are involved in a conflict. The latter is defined as

$$\mathsf{Conflictbase}(i^3) = \{x \in \mathsf{At}(\mathcal{K}) \mid i^3(x) = B\}.$$

Finally, we can define the contension inconsistency measure $\mathcal{I}_c$ wrt. a knowledge base $\mathcal{K}$ as follows:

$$\mathcal{I}_c(\mathcal{K}) = \min\{|\mathsf{Conflictbase}(i^3)| \mid i^3 \in \mathsf{Models}(\mathcal{K})\}.$$

Hence, $\mathcal{I}_c$ describes the minimum number of atoms that are assigned truth value B wrt. a knowledge base $\mathcal{K}$.

x	y	$x \wedge y$	$x \vee y$
T	T	T	T
T	B	B	T
T	F	F	T
B	T	B	T
B	B	B	B
B	F	F	B
F	T	F	T
F	B	F	B
F	F	F	F

x	$\neg x$
T	F
B	B
F	T

Table 1: Truth tables for Priest's propositional three-valued logic.

Example 1 *Consider the following inconsistent knowledge base:*

$$\mathcal{K}_1 = \{a \wedge b, \neg a \wedge c, a, \neg a, \neg b\}$$

Let i_1^3 be the interpretation that assigns T to c, and B to a and b. Thus, $\mathsf{Conflictbase}(i_1^3) = \{a, b\}$. *Because each formula in $\mathcal{K}_1$ evaluates to either T or B given i_1^3, then i_1^3 is also a model of $\mathcal{K}_1$. It is easy to see that a and b must be assigned B in order to make the knowledge base satisfiable, and that no lower number of atoms being assigned B could make $\mathcal{K}_1$ satisfiable. Hence, we get $\mathcal{I}_c(\mathcal{K}_1) = 2$.*

The Forgetting-Based Inconsistency Measures The intuition behind the *forgetting-based inconsistency measure* (Besnard 2016) is to count how many atom occurrences in a knowledge base $\mathcal{K}$ have to be "forgotten" in order to recover consistency in $\mathcal{K}$, where "forgetting" is interpreted as replacing the atom occurrence with either $\top$ or $\bot$. To illustrate this, we first label each atom occurrence according to its position in $\mathcal{K}$. For instance, we can give label "1" to the first occurrence of an atom a, label "2" to the second occurrence, and so forth.

Example 2 *Recall knowledge base $\mathcal{K}_1$ given in Example 1. Assigning labels as described above yields the knowledge base*

$$\mathcal{K}_1^l = \{a^1 \wedge b^1, \neg a^2 \wedge c^1, a^3, \neg a^4, \neg b^2\}.$$

For a formula ϕ, let $\phi[x_1^{n_1} \rightarrow \psi_1, \ldots, x_p^{n_k} \rightarrow \psi_k]$ denote the formula ϕ' where the atoms $x_1, \ldots, x_p$ with labels $n_1, \ldots, n_k$ are replaced by $\psi_1, \ldots, \psi_k$.

Example 3 *Let* $\phi_1 := (a^1 \wedge b^1) \vee (\neg a^2 \wedge b^2)$.

$$\phi_1[a^2 \rightarrow \top, b^1 \rightarrow\bot] = (a\wedge \bot) \vee (\neg\top \wedge b)$$

Consequently, we can define the forgetting-based inconsistency measure as

$$\mathcal{I}_f(\mathcal{K}) = \min\{k \mid (\bigwedge \mathcal{K})[x_1^{n_1} \rightarrow \psi_1, \ldots, x_p^{n_k} \rightarrow \psi_k] \not\models \bot, \psi_1, \ldots, \psi_k \in \{\top, \bot\}\}$$

for all $\mathcal{K} \in \mathbb{K}$ with $\{x_1, \ldots, x_p\} \in \mathsf{At}(\mathcal{K})$ and $n_1, \ldots, n_k$ being the corresponding labels.

Example 4 *With regard to the labeled knowledge base $\mathcal{K}_1^l$, given in Example 2, we could replace a^1, a^3, and b^1 by $\top$,* *i.e., we could forget a^1, a^3, and b^1, in order to restore consistency. Although there are other options to recover consistency, e. g., by forgetting a^2, a^4, and b^2, one can clearly see that it is not possible to obtain consistency by forgetting fewer than 3 atom occurrences. Hence, $\mathcal{I}_f(\mathcal{K}_1) = 3$.*

The Hitting Set Inconsistency Measure A subset $H \subseteq \mathsf{Int}(\mathsf{At})$ is a *hitting set* of a knowledge base $\mathcal{K}$ if for every formula $\phi \in \mathcal{K}$ there is an interpretation $\omega \in H$ with $\omega \models \phi$. Thus, there only exists a hitting set for $\mathcal{K}$ iff there is no $\phi \in \mathcal{K}$ with $\phi \models \perp$, i.e., no formula $\phi \in \mathcal{K}$ is contradictory. Moreover, if there exists a hitting set H wrt. $\mathcal{K}$, and $|H| = 1$, the only element in H is a model of $\mathcal{K}$. Thus, in this case, $\mathcal{K}$ is consistent. Based on the preceding definitions, the hitting set inconsistency measure $\mathcal{I}_h(\mathcal{K})$ (Thimm 2016) is defined as the minimum number of elements in the hitting set, subtracted by 1. If there exists no hitting set wrt. $\mathcal{K}$, then $\mathcal{I}_h(\mathcal{K}) = \infty$. Formally,

$$\mathcal{I}_h(\mathcal{K}) = \min\{|H| \mid H \text{ is a hitting set of } \mathcal{K}\} - 1,$$

with $\min \emptyset = \infty$ for all $\mathcal{K} \in \mathbb{K}\backslash\{\emptyset\}$. Further, $\mathcal{I}_h(\emptyset) = 0$.

Example 5 *Consider again $\mathcal{K}_1$ as defined in Example 1. As none of the formulas in $\mathcal{K}_1$ is contradictory, there must exist a hitting set. Also, as the knowledge base is obviously inconsistent, we need at least two interpretations to compile a hitting set. Let interpretation i_1 assign T to the formulas a, b, and c, and let interpretation i_2 assign F to a and b, and T to c. Each formula $\phi \in \mathcal{K}_1$ is satisfied by one of these two interpretations. Hence, $\mathcal{I}_h(\mathcal{K}_1) = 2 - 1 = 1$.*

2.2 Answer Set Programming

Answer set programming (ASP) (Gebser et al. 2012; Lifschitz 2008; Brewka, Eiter, and Truszczynski 2011) is a declarative problem solving approach targeted at difficult search problems. ASP incorporates ideas of logic programming and Reiter's default logic (Reiter 1980). A problem is modeled as an *extended logic program* which consists of a set of *rules*. An ASP rule is of the form

$$r = H \leftarrow A_1, \ldots, A_n, \mathtt{not}\, B_1, \ldots, \mathtt{not}\, B_m. \qquad (1)$$

where H, A_j with $j \in \{1, \ldots, n\}$, and B_k with $k \in \{1, \ldots, m\}$ are classical literals. ASP rules consist of a *head* and a *body*, both of which can be empty. We denote the sets of literals contained in the head and body of a rule r as $\mathsf{head}(r)$, and $\mathsf{body}(r)$, respectively. A rule with an empty body is called a *fact*, a rule with an empty head is referred to as a *constraint*. In (1), $\mathsf{head}(r) = \{H\}$, and $\mathsf{body}(r) = \{A_1, \ldots, A_n, B_1, \ldots, B_m\}$. An extended logic program is *positive* if it does not contain any instance of $\mathtt{not}$. Moreover, a set of literals L is called *closed* under a positive program P if and only if for any rule $r \in P$, $\mathsf{head}(r) \in L$ whenever $\mathsf{body}(r) \subseteq L$. The set L is consistent if it does not contain both A and $\neg A$ for some literal A. The smallest of such sets wrt. a positive program P, which is always uniquely defined, is referred to as $\mathsf{Cn}(P)$. With regard to an arbitrary program P, a set L is an *answer set* of P if $L = \mathsf{Cn}(P^L)$ and L is consistent, with

$$\begin{aligned} P^L = \{H \leftarrow A_1, \ldots, A_n \mid \\ & H \leftarrow A_1, \ldots, A_n, \mathtt{not}\, B_1, \ldots, \mathtt{not}\, B_m. \in P, \\ & \{B_1, \ldots, B_m\} \cap L = \emptyset\} \end{aligned}$$

The head of an ASP rule is not necessarily comprised of only one literal. Some ASP dialects allow for more complex structures, such as *cardinality constraints*, which can be used as both body elements and heads. A cardinality constraint with lower bound l and upper bound u is defined as

$$l\{A_1, \ldots, A_n, \mathtt{not}\, B_1, \ldots, \mathtt{not}\, B_m\}u.$$

This can be interpreted as follows: if at least l and at most u of the literals $A_1, \ldots, A_n, B_1, \ldots, B_m$ are included in an answer set, a cardinality rule is satisfied by this answer set.

ASP additionally offers the option to express cost functions involving minimization and/or maximization in order to solve optimization problems (Gebser et al. 2012). Here, we only need optimisation statements of the form

$$minimize\{\ell_1, \ldots, \ell_n\}$$

which instruct the ASP solver to include only a minimal number of the literals $\ell_1, \ldots, \ell_n$ in any answer set.

3 Measuring Inconsistency Using ASP

The proposed algorithms involve the development of ASP encodings for each one of the three previously described inconsistency measures. Although the specifics of each inconsistency measure have to be considered individually, there are some aspects that all ASP-based algorithms we propose have in common. To begin with, each atom is supposed to be assigned a unique truth value.

Example 6 *In classical propositional logic, we could model that an atom x is supposed to be assigned either T or F by introducing two corresponding ASP atoms e_{x_T} and e_{x_F}. An atom is true, if it is not false, and vice versa. The respective ASP rules can be defined as follows:*

$$e_{x_T} \leftarrow \mathtt{not}\, e_{x_F}.$$
$$e_{x_F} \leftarrow \mathtt{not}\, e_{x_T}.$$

In the remainder of this paper, we refer to this part as *unique atom evaluation*. Note that ASP atoms e_{ϕ_T}, e_{ϕ_F} representing the evaluation of a formula ϕ can be created in the same manner as shown above wrt. propositional atoms.

Moreover, within the encoding, each formula ϕ must be satisfied, i.e., no formula should evaluate to F. This is achieved through *integrity constraints* of the form

$$\leftarrow e_{\phi_F}.$$

for every $\phi \in \mathcal{K}$. Another element that is common in all three encodings is that the relations between elements within a formula have to be encoded. More precisely, encodings for the connectors $\wedge$, $\vee$, and $\neg$ have to be created.

Example 7 *The evaluation of a conjunction of two propositional formulas ϕ, ψ can be modeled in ASP by encoding that it is only true if both ϕ and ψ are true, and false otherwise:*

$$e_{(\phi\wedge\psi)_T} \leftarrow e_{\phi_T}, e_{\psi_T}.$$
$$e_{(\phi\wedge\psi)_F} \leftarrow \mathtt{not}\, e_{(\phi\wedge\psi)_T}.$$

In the following, we will refer to this part as *connector encodings*.

Since the possible inconsistency values regarding all three considered measures are natural numbers from a well-defined interval (Thimm and Wallner 2019), we can make use of a minimization statement to compute the desired inconsistency value.

Example 8 *Let our aim be to minimize the number of atoms $x \in \mathsf{At}(\mathcal{K})$ that are assigned truth value B in three-valued logic, wrt. a knowledge base $\mathcal{K}$. Let the ASP atom e_{x_B} encode the assignment of B to atom x. The corresponding minimize statement can be expressed as*

$$minimize\{e_{x^1_B}, \dots, e_{x^n_B}\}.$$

with $\{x^1, \dots, x^n\} = \mathsf{At}(\mathcal{K})$

The subsequent sections describe the specific ASP encodings for $\mathcal{I}_c$, $\mathcal{I}_h$, and $\mathcal{I}_f$.

3.1 The Contension Inconsistency Measure

With regard to the contension inconsistency measure $\mathcal{I}_c$, we can construct an extended logic program $P_c(\mathcal{K})$ to compute $\mathcal{I}_c(\mathcal{K})$ wrt. a knowledge base $\mathcal{K}$ as follows.

1. We first include rules that guess a three-valued interpretation. For that, we need to ensure unique atom evaluation for each $x \in \mathsf{At}(\mathcal{K})$ wrt. Priest's three-valued logic. Thus, an atom is *true* if it is neither *both* nor *false*. The other two cases follow analogously:

$$e_{x_T} \leftarrow \mathtt{not}\, e_{x_B}, \mathtt{not}\, e_{x_F}.$$
$$e_{x_B} \leftarrow \mathtt{not}\, e_{x_T}, \mathtt{not}\, e_{x_F}.$$
$$e_{x_F} \leftarrow \mathtt{not}\, e_{x_B}, \mathtt{not}\, e_{x_T}.$$

2. The connector encodings for each (sub)formula in $\mathcal{K}$ follow from the truth tables given in Table 1. For instance, a conjunction is only *true* if both conjuncts are *true*. It is *false*, if at least one of its conjuncts is *false*, and it is *both* if it is neither *true* nor *false*. The rules for disjunction and negation are created in the same fashion.

$$\begin{aligned} \phi \wedge \psi \mapsto \quad & e_{(\phi\wedge\psi)_T} \leftarrow e_{\phi_T}, e_{\psi_T}. \\ & e_{(\phi\wedge\psi)_F} \leftarrow e_{\phi_F}. \\ & e_{(\phi\wedge\psi)_F} \leftarrow e_{\psi_F}. \\ & e_{(\phi\wedge\psi)_B} \leftarrow \mathtt{not}\, e_{(\phi\wedge\psi)_F},\ \mathtt{not}\, e_{(\phi\wedge\psi)_T}. \end{aligned}$$

$$\begin{aligned} \phi \vee \psi \mapsto \quad & e_{(\phi\vee\psi)_F} \leftarrow e_{\phi_F}, e_{\psi_F}. \\ & e_{(\phi\vee\psi)_T} \leftarrow e_{\phi_T}. \\ & e_{(\phi\vee\psi)_T} \leftarrow e_{\psi_T}. \\ & e_{(\phi\vee\psi)_B} \leftarrow \mathtt{not}\, e_{(\phi\vee\psi)_F},\ \mathtt{not}\, e_{(\phi\vee\psi)_T}. \end{aligned}$$

$$\begin{aligned} \neg\phi \mapsto \quad & e_{(\neg\phi)_B} \leftarrow e_{\phi_B}. \\ & e_{(\neg\phi)_T} \leftarrow e_{\phi_F}. \\ & e_{(\neg\phi)_F} \leftarrow e_{\phi_T}. \end{aligned}$$

3. Every formula $\phi \in \mathcal{K}$ must be evaluated to *true* or *both* in three-valued logic, i.e., it must not be evaluated to *false*. We therefore add an integrity constraint for each formula:

$$\leftarrow e_{\phi_F}.$$

4. Finally, we want to minimize the number of atoms in $\mathcal{K}$ that are assigned the truth value B. Hence, we add the following minimize statement:

$$minimize\{e_{x^1_B}, \dots, e_{x^n_B}\}.$$

Now $P_c(\mathcal{K})$ is the union of all rules defined in 1–4. Further, let i^3_M be the three-valued interpretation represented by an answer set M of $P_c(\mathcal{K})$.

Theorem 1 *Let M be an optimal answer set of $P_c(\mathcal{K})$. Then $|i^3_M(B)^{-1}| = \mathcal{I}_c(\mathcal{K})$*[1].

The proof of the above theorem as well as further technical results are omitted due to space restrictions, but can be found in the appendix[2].

3.2 The Forgetting-Based Inconsistency Measure

The forgetting-based inconsistency measure $\mathcal{I}_f(\mathcal{K})$ is determined by the number of atom occurrences that need to be "forgotten" in order to make the knowledge base $\mathcal{K}$ consistent. An extended logic program $P_f(\mathcal{K})$ which computes $\mathcal{I}_f(\mathcal{K})$ wrt. a knowledge base $\mathcal{K}$ can be constructed as described below.

1. We first include rules that guess a model for the knowledge base after forgetting operations took place, in order to ensure that the knowledge base is consistent. Although individual atom *occurrences* may be replaced by $\top$ or $\bot$, an atom must be either *true* or *false* in that interpretation. Thus, for every $x \in \mathsf{At}(\mathcal{K})$:

$$e_{x_T} \leftarrow \mathtt{not}\, e_{x_F}.$$
$$e_{x_F} \leftarrow \mathtt{not}\, e_{x_T}.$$

2. We need to ensure that each atom occurrence is evaluated uniquely. This means that an atom occurrence x^n can either be *true*, *false*, or forgotten, i.e., replaced by either $\top$ or $\bot$. If an atom occurrence x^n is supposed to be replaced by $\top$ or $\bot$, we represent this using the ASP atoms $e_{x^n_{\mathrm{forget}_\top}}$ or $e_{x^n_{\mathrm{forget}_\bot}}$, respectively:

$$e_{x^n_{\mathrm{forget}_\top}} \leftarrow \mathtt{not}\, e_{x^n_T}, \mathtt{not}\, e_{x^n_F}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\bot}}.$$
$$e_{x^n_{\mathrm{forget}_\bot}} \leftarrow \mathtt{not}\, e_{x^n_T}, \mathtt{not}\, e_{x^n_F}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\top}}.$$

We also need to ensure that an individual atom occurrence is only set to *true* or *false* if the atom as a whole is evaluated to *true* or *false*, respectively.

$$e_{x^n_T} \leftarrow e_{x_T}, \mathtt{not}\, e_{x^n_F}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\top}}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\bot}}.$$
$$e_{x^n_F} \leftarrow e_{x_F}, \mathtt{not}\, e_{x^n_T}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\top}}, \mathtt{not}\, e_{x^n_{\mathrm{forget}_\bot}}.$$

3. The connector encodings for all (sub)formulas ϕ, ψ in $\mathcal{K}$

[1] For any function $f : X \mapsto Y$ and $y \in Y$ we define $f^{-1}(y) = \{x \in X \mid f(x) = y\}$

[2] http://mthimm.de/misc/nmr21_ikmt.pdf

simply model propositional entailment:

$$
\begin{aligned}
\phi \land \psi \mapsto \qquad & e_{(\phi\land\psi)_T} \leftarrow e_{\phi_T}, e_{\psi_T}. \\
& e_{(\phi\land\psi)_F} \leftarrow \texttt{not}\, e_{(\phi\land\psi)_T}. \\[1ex]
\phi \lor \psi \mapsto \qquad & e_{(\phi\lor\psi)_F} \leftarrow e_{\phi_F}, e_{\psi_F}. \\
& e_{(\phi\lor\psi)_T} \leftarrow \texttt{not}\, e_{(\phi\lor\psi)_F}. \\[1ex]
\neg\phi \mapsto \qquad & e_{(\neg\phi)_T} \leftarrow e_{\phi_F}. \\
& e_{(\neg\phi)_F} \leftarrow \texttt{not}\, e_{(\neg\phi)_T}.
\end{aligned}
$$

4. If a (sub)formula ϕ is actually an atom occurrence x^n, it can either be set to *true* or *false*, or forgotten:

$$
\begin{aligned}
& e_{\phi_T} \leftarrow e_{x^n_T}. \\
& e_{\phi_T} \leftarrow e_{x^n_{\text{forget}_\top}}. \\
& e_{\phi_F} \leftarrow e_{x^n_F}. \\
& e_{\phi_F} \leftarrow e_{x^n_{\text{forget}_\bot}}.
\end{aligned}
$$

5. All formulas $\phi \in \mathcal{K}$ must evaluate to *true* after the forgetting operation is applied. Hence, we add the following integrity constraint for each $\phi \in \mathcal{K}$:

$$\leftarrow e_{\phi_F}.$$

6. Lastly, we minimize the number of atom occurrences which are forgotten:

$$
\begin{aligned}
\mathit{minimize}\{ & e_{x^n_{\text{forget}_\top}}, e_{x^n_{\text{forget}_\bot}}, \ldots, \\
& e_{y^n_{\text{forget}_\top}}, e_{y^n_{\text{forget}_\bot}}, \ldots \}.
\end{aligned}
$$

Note that the rules described in 2. ensure that no atom occurrence is simultaneously replaced by $\top$ and $\bot$.

The union of all rules defined above (in 1–6) constitute the extended logic program $P_f(\mathcal{K})$. We denote the set of atom occurrences that are replaced by $\top$ wrt. a knowledge base $\mathcal{K}$ as T_M, and the set of atom occurrences that are replaced by $\bot$ as F_M. With M being an answer set of $P_f(\mathcal{K})$ we define

$$
\begin{aligned}
T_M &= \{x^n \mid x \in \mathsf{At}(\mathcal{K}), e_{x^n_{\text{forget}_\top}} \in M\}, \\
F_M &= \{x^n \mid x \in \mathsf{At}(\mathcal{K}), e_{x^n_{\text{forget}_\bot}} \in M\}.
\end{aligned}
$$

Theorem 2 *Let M be an optimal answer set of $P_f(\mathcal{K})$. Then $|T_M| + |F_M| = \mathcal{I}_f(\mathcal{K})$.*

3.3 The Hitting Set Inconsistency Measure

The hitting set inconsistency measure $\mathcal{I}_h(\mathcal{K})$ is defined by the size of the minimal hitting set wrt. a knowledge base $\mathcal{K}$, subtracted by 1. The maximal size of such a hitting set is determined by the number of formulas in $\mathcal{K}$. In the following, we denote the number of formulas in $\mathcal{K}$ as N. Further, i_n refers to the n-th interpretation out of the N possible interpretations we need to consider, assuming that the interpretations have an arbitrary, but fixed order. An interpretation i_n is represented as ω_n in our ASP encoding. Note that the notation ω_n does not only appear as an ASP atom on its own, but also serves the purpose of a label linking the representations of formulas and atoms to specific interpretations. For example, the ASP atom e_{ϕ_T,ω_n} represents the formula ϕ being evaluated to T under the interpretation i_n. We construct an extended logic program $P_h(\mathcal{K})$ which computes $\mathcal{I}_h(\mathcal{K})$ as follows.

1. We first include rules that guess the N interpretations, some of those may be used in the final hitting set. We need to ensure unique atom evaluation wrt. each atom $x \in \mathsf{At}(\mathcal{K})$, as we did with the previous two encodings. However, this time we need to take each possible interpretation into account as well, because an atom may be assigned the truth value T wrt. one interpretation in the hitting set, but F wrt. another one. Thus, for each atom $x \in \mathsf{At}(\mathcal{K})$ and each interpretation i_n, $n \in \{1, \ldots, N\}$, we define:

$$
\begin{aligned}
& e_{x_T,\omega_n} \leftarrow \texttt{not}\, e_{x_F,\omega_n}. \\
& e_{x_F,\omega_n} \leftarrow \texttt{not}\, e_{x_T,\omega_n}.
\end{aligned}
$$

2. We ensure that at least one ASP atom ω_n representing an interpretation is contained in the answer set by constructing the following cardinality constraint:

$$1\{\omega_1, \ldots, \omega_N\}N.$$

3. The connector encodings for each (sub)formula in $\mathcal{K}$ follow classical propositional entailment. Again, each rule has to be created with regard to each possible interpretation:

$$
\begin{aligned}
\phi \land \psi \mapsto \qquad & e_{(\phi\land\psi)_T,\omega_n} \leftarrow e_{\phi_T,\omega_n}, e_{\psi_T,\omega_n}. \\
& e_{(\phi\land\psi)_F,\omega_n} \leftarrow \texttt{not}\, e_{(\phi\land\psi)_T,\omega_n}. \\[1ex]
\phi \lor \psi \mapsto \qquad & e_{(\phi\lor\psi)_F,\omega_n} \leftarrow e_{\phi_F,\omega_n}, e_{\psi_F,\omega_n}. \\
& e_{(\phi\lor\psi)_T,\omega_n} \leftarrow \texttt{not}\, e_{(\phi\lor\psi)_F,\omega_n}. \\[1ex]
\neg\phi \mapsto \qquad & e_{(\neg\phi)_T,\omega_n} \leftarrow e_{\phi_F,\omega_n}. \\
& e_{(\neg\phi)_F,\omega_n} \leftarrow \texttt{not}\, e_{(\neg\phi)_T,\omega_n}.
\end{aligned}
$$

4. In order to meet the definition of a hitting set, we need to ensure that each formula $\phi \in \mathcal{K}$ is satisfied wrt. at least one interpretation:

$$
\begin{aligned}
& e_{\phi_T} \leftarrow e_{\phi_T,\omega_n}, \omega_n. \\
& e_{\phi_F} \leftarrow \texttt{not}\, e_{\phi_T}.
\end{aligned}
$$

5. Again, we add an integrity constraint for each formula $\phi \in \mathcal{K}$:

$$\leftarrow e_{\phi_F}.$$

6. We minimize the number of interpretations that are required to satisfy each formula in the given knowledge base using the following minimize statement:

$$\mathit{minimize}\{\omega_1, \ldots, \omega_N\}.$$

As opposed to the minimize statements of the other two encodings, the minimal value is not 0, but 1. This is because we minimize the number of interpretations required

Data-set	Signature size	Formulas per knowl. base	Atoms per formula (mean)	Atoms per formula (max)	Timeouts $\mathcal{I}_c$ naive	Timeouts $\mathcal{I}_c$ ASP	Timeouts $\mathcal{I}_h$ naive	Timeouts $\mathcal{I}_h$ ASP	Timeouts $\mathcal{I}_f$ naive	Timeouts $\mathcal{I}_f$ ASP
A	3	5–15	2.22	6	0	0	0	0	116	0
A	5	15–25	3.10	11	0	0	107	0	200	0
A	10	15–25	3.14	10	0	0	74	0	200	0
A	15	15–25	3.11	11	25	0	108	0	200	0
A	15	25–50	3.11	11	195	0	179	0	200	0
A	20	25–50	3.12	11	199	0	198	0	200	0
B	25	25–50	3.08	13	198	0	199	0	200	0
B	25	50–100	3.11	11	200	0	200	24	200	116
B	30	50–100	3.10	13	200	0	200	9	200	140

Table 2: Overview of the sets of knowledge bases making up dataset A and dataset B.

to make a knowledge base consistent. Hence, if we only need one interpretation, the respective knowledge base is consistent. Consequently, we need to subtract 1 from the computed minimum in order to get the correct value of $\mathcal{I}_h(\mathcal{K})$.

It should be noted that there are knowledge bases which contain one or more contradictory formulas, such as $a \wedge \neg a$. In such a case, there exists no interpretation (in classical propositional logic) which could satisfy the respective formula. If a formula $\phi \in \mathcal{K}$ is contradictory, we cannot include e_{ϕ_T} in any answer set of $P_h(\mathcal{K})$. We therefore needed to include e_{ϕ_F} in the answer set—which is not allowed due to the integrity constraint. Thus, no answer set of $P_h(\mathcal{K})$ exists, and $\mathcal{I}_h(\mathcal{K})$ has the value ∞.

We define $P_h(\mathcal{K})$ to be the extended logic program specified by the union of all rules defined in 1–6. For each $\omega_n \in M$, with M being an answer set of $P_h(\mathcal{K})$, we define i_{M,ω_n} via

$$i_{M,\omega_n}(x) = \begin{cases} \top & e_{a_T,\omega_n} \in M \\ \bot & e_{a_F,\omega_n} \in M \end{cases}$$

for all $x \in \mathsf{At}(\mathcal{K})$. Further, we define

$$\Omega(M) = \{i_{M,\omega_n} \mid \omega_n \in M\},$$

which corresponds to the minimal hitting set of $\mathcal{K}$.

Theorem 3 *Let M be an optimal answer set of $P_h(\mathcal{K})$. Then $|\Omega(M)| - 1 = \mathcal{I}_h(\mathcal{K})$. If no answer set of $P_h(\mathcal{K})$ exists, $\mathcal{I}_h(\mathcal{K}) = \infty$.*

4 Experiments

The goal of our experimental evaluation is to compare the empirical runtime of our ASP-based implementations with existing baseline implementations of the individual measures.

The three introduced ASP encodings for inconsistency measurement were constructed by means of the Java libraries provided by *TweetyProject*[3]. The actual calculation of the answer sets is performed by the Clingo solver, version 5.4.0 (Gebser et al. 2016). TweetyProject also includes naive (brute-force) implementations of all three inconsistency measures. More precisely, $\mathcal{I}_c$ is implemented by iterating through all subsets of atoms (with increasing cardinality), forgetting all occurrences of the atoms of the current set in the knowledge base (thus effectively setting their three-valued truth value to B), and then checking whether the resulting knowledge base is consistent by means of a SAT solver (here, Sat4j v2.3.5[4]). Once a consistent knowledge base is found, the cardinality of the current set of atoms is returned. The measure $\mathcal{I}_f$ is implemented by iterating through all possible forgetting operations (with increasing number) and checking whether the resulting knowledge base is consistent (again using Sat4j v2.3.5). The measure $\mathcal{I}_h$ is implemented by considering every set of interpretations (with increasing cardinality) and checking whether each formula of the knowledge base is satisfied by at least one interpretation. To the best of our knowledge, no further implementations of the three inconsistency measures exist.

All experiments were run on a computer with 16 GB RAM and a quad core Intel Core i7-8550U CPU which has a maximum clock speed of 4000 MHz.

4.1 Datasets

For evaluation, we consider both some existing benchmarks as well as a set of newly compiled knowledge bases which were created using *TweetyProject*. It should be noted that, to the best of our knowledge, in the field of inconsistency measurement no dedicated dataset exists that could be utilized to evaluate different implementations against each other. Hence, we compile our own dataset. More specifically, we generated four different sets of knowledge bases (datasets A–D)[5] tailored for different purposes as elaborated in the following.

To get a fundamental overview of the behavior of our implementations, we compiled dataset A, which consists of overall rather small random knowledge bases of varying complexity. To be precise, the dataset is comprised of six subsets, each containing 200 knowledge bases. The simplest subset contains between 5 and 15 formulas per knowledge

[3] http://tweetyproject.org/

[4] https://www.sat4j.org

[5] All datasets will be made publicly available.

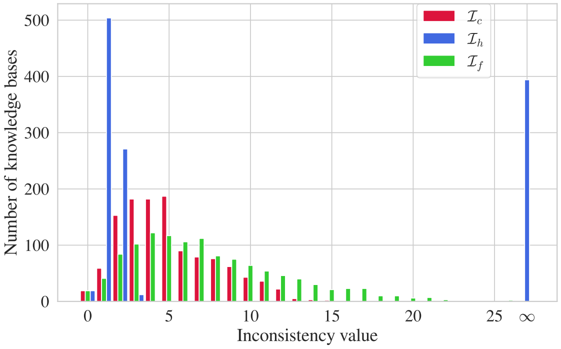


Figure 1: Overview of the inconsistency values of the knowledge bases in dataset A.

base wrt. a signature size of 3, the most complex one between 25 and 50 wrt. a signature size of 20. More details can be obtained from the upper left part of Table 2. To generate these knowledge bases, the *SyntacticRandomSampler*[6] provided by *TweetyProject* was utilized. A few knowledge bases in dataset A are consistent, and some knowledge bases contain contradictory formulas (i. e., wrt. the latter knowledge bases, $\mathcal{I}_h = \infty$). More details regarding the inconsistency values of the knowledge bases of dataset A are provided in Figure 1.

As dataset A already reveals the limits of some of the implementations, dataset B is designed to be a bit more challenging for the remaining ones. Again, the *SyntacticRandomSampler* was used for the generation process, and again, the dataset consists of subsets of 200 formulas each. Overall, the dataset is comprised of 600 knowledge bases which contain between 25 and 150 formulas with signature sizes of 25 or 30. More details are given in the lower left part of Table 2. Besides, an overview of the inconsistency values of dataset B is provided in Figure 2.

In addition to the sampled knowledge bases, we considered benchmark data from different SAT competitions in dataset C. Because the subject of this work is to measure inconsistency, only inconsistent instances were considered. In total, we gathered 105 instances from four different sources:

1. 5 instances referring to the Pigeon Hole problem[7]. They consist of between 42 and 110 variables as well as between 133 and 561 clauses.

2. 8 knowledge bases encoding the two-coloring of a graph consisting of 60 to 160 variables and 160 to 400 clauses.

3. 8 knowledge bases from circuit fault analysis which comprise between 435 and 10,410 variables, and between 1027 and 34,238 clauses.

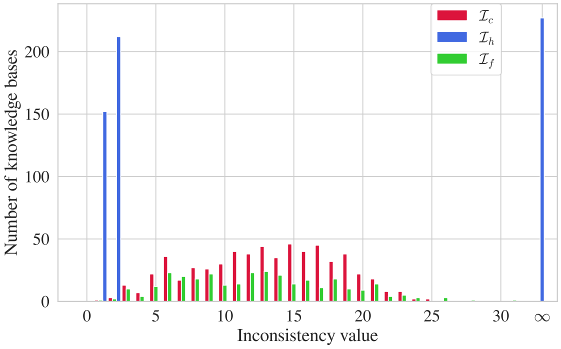


Figure 2: Overview of the inconsistency values of the knowledge bases in dataset B. Note that the set of values regarding $\mathcal{I}_f$ is incomplete due to both implementations timing out wrt. 256 out of the 600 instances.

4. 84 DaimlerChrysler benchmarks[8] with between 1608 and 2038 variables and between 4496 and 11,352 clauses.

The inconsistency values of the knowledge bases in dataset C are 1 in all cases wrt. $\mathcal{I}_c$ and $\mathcal{I}_f$. Regarding $\mathcal{I}_h$, the inconsistency values are either 1 or ∞. Note that we refer only to those knowledge bases which did not cause a timeout for both implementations of $\mathcal{I}_f$ and $\mathcal{I}_h$.

Dataset D consists of knowledge bases extracted from benchmark data of the International Competition on Computational Models of Argumentation 2019 (ICCMA'19)[9]. An abstract argumentation framework (Dung 1995) is a directed graph $F = (A, R)$ where A is a set of arguments and R models a conflict relation between arguments. A computational task here is to find a *stable extension*, i. e., a set $E \subseteq A$ with $(a, b) \notin R$ for all $a, b \in E$ and $(a, c) \in R$ for all $c \in A \setminus E$ and some $a \in E$. For each instance from ICCMA'19, we encode the instance and the problem of finding such a stable extension via the approach from (Besnard, Doutre, and Herzig 2014) and, additionally, add constraints to ensure that 20% of randomly selected arguments have to be contained in E. Note that the latter constraints usually make the knowledge base inconsistent.

4.2 Results

To begin with, we measure the runtime of both the naive (brute-force) and the ASP-based versions of all three inconsistency measures $\mathcal{I}_c$, $\mathcal{I}_h$, and $\mathcal{I}_f$ on dataset A. A timeout is set to 120 seconds. The results clearly demonstrate the limitations of all three brute-force algorithms. Figure 3 shows a cactus plot which illustrates a direct comparison between the naive versions of all measures and their respective ASP-based counterparts. The measured execution times were sorted from low to high wrt. each algorithm. None of the ASP-based algorithms produced a timeout, while all

[6] http://tweetyproject.org/api/1.17/net/sf/tweety/logics/pl/util/SyntacticRandomSampler.html

[7] Those instances referring to the two-coloring of graphs, to circuit fault theory, and to the Pigeon Hole problem are available at https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html

[8] Available at https://web.archive.org/web/20080820084020/http://www-sr.informatik.uni-tuebingen.de/~sinz/DC/

[9] http://argumentationcompetition.org/2019/

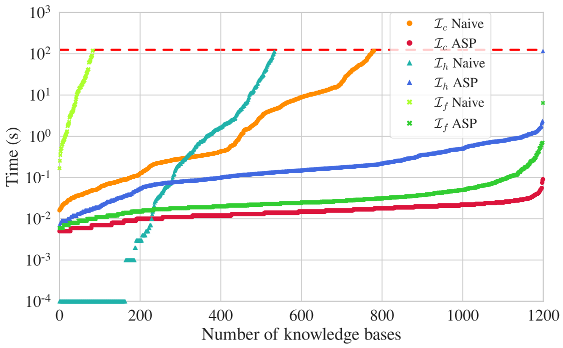


Figure 3: Runtimes of $\mathcal{I}_c$, $\mathcal{I}_h$, and $\mathcal{I}_f$ regarding both the naive and the ASP-based algorithms wrt. dataset A. The red dashed line indicates the timeout of 120 seconds.

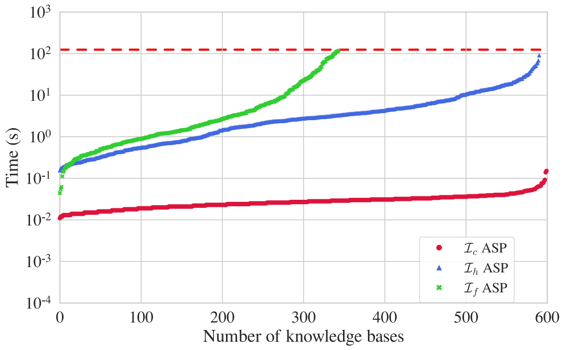


Figure 4: Runtimes of the ASP-based algorithms wrt. dataset B. Because the naive algorithms produced timeouts for all instances (except $\mathcal{I}_c$ in two cases and $\mathcal{I}_h$ in one case), they are not visualized in the plot. The red dashed line indicates the timeout of 120 seconds.

three naive versions did so in several hundred cases. In particular, the naive algorithm for $\mathcal{I}_f$ performs very poorly. The right part of Table 2 reveals that it could only handle some instances of the simplest subset of dataset A at all—for all other instances, it produced a timeout. Another noteworthy point is that both implementations for $\mathcal{I}_c$ performed comparatively well. The reason for this presumably lies in the nature of the inconsistency measure itself. For example, the number of possible values is, in most cases, smaller than that of $\mathcal{I}_h$ or $\mathcal{I}_f$.

Next, we applied all algorithms on dataset B. However, it turned out that all three naive algorithms produce timeouts in almost all instances. The only exceptions are two instances that could be solved by the naive variant of $\mathcal{I}_c$ and one instance that could be solved by the naive variant of $\mathcal{I}_h$. Hence, when considering dataset B, we focus on the ASP-based algorithms. Although the knowledge bases in dataset B are not much more complex than those of dataset A (see Table 2 for details), the ASP-based algorithms for $\mathcal{I}_h$ and $\mathcal{I}_f$ exhibit some difficulties, as Figure 4 visualizes. More precisely, the ASP-based implementation of $\mathcal{I}_h$ produces a timeout in 33 out of 600 cases, and the implementation for $\mathcal{I}_f$ even in 256 cases. More details regarding the number of timeouts for each implementation wrt. each subset of dataset B are provided in the lower right section of Table 2. Nevertheless, it should be noted that the ASP-based implementation of the contension inconsistency measure behaved differently: not only did it not produce any timeouts, it took only < 1 second for each knowledge base in dataset B.

We also applied all algorithms on dataset C. Because of the large size of some of the knowledge bases (up to 2038 variables and 11,352 clauses), we increased the timeout from 2 to 5 minutes. As Figure 5 shows, both implementations of $\mathcal{I}_c$ as well as both implementations of $\mathcal{I}_f$ were able to compute inconsistency values for most knowledge bases. However, regarding both measures, the ASP version produced fewer timeouts than its respective naive counterpart. The naive implementation of $\mathcal{I}_h$ could not solve a single instance of dataset C. The corresponding ASP-based implementation could at least compute the inconsistency values of 11 knowledge bases.

The overall rather poor performance of both implementations of $\mathcal{I}_h$ compared to the implementations of the other two measures is presumably due to the nature of dataset C: all knowledge bases are given in the DIMACS[10] file format. This means that all knowledge bases are in conjunctive normal form and each clause is considered an individual formula. Moreover, the inconsistency value is always 1, except for some instances regarding $\mathcal{I}_h$, where the inconsistency value is ∞. Further, most knowledge bases contain a large number of clauses. The size of the ASP encodings regarding $\mathcal{I}_c$ and $\mathcal{I}_f$ largely depends on the number of atoms, or atom occurrences, respectively, as well as the size and complexity of the individual formulas. The size of the ASP encoding of $\mathcal{I}_h$, on the other hand, highly depends on the number of possible interpretations, i. e., the number of formulas, because every formula, subformula and atom needs to be encoded wrt. each of these interpretations. Consequently, with a large number of formulas in a knowledge base, we also get an answer set program containing a vast number of rules. This makes $\mathcal{I}_h$ slower and less practically applicable as the other two measures in a dataset which possesses properties like datset C.

Another aspect that strikes out with regard to dataset C is that the naive implementations of $\mathcal{I}_c$ and $\mathcal{I}_f$ perform relatively well in comparison to dataset A and B. The reason for this lies most probably in the inconsistency values of the knowledge bases in dataset C, which is always 1 for $\mathcal{I}_c$ and $\mathcal{I}_f$. The inconsistency values of most instances in both dataset A and B are significantly higher (see Figures 1 and 2). Since the brute-force implementations of $\mathcal{I}_c$ and $\mathcal{I}_f$ check the lowest possible values first, they can compute lower inconsistency values faster than higher ones, given the size of the respective knowledge bases is the same.

Finally, we run all implementations on dataset D. As with

[10] http://www.satcompetition.org/2011/format-benchmarks2011.html

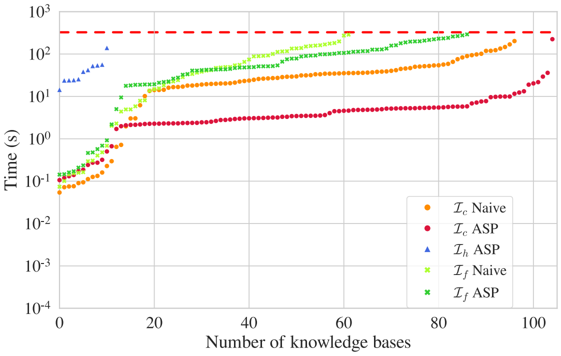


Figure 5: Runtimes of $\mathcal{I}_c$, $\mathcal{I}_f$, and $\mathcal{I}_h$ regarding dataset C. The naive implementation of $\mathcal{I}_h$ produced a timeout for all instances, so it is not shown in the plot. The red dashed line indicates the timeout of 300 seconds.

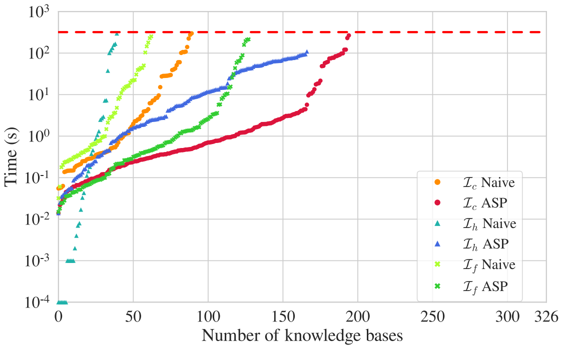


Figure 6: Runtimes of $\mathcal{I}_c$, $\mathcal{I}_h$, and $\mathcal{I}_f$ regarding both the naive and the ASP-based algorithms wrt. dataset D. The red dashed line indicates the timeout of 300 seconds.

dataset C, we set a timeout to 5 minutes. As Figure 6 visualizes, all six implementations could solve a non-empty subset of the knowledge bases in the dataset. Nonetheless, it is noticeable that none of the implementations could compute inconsistency values for the entire dataset. All implementations produced a timeout for at least 100 instances. Again, each ASP-based implementation had fewer timeouts than its respective naive counterpart.

5 Conclusion

In the course of this paper, we presented algorithms based on reductions to ASP for the contension inconsistency measure, the forgetting-based inconsistency measure, and the hitting set inconsistency measure. Moreover, we experimentally evaluated them against corresponding brute-force algorithms wrt. execution time. The evaluation showed that the novel ASP-based implementations perform clearly superior. We also learned that the naive implementations perform relatively worse when inconsistency values are large. The ASP encodings, on the other hand, are not dependent on the level of inconsistency.

With regard to future work, one aim is to utilize answer set programming to encode other inconsistency measures as well. For example, measures with higher computational complexity than those considered in this paper may be examined. Furthermore, it is of interest to investigate how our ASP-based algorithms perform in real-world applications. For instance, Nagel et al. presented a study about inconsistencies in business rules, which takes a quantitative perspective (Nagel, Corea, and Delfmann 2019), and thus could benefit from practically applicable algorithms. Other possible areas of application are mentioned in Section 1.

One of the insights gained throughout the course of this work is that large-sized knowledge bases are still problematic. With regard to datasets B and C, the ASP-based implementations for both $\mathcal{I}_f$ and $\mathcal{I}_h$ produced some timeouts, and with regard to dataset D, none of the three implementations could compute inconsistency values for a number of knowledge bases within the time limit. Therefore, another area of research that may be relevant with respect to the algorithmic perspective on inconsistency measures is that of approximate algorithms.

Acknowledgements This research is supported by the German Research Association (DFG), project number 424710479.

References

Bertossi, L. 2018. Measuring and Computing Database Inconsistency via Repairs. In *Proceedings of the 12th International Conference on Scalable Uncertainty Management (SUM'18)*.

Besnard, P. 2016. Forgetting-based inconsistency measure. In *International Conference on Scalable Uncertainty Management*, 331–337. Springer.

Besnard, P.; Doutre, S.; and Herzig, A. 2014. Encoding argument graphs in logic. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems - IPMU 2014*.

Brewka, G.; Eiter, T.; and Truszczynski, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12): 92–103.

Cholvy, L.; Perrussel, L.; and Thevenin, J.-M. 2017. Using inconsistency measures for estimating reliability. *International Journal of Approximate Reasoning* 89: 41–57.

Dung, P. M. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence* 77(2): 321–358.

Dvorák, W.; Gaggl, S. A.; Rapberger, A.; Wallner, J. P.; and Woltran, S. 2020. The ASPARTIX System Suite. In Prakken, H.; Bistarelli, S.; Santini, F.; and Taticchi, C., eds., *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, 461–462. IOS Press.

Erdem, E.; Gelfond, M.; and Leone, N. 2016. Applications of answer set programming. *AI Magazine* 37(3): 53–68.

Erdem, E.; Patoglu, V.; Saribatur, Z. G.; Schüller, P.; and Uras, T. 2013. Finding optimal plans for multiple teams of robots through a mediator: A logic-based approach. *Theory and Practice of Logic Programming* 13(4-5): 831–846.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Ostrowski, M.; Schaub, T.; and Wanko, P. 2016. Theory solving made easy with clingo 5. In *Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2012. Answer set solving in practice. *Synthesis lectures on artificial intelligence and machine learning* 6(3): 1–238.

Gelfond, M.; and Leone, N. 2002. Logic programming and knowledge representation—the A-Prolog perspective. *Artificial Intelligence* 138(1-2): 3–38.

Gelfond, M.; and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New generation computing* 9(3-4): 365–385.

Grant, J. 1978. Classifications for Inconsistent Theories. *Notre Dame Journal of Formal Logic* 19(3): 435–444.

Grant, J.; and Hunter, A. 2011. Measuring consistency gain and information loss in stepwise inconsistency resolution. In *Proceedings ECSQARU'11*, 362–373. Springer.

Grant, J.; and Hunter, A. 2017. Analysing Inconsistent Information Using Distance-Based Measures. *Int. J. Approx. Reasoning* 89(C): 3–26.

Grant, J.; and Martinez, M. V., eds. 2018. *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications.

Hunter, A. 2006. How to act on inconsistent news: Ignore, resolve, or reject. *Data & Knowledge Engineering* 57(3): 221–239.

Kuhlmann, I.; and Thimm, M. 2020. An Algorithm for the Contension Inconsistency Measure using Reductions to Answer Set Programming. In *International Conference on Scalable Uncertainty Management*, 289–296. Springer.

Lifschitz, V. 2008. What is answer set programming? In *Proceedings AAAI'08*, 1594–1597.

Martinez, A. B. B.; Arias, J. J. P.; and Vilas, A. F. 2004. On Measuring Levels of Inconsistency in Multi-Perspective Requirements Specifications. In *Proceedings of the 1st Conference on the Principles of Software Engineering (PRISE'04)*.

Nagel, S.; Corea, C.; and Delfmann, P. 2019. Effects of quantitative measures on understanding inconsistencies in business rules. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 146–155.

Potyka, N.; and Thimm, M. 2017. Inconsistency-tolerant reasoning over linear probabilistic knowledge bases. *International Journal of Approximate Reasoning* 88: 209–236.

Priest, G. 1979. Logic of Paradox. *Journal of Philosophical Logic* 8: 219–241.

Reiter, R. 1980. A logic for default reasoning. *Artificial intelligence* 13(1-2): 81–132.

Thimm, M. 2016. Stream-based inconsistency measurement. *International Journal of Approximate Reasoning* 68: 68–87.

Thimm, M. 2019. Inconsistency Measurement. In Amor, N. B.; Quost, B.; and Theobald, M., eds., *Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'19)*, 9–23. Springer International Publishing.

Thimm, M.; and Wallner, J. 2019. On the Complexity of Inconsistency Measurement. *Artificial Intelligence* 275.

An Argumentative Perspective to Conflictive Interpretation of Knowledge

Diego C. Martínez [1] , **Maximiliano C. D. Budán** [3] , **María Laura Cobo** [2] , **Antonino Rotolo** [4]

[1]Institute for Computer Science and Engineering (ICIC UNS-CONICET),
[2]Dept. of Computer Science and Engineering, Universidad Nacional del Sur, Argentina.
[3] Dept. of Mathematics, Universidad Nacional de Santiago del Estero, Argentina.
[4] CIRSFID, University of Bologna, Italy.
{dcm,mcdb,lc}@cs.uns.edu.ar, antonino.rotolo@unibo.it

Abstract

In many scenarios, a set of beliefs can be interpreted in different ways, leading to different outcomes. In this work we propose an argumentation-based view of interpretation of pieces of knowledge, using legal provisions as a leading example. We formalize conflicts and entailment towards a characterization of an acceptable, rational position of the agent on a set of knowledge, *i.e.* a subset of interpretations, inspired by argumentation semantics.

1 Introduction

There exist several scenarios in which pieces of information are subject to interpretation, for several reasons. A political discourse highlighting a set of beliefs, a plan of actions for financial investments, the official report of a setback in war and the set of norms in a legal system are all examples of structures of knowledge that require some form of interpretation in order to properly work with them. Generally speaking, to interpret a piece of information X is to provide a link between X and another piece of information about the meaning of X, the interpretation itself. In a lot of situations, more than one interpretation can be associated to a particular piece of information. For instance, if the police stops a driver and says "what is the emergency?", the driver may interpret that as a request for information about an actual ongoing emergency, or as a sarcastic way for referencing excessive speed. The first interpretation, a merely syntactic one, is probably not the intended one in the encounter. Nevertheless, one may argue that it is still a valid interpretation. This simple, perhaps funny, example serves to illustrate an elemental aspect of any set of beliefs: they are exposed to be interpreted in several ways leading to different outcomes.

A model of interpretations must take into account some intrinsic characteristics. Mainly, the fact that interpretations are not isolated units of knowledge and may be related to other interpretations. One interpretation may be in conflict with another, or it may be supporting another interpretation on a different piece of knowledge. For instance, the concept of "*freedom of speech*" ($k1$) may be interpreted as "*the right to express ourselves freely on any subject on any context*" ($i1$). This interpretation is in conflict with the one that states that freedom of speech is a limited right of expression that excludes offenses ($i2$). Following interpretation $i2$, a denial of Holocaust ($k2$) may be interpreted as a crime ($i3$). Conflict also arises on different interpretations for different pieces of knowledge. The interpretation $i3$ cannot be applied to $k2$ under interpretation $i1$ of $k1$. Hence, there are two rational standing positions here, namely $S_1 = \{i1\}$ and $S_1 = \{i2, i3\}$.

Given these conflicts and supports among interpretations, it is interesting to define a framework for the characterization of rational *standing positions* on a given set of knowledge, *i.e.* the identification of sets of interpretations with particular properties. We think abstract argumentation provides a pathway for the study of complex situations regarding multiplicity of interpretations, and here we propose an abstract formalism as a basic framework for this.

This paper is organized as follows. Section 2 analyzes the idea of *interpreted knowledge* in a logical scenario for these notions: the law and its defined norms. Section 3 presents the abstract framework to model provisions and interpretations. Mandatory and permitted interpretations are characterized. Section 4 discusses classical notions of argument semantics in the context of interpretations. In Section 5 a concordance relation between legal systems is introduced. Finally, in Section 6 we present a related works, and conclusions and future work in Section 7.

2 Law as interpretable knowledge

One of the most common scenarios where the interpretation of knowledge is relevant is the law. There are many ways to read legal texts, being this an important subject in legal studies. Legal interpretation is an essential method to assign a meaning to legal provisions, i.e., to determine the *content* of the law, often beyond the literal meaning of the legal texts (Greenberg 2017). However, due to the proper nature of texts and the human process of contextual understanding, there are constant debates over legal interpretation.

Recent formal studies refer to interpretations from the point of view of logic and computer science (Rotolo, Governatori, and Sartor 2015; Malerba, Rotolo, and Governatori 2016; Boella et al. 2010; Broezk 2013). Such previous works (in particular (Rotolo, Governatori, and Sartor 2015; Malerba, Rotolo, and Governatori 2016; Boella et al. 2010)) proposed complex rule-based systems for capturing several subtleties behind reasoning about interpretive canons. In (Maranhão 2017) a logical framework is proposed for the representation of legal interpretation. Interpretations are considered as a dynamic of theory change, where rules, values and meaning ascriptions are related and revised in order to reach a coherent explanation of the legal order. In (Walton, Sartor, and Macagno 2018a) a relation between argumentation and interpretations is explored. There, interpretive schemes are incorporated into a formal argumentation system such as Carneades or APSIC+ and then applied to displaying the pro–contra structure of the argumentation using argument maps applied to legal cases. These proposals provide logic details suitable to model specific, different aspects of legal interpretation.

In this paper we propose a contribution to the topic by defining a simpler approach, i.e., an abstract framework which deals with interpretations and conflicts between interpretive solutions. The focus is put in discovering the argumentation-like behaviour of the general interaction of interpretations linked to abstract pieces of knowledge. Consider the following example.

EXAMPLE 1. *Suppose the teenager Jim steals a horse for the first time in his life, and rides into the City Park. Jim is arrested and put on trial. Consider the following provisions:*

- n_1 = *"A man stealing a horse should be punished with jail."*
- n_2 = *"Vehicles are not allowed in the pedestrian area, and fines should be imposed"*
- n_3 = *"Unclaimed recovered vehicles in the Police Lot will be sent to the car shredder machine after three months."*

and consider the following interpretations:

- *for* n_1*:*
  ϕ_a=*"A man is an adult male",*
  ϕ_b=*"A man is a person of any age and gender"*
- *for* n_2*:*
  ϕ_c=*"A vehicle is a machine that transports people"*
  ϕ_d=*"A vehicle is any form of transportation used by humans".*
- *for* n_3*:*
  ϕ_e=*"A horse can be killed by police after three months"*
  ϕ_f=*"Only cars can be sent to car shredder"*

Provision n_1 seems to indicate that Jim faces a time in jail. However, interpretation ϕ_a considers that the reference to "*man*" is about an adult male, and then provision n_1 cannot be applied to Jim, a young teenager. What makes provision n_1 relevant here is an interpretation of the word "man" as a reference to any human being, that is, interpretation ϕ_b. Clearly, there is a *conflict* between ϕ_a and ϕ_b. On the other hand, provision n_2 establishes that, additionally, Jim should pay a fine for entering the Park. This makes sense only under interpretation ϕ_d that considers a horse a legal vehicle. Under interpretation ϕ_c, however, the fine cannot be applied since a horse is obviously an animal and not a machine. Again, it is not possible to accept interpretation ϕ_c and ϕ_d simultaneously and then both interpretations are in *conflict*. Provision n_3 establishes the destination for storing vehicles that are not claimed by its owners after a certain period of time. Note that if a horse is considered a vehicle (interpretation ϕ_d) then the horse must be sent to the car shredder, *i.e.* ϕ_e is the reasonable interpretation for n_3. In this case we say that interpretation ϕ_d *entails* ϕ_e. Also, there is a conflict between interpretation ϕ_c and ϕ_e because since a horse is not considered a vehicle, then its life is not at risk.

Conflicts and entailments are two basic elements of argumentation and then an argumentative analysis of the set of interpretations using abstract frameworks is interesting and constitutes a novel approach in the literature. The overall scenario deserves further studies. Some interpretations may be the only ones that a rational agent may adopt given the sets of conflicts and entailments. For instance, suppose in Example 1 there is only one interpretation for n_3, say ϕ_e. Then, since there are no alternative interpretations for n_3, the only valid non-conflictive interpretation for n_2 is ϕ_d.

In the following section we present the abstract formalism for knowledge and interpretations.

3 Abstract Framework for Interpretations

In this work, provisions and interpretations are treated abstractly, leading out their logical structures and representing the possible conceptual relationships between them. Thus, we define a framework where these elements are formalized, together with two distinct relations between interpretations.

Definition 1. *An interpretative framework is defined as* $\langle Pr, I, Ln, C, T\rangle$*, where*

- *Pr is a set of abstract legal provisions, denoted $n_1, n_2, \ldots$*
- *I is a set of abstract interpretations, denoted $\phi_1, \phi_2, \ldots$ providing a sentential meaning to any provision n.*
- *$Ln : Pr \to 2^I$ a function denoting the set of all the interpretations for a given provision.*
- *$C \subseteq I \times I$ is a symmetric conflict relation between interpretations.*
- *$T \subseteq I \times I$ is the entailment relation between interpretations.*

The interpretative framework characterizes an abstract legal system, formed by provisions, the universe of interpretations for every one of them, and two simple relations between interpretations: conflicts and entailments. The symmetric conflict relation between interpretations models the fact that some interpretations cannot be adopted simultaneously. Hence if $(\phi_1, \phi_2) \in C$ then whenever interpretation ϕ_1 is adopted, ϕ_2 should be not, or vice-versa being C symmetric. On the other hand, relation T establishes an *entailment* relation between interpretations. If $(\phi_1, \phi_2) \in T$ then interpretation ϕ_2 should be adopted given the adoption of interpretation ϕ_1. In this direction, we can specify a sequence of interpretation under an entailment relation. Formally:

Definition 2. *Let $\langle Pr, I, Ln, C, T\rangle$ be an interpretative framework. We define a sequence of interpretations under entailment relation as $(\phi_1, \phi_2) \in T, (\phi_2, \phi_3) \in T \ldots, (\phi_{n-1}, \phi_n) \in T$. We will denote this sequence of entailments as $(\phi_1, \phi_n)^\star$.*

For a particular purpose, usually a subset of the legal system is considered. We characterize then a restricted set of provisions equipped with a selection of interpretations, called here *dossier*.

Definition 3. *A dossier $\mathscr{D}$ is an ordered set of pairs $(n_1, S_1), (n_2, S_2), .., (n_n, S_n)$ where (n_i, S_i) is such that n_i is* *a provision and $S_i \subseteq Ln(n_i)$. The set of interpretations of the dossier $\mathscr{D}$ is defined as $\mathscr{D}^I = \bigcup S_i, 1 \leq i \leq n$.*

The dossier is a collection of provisions to be considered as a whole for some legal purpose, such as a criminal case, civil action or a legislative reformation. Every provision has attached a set of relevant interpretations that *may* be applied to that provision. The pair (n_i, S_i) states that provision n_i could be interpreted as any of the members of S_i.

It is possible for some interpretations in a dossier to be in conflict. This may occur between interpretations of a single provision (*i.e.*, inside S_i) called *intra-provision conflicts* or between interpretations of different norms (*i.e.*, an interpretation of S_i in conflict with an interpretation of S_j), called *inter-provisions* conflicts.

Definition 4. *A dossier $\mathscr{D}$ is said to be consistent if $\mathscr{D}^I$ is conflict-free.*

A *consistent* dossier is such that any provision can eventually be interpreted in any of the given alternatives. It represents a legal system with no conflictive interpretations on any provision. However, a non-consistent dossier requires further examination, since a selection of interpretations must be addressed. Suppose $(n_1, \{\phi_a, \phi_b\})$ and $(n_2, \{\phi_c, \phi_d\})$ are in dossier $\mathscr{D}$, such that $(\phi_a, \phi_c) \in C$. Here there is a risk to interpret two different provisions under a contradiction: according to the legal framework, ϕ_a and ϕ_c are not compatible. If n_1 is interpreted as ϕ_a then provision n_2 should not be interpreted as ϕ_c. In other words, the set $\{\phi_a, \phi_c\}$ is not a rational interpretation of the dossier as a whole. It constitutes indeed a *position* of the rational agent towards the dossier, although contradictory. On the other hand, the set $\{\phi_a, \phi_d\}$ represents a *conflic-free position* towards provisions n_1 and n_2. Note that here, in order to avoid conflicts, n_2 must be interpreted as ϕ_d because the dossier does not allows other interpretations for n_2. This constitutes an obligation for the agent, which we will address in later sections.

Given a dossier, which is simply a set of legal provisions equipped with plausible interpretations, a rational agent may adopt a particular view of every provision, adopting then a *position* about them, formalized as follows.

Definition 5. *Let $\mathscr{D}$ be dossier. A* position *for $\mathscr{D}$ is a set of interpretations $\Phi \subseteq \mathscr{D}^I$ such that for every norm (n_i, S_i) in $\mathscr{D}$ it holds that $\Phi \cap S_i \neq \emptyset$. A position Φ is said to be definite if $|\Phi \cap S_i| = 1$ for every S_i. The restriction of $\mathscr{D}$ to position Φ is defined as $\mathscr{D}(\Phi) = \{(n_i, \Phi \cap S_i), 1 \leq i \leq n\}$. The set of all of positions for $\mathscr{D}$ is denoted as $\mathscr{D}^*$*

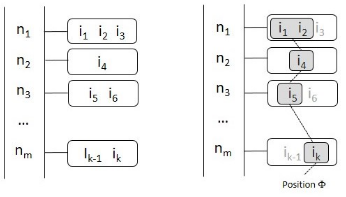


Figure 1: A dossier and a position.

A position is simply a selection of interpretations for *every* provision. The restriction of a dossier is simply the pairing of its provisions with the selected interpretations of a given position. Note that $\mathscr{D}^I$ is also a position of $\mathscr{D}$, since it includes every possible interpretation. It is in fact the most general position that can be defined on $\mathscr{D}$.

Some provisions may receive more than one interpretation, which may be even still in conflict with other interpretations. Hence, even as a subset of $\mathscr{D}^I$, a position is not necessarily free of conflicts.

PROPOSITION 1. *Any restriction of a consistent dossier is also a consistent dossier.*

Since the characterization of *rational* interpretative positions is our main subject, positions that are free of conflicts are of primary attention. These positions, applied to the provisions in the dossier, yields to a set of legal norms under consistent interpretations.

Definition 6. *Let $\mathscr{D}$ be dossier. A* position *Φ for $\mathscr{D}$ is said to be sound if there are no $\phi_a, \phi_b \in \Phi$ such that $(\phi_a, \phi_b) \in C$. A sound position Φ is said to be maximal if there is no sound position Φ' such that $\Phi \subset \Phi'$.*

A sound position Φ for a dossier $\mathscr{D}$ makes $\mathscr{D}(\Phi)$ consistent, and then it represents a reasonable set of interpretations that can be adopted. Thus, *soundness* is the first, most basic notion of rational stand towards a dossier as a whole. In fact, in a consistent dossier any position is sound.

EXAMPLE 2. *Consider the running example about Jim and the horse. It can be represented by the dossier $\mathscr{D}_2 = \{(n_1, \{\phi_a, \phi_b\}), (n_2, \{\phi_c, \phi_d\}), (n_3, \{\phi_e \phi_f\})\}$ such that $(\phi_a, \phi_b), (\phi_c, \phi_d), (\phi_c, \phi_e), (\phi_e, \phi_f) \in C$ and $(\phi_d, \phi_e) \in T$. Position $\Phi_1 = \{\phi_a, \phi_c, \phi_f\}$ is sound and corresponds to the position affirming that "Jim should not be charged since it is a boy and horses are not vehicles". On the other hand, position $\Phi_2 = \{\phi_b, \phi_d\}$ is also sound and corresponds to the position stating that "Jim is a person and should be charged of stealing and making use of vehicle in a pedestrian area". Note that position $\Phi_3 = \{\phi_a, \phi_c, \phi_e\}$ is not sound since it uses contradictory interpretations.*

As showing in the previous example, there may be several sound positions for a dossier. On these alternatives, a primary notion of *mandatory* interpretation emerges, as illustrated in the following example.

EXAMPLE 3. *Let $\mathscr{D}_3 = \{(n_1, \{\phi_1, \phi_2\}), (n_2, \{\phi_3, \phi_4\}), (n_3, \{\phi_5\})\}$ be a dossier such that (ϕ_1, ϕ_4), $(\phi_2, \phi_4) \in C$. Here interpretation ϕ_4 is in conflict with all of the interpretations for n_1. Since a position, as such, must provide an interpretation for n_1, no sound position can include ϕ_4. There are only three sound positions: $\Phi_1 = \{\phi_1, \phi_3, \phi_5\}$, $\Phi_2 = \{\phi_2, \phi_3, \phi_5\}$ and $\Phi_3 = \{\phi_1, \phi_2, \phi_3, \phi_5\}$.*

In Example 3, interpretation ϕ_5 is the only interpretation for provision n_3. For some lawyers, n_3 is a well-written provision, without alternative interpretations. On the other hand, interpretation ϕ_3 is not the only one provided for n_2, but it is the only interpretation that can be consistently selected for n_2. Therefore, they are a *necessity* in order to construct a sound position for the dossier. Interpretations ϕ_1

and ϕ_2 are permitted, although not mandatory since there is one position excluding one of them.

Definition 7. *Let $\mathscr{D}$ be a dossier. An interpretation is said to be mandatory in $\mathscr{D}$, if it is included in every sound position of $\mathscr{D}$. An interpretation is said to be permitted if it is not mandatory and it is included in at least one sound position.*

The trivial reason for an interpretation ϕ_i to be a *necessity* is because it is the only one provided for a provision n_i. If there are more than one interpretation for provision n_i, then in order to be ϕ_i a necessity, it must be the only "survivor" in the overall scene of interpretations and conflicts for that provision, just like ϕ_3 in Example 3.

Example 4. *Let $\mathscr{D}_4 = \{(n_1, \{\phi_1\}), (n_2, \{\phi_2\})\}$ be a dossier such that $(\phi_1, \phi_2) \in C$. The only position for $\mathscr{D}_4$ is $\Phi_1 = \{\phi_1, \phi_2\}$ and it is not sound. These interpretations are not considered mandatory for $\mathscr{D}_4$.*

Hence, necessity as a mandatory act of interpretation, makes sense towards a non-contradictory stand for a legal system. In Example 4 there are no other choices to interpret both provisions and it is impossible to avoid contradiction. The problem here is the dossier, lacking of sound positions, being then a non-consistent set of norms.

Although, as stated in Definition 7, sound positions are the basis for determining necessities in a dossier, the analysis is not complete since in order to model a rational stand for a legal system, the entailment between interpretations must be taken into account. This relation models a different concept of obligation, where the use of some interpretation for a given provision may result in the adoption of others for different provisions. We may call these as *interpretations as a consequence*. This notion is explicitly characterized in the abstract interpretive framework. As stated before, relation T models an *entailment* relation between interpretations. If interpretation ϕ_a entails ϕ_b, then $(\phi_a, \phi_b) \in T$ denoting that ϕ_b should be adopted given the adoption of interpretation ϕ_a. This has an effect on positions, since some interpretations are explicitly entailed. Suppose there are two norms in a dossier $(n_1, \{\phi_{11}, \phi_{12}\})$,$(n_2, \{\phi_{21}, \phi_{22}\})$ such that $(\phi_{11}, \phi_{21}) \in C$. Any sound position including ϕ_{11} cannot include ϕ_{21} and viceversa. Suppose now that ϕ_{12} *entails* ϕ_{21}. Then the sound position $\{\phi_{12}, \phi_{21}\}$ is somehow *better* than the sound position $\{\phi_{12}, \phi_{22}\}$, since in the former one interpretation entails the other. In fact, $\{\phi_{12}, \phi_{22}\}$ violates the entailment by choosing a different interpretation for provision n_2. Hence, this position should not be valid according to entailments.

Definition 8. *A position Φ is said to be closed if it includes every interpretation ϕ_i such $\exists \phi_j \in \Phi, (\phi_j, \phi_i) \in T$ and $(\phi_i, \phi_k) \notin C$ for any $\phi_k \in \Phi$.*

A closed position Φ includes every entailed interpretation that is not in conflict with Φ. Closed positions are not necessarily definite, since they must include some interpretations because of the entailment relation. Hence, there may be a provision with more than one interpretation in a closed position.

Due to entailments, there is another level of inconsistency within a position. Note that in the previous example any position including $\{\phi_{11}, \phi_{12}\}$ is somehow contradictory in the sense that these interpretations are in conflict with, yet entailing, the same interpretation ϕ_{21}. This is formalized in the following definition.

Definition 9. *Let $\mathscr{D}$ be a dossier and let $\Phi \subseteq \mathscr{D}^I$. The position Φ is said to be internally coherent if it is conflict-free and $\nexists \phi_j \in \mathscr{D}^I$, such that $(\phi_m, \phi_j) \in C$ and $(\phi_n, \phi_j)^\star$ is possible, for some $\phi_m, \phi_n \in \Phi$.*

A position is internally coherent if, besides being conflict-free, it does not entails an interpretation that falls into conflict with itself. It is possible for a position to be sound and not internally coherent.

Example 5. *Let $\mathscr{D}_5 = \{(n_1, \{\phi_1\}), (n_2, \{\phi_2, \phi_3\}), (n_3, \{\phi_4\})\}$ be a dossier such that $(\phi_1, \phi_3) \in C$ and $(\phi_4, \phi_3) \in T$. The position $\Phi = \{\phi_1, \phi_2, \phi_4\}$ is closed. It does not include the entailed interpretation ϕ_3 because it is in conflict with ϕ_1. Although it is sound, this position is not internally coherent, because it entails interpretation ϕ_3 that is in conflict with $\phi_1 \in \Phi$.*

The dossier of Example 5 has the particularity that the only sound position is not internally coherent. However, ϕ_1 and ϕ_4 are the only available interpretations for provisions n_1 and n_3 respectively. Are these positions a necessity for dossier $\mathscr{D}_5$? Indeed they are, for a lack of better interpretations. But the problem here, just as in Example 4, is the dossier: this selection of provisions and interpretations is not rational in the sense that a contradiction is present.

Definition 10. *Let $\mathscr{D}$ be a dossier. A position Φ for $\mathscr{D}$ is said to be robust if it is closed.*

A robust position is a semantic concept characterizing a rational selection of interpretations for a dossier, where conflicts and entailments are observed. In a robust position there are no conflicts nor conflictive interpretations are entailed. This position is not unique and a dossier may have several robust extensions. Or it may have none, as in the dossier of Example 5.

Dossiers of Example 4 and 5 are problematic. Both of them are populated with provisions and interpretations in such a way that no internally coherent positions can be induced. It can be viewed as a legal system in which the interpreter of the law is forced to incur in contradiction. Any law with this characteristic behavior should be revised.

Definition 11. *A dossier is said to be well-formed if it has at least one robust position.*

Hence, our concept of necessity on interpretations only applies to well-formed dossiers, where there is an open criterion of interpretations in all the provisions that allows to any agent, beyond its particular bias, to adopt a non-contradictory position towards this notion of legal system.

Next, we analyze positions from the point of view of argumentation semantics. This is interesting since the set of interpretations and its conflicts resembles a symmetric argumentation framework (Coste-Marquis, Devred, and Marquis 2005). Hence, some classic argumentation semantics can be applied.

4 Argumentation Semantics on Interpretations

For a given dossier $\mathscr{D}$, a symmetric argumentation framework $AF_{\mathscr{D}}$ may be induced, where $AF_{\mathscr{D}} = \langle \mathscr{D}^I, C\downarrow\mathscr{D}^I \rangle$ formed by the set of interpretations of the dossier and the corresponding conflict relation on these interpretations only. If we take the entailment relation into account, it is similar to bipolar argumentation frameworks (Cayrol and Lagasquie-Schiex 2005).

An interpretation ϕ *is acceptable with respect to a set of interpretations* S if whenever ϕ_j is in conflict with ϕ, an interpretation of S is in conflict with ϕ_j. A set S of interpretations is admissible if every interpretation in S is acceptable with respect to S. Since every conflict is symmetric, from the point of view of admissibility, every interpretation is defended by itself. As a consequence, $\{\phi\}$ is an admissible set for any interpretation ϕ. Clearly, maximal admissible sets are of interest.

PROPOSITION 2. *Every sound position is an admissible set of interpretations.*

Preferred extensions are maximal (w.r.t. set inclusion) admissible sets and they are not necessarily unique. They provide a set of interpretations free of conflict that can be applied to the dossier. Even more, in a symmetric framework, every preferred extension is stable. This means that the extension is in conflict with every interpretation outside the set, which seems to capture a strong adoption of interpretations.

PROPOSITION 3. *Since conflicts between interpretations are symmetric, every interpretation belongs to at least one preferred extension.*

Preferred extensions are defined from interpretations without taking provisions into account. Suppose α is a preferred extension of AF. Since it is an admissible set, there are no conflictive interpretations in α. However, it may not provide interpretations for some provisions. In other words, not every preferred extension is a position for $\mathscr{D}$.

EXAMPLE 6. *In the framework of Example 4, there are only two preferred extensions $S_1 = \{\phi_1\}$, and $S_2 = \{\phi_2\}$. Both extensions fail to provide an interpretation for a norm in the dossier.*

The coverage of interpretations under preferred extensions, however, provide an indication of coherence for a dossier. If, for a given dossier, every preferred extension fails to provide an interpretation for some provision (of course, possibly not the same), then the dossier has no sound position and vice-versa.

PROPOSITION 4. *A dossier $\mathscr{D}$ has no sound position if and only if every preferred extension of $AF_{\mathscr{D}}$ leaves a provision of $\mathscr{D}$ without interpretation*

Proof ($\Rightarrow$): Consider all the preferred extensions, $E_1, E_2, .., E_n$, of $AF_{\mathscr{D}}$. Suppose that $AF_{\mathscr{D}}$ has no sound position, but there is a an extension E_i, $1 \leq i \leq n$, such that provides an interpretation for each provision. Since E_1 is conflic-free and provides an interpretation for each provision then it is also a sound position, wich is absurd.

Proof ($\Leftarrow$): Suppose $E_1, E_2, .., E_n$ are the preferred extensions of $AF_{\mathscr{D}}$ such that a provision n_i has not an interpretation in E_i, $1 \leq i \leq n$. Suppose there is a sound position $P = \{\phi_1, \phi_2, ..., \phi_m\}$. If P is sound, then it is conflict free. If P is conflict free, then P is admissible (since the framework is symmetric) and then P is included in at least one preferred extension E_k for some $1 \leq k \leq n$. But then E_k provides an interpretation for every norm in $\mathscr{D}$, which is absurd.

Some interpretations may be free of conflicts. In a symmetric framework, these interpretations constitute the grounded extension. The grounded extension is the least complete extension with respect to set inclusion, representing the skeptical point of view. A *complete* extension S includes every interpretation that is acceptable with respect to S. Hence, given a dossier $\mathscr{D}$ with corresponding framework $AF_{\mathscr{D}}$, the grounded extension of $AF_{\mathscr{D}}$ is defined as $GE(AF_{\mathscr{D}}) = \{a \in \mathscr{D}^I | \nexists b \in \mathscr{D}^I, (a,b) \in C\downarrow\mathscr{D}^I\}$. These interpretations are included in every *maximal* position of the dossier.

REMARK 1. *Interpretations in $GE(AF_{\mathscr{D}})$ are not necessarily mandatory. Although these interpretations are included in every preferred extension, there may be non-maximal sound positions excluding some of them, if they are alternative interpretations for the same provision.*

In bipolar argumentation frameworks, an indirect conflict arises when an argument A supports another argument B which attacks C. In this case, there is certain contradiction between A and C, since the former supports an attacker of the latter. In our interpretative framework there is a similar situation, although our notion of entailment has a different meaning than the notion of support. An interpretation may entail another, which in turn may be in confict with a third interpretation. This indirect conflict is captured in Definition 9, inspired by the same situation in bipolar frameworks. We do not, however, consider the entailment relation as a support relation that strengthens or weakens the consequent.

5 Legal Doctrines: Interpretations as a Principle

As stated before, there is another form of *mandatory* interpretation besides the one defined in Definition 7. Some interpretations must be adopted as a *principle*, *i.e.*, there is a fundamental point of view, constituting a doctrine, that demands the use of these interpretations. For instance, political ideologies may define particular interpretations on some civil rights as freedom of speech, or a high-level judicial institution may promote only some interpretations, hence constituting a legal doctrine on some aspects of the legal system. We call this kind of mandatory interpretation an *interpretation as a principle*. Then, in some contexts, legal provisions may have only a reduced sets of acceptable interpretations, even when more interpretations exists. Since these legal stands also involves provisions and interpretations, they can be modelled as dossiers. The question then is how a dossier *conforms* to another referential dossier according to positions on interpretations. They can refer to different provisions although with different sets of interpretations. This is formalized as follows.

Definition 12. *Let $\mathscr{D}_1, \mathscr{D}_2$ be two dossiers. We say that $\mathscr{D}_2$ conforms to $\mathscr{D}_1$, denoted $\mathscr{D}_1 \lhd \mathscr{D}_2$, iff $\mathscr{D}_2^I \subseteq \mathscr{D}_1^I$.*

A dossier $\mathscr{D}_2$ conforms to another dossier $\mathscr{D}_1$ if the former includes provisions with a (possibly) reduced set of interpretations. Hence, any position for $\mathscr{D}_2$ also provides a position for $\mathscr{D}_1$. Note that not necessarily a position for $\mathscr{D}_1$ is a position for $\mathscr{D}_2$ since $\mathscr{D}_1$ may have more interpretations than $\mathscr{D}_2$.

REMARK 2. *For any dossier $\mathscr{D}$, it holds that $\mathscr{D} \lhd \mathscr{D}$ and $\mathscr{D} \lhd \mathscr{D}(\Phi)$ for any position Φ.*

Given the conformance relation, it is possible to evaluate dossiers according to the point of view of a referential dossier. Suppose dossier $\mathscr{D}_1$ includes the pair $(n_a, \{\phi_a\})$. It means that the only valid interpretation for provision n_a is ϕ_a, *i.e.*, this interpretation is mandatory. Thus, any dossier $\mathscr{D}_i$ conforming $\mathscr{D}_1$ is obligated to adopt interpretation ϕ_a. In other words, there may be a position on $\mathscr{D}_i$ that leads to a restriction of that dossier such that this restriction conforms to $\mathscr{D}_1$.

EXAMPLE 7. *Let $\mathscr{D}_a = \{(n_1, \{\phi_1\})\}$ and $\mathscr{D}_b = \{(n_1, \{\phi_1, \phi_2\}), (n_2, \{\phi_3\})\}$ be two dossiers. Dossier $\mathscr{D}_b$ does not conform to $\mathscr{D}_a$ since it includes another interpretation for n_1. However, for position $\Phi = \{\phi_1, \phi_3\}$, the restriction $\mathscr{D}_b(\Phi)$ does conforms to $\mathscr{D}_a$.*

A particular position may lead then to a restriction satisfying conformity. The conformance relation then *induces* some interpretations in other, non-conforming dossiers towards the satisfaction of conformity. Although maybe $\mathscr{D}_1 \not\lhd \mathscr{D}_2$ it is possible that $\mathscr{D}_1 \lhd \mathscr{D}_2(\Phi)$ for a some position Φ. It turns out then that some interpretation is considered mandatory for $\mathscr{D}_1$ not because of its constant presence in semantic extensions (such as sound positions), but because it is required to conform to a referential dossier.

Definition 13. *An interpretation ϕ is mandatory in $\mathscr{D}$ according to $\mathscr{D}'$ if ϕ is in every robust position Φ of $\mathscr{D}$ such that $\mathscr{D}' \lhd \mathscr{D}(\Phi)$.*

This means that, for every position Φ that makes $\mathscr{D}'(\Phi)$ able to conform to $\mathscr{D}$, the interpretation ϕ is always present. Hence, the dossier $\mathscr{D}$ marks a referential point of view for the interpretation of dossier $\mathscr{D}'$, by *filtering* some alternative positions. This concept of conformity is simple since the underlying idea is to properly share interpretations. However, the entailment relation provides a more subtle notion of conformity.

EXAMPLE 8. *Consider $\mathscr{D}_{It} = \{(n_1, \{\phi_a, ...)(n_2, \{\phi_b ...)\}$ where $\phi_a =$"Nationalisation of companies is against the Treaty of Rome" and $\phi_b =$"The Treaty of Rome does not apply since a subsequent national statute applies" Let $\mathscr{D}_{EU} = \{(n_3, \{\phi_c\})\}$ where $\phi_c =$"European treaties cannot be overruled by domestic legal provisions" Here dossier $\mathscr{D}_{It}$ does not conforms to $\mathscr{D}_{EU}$ since interpretations are different. However, clearly $(\phi_c, \phi_a) \in T$. Hence, technically a position for $\mathscr{D}_{It}$ that includes ϕ_a may be in concordance with $\mathscr{D}_{EU}$, since the only mandatory interpretation ϕ_c entails the one selected for $\mathscr{D}_{It}$. Moreover, since $(\phi_a, \phi_b) \in C$, this notion of conformity, by preferring ϕ_a, forbids the use of ϕ_b. The position for the dossier $\mathscr{D}_{It}$ implies that the Treaty of Rome must prevail, despite other reasons for and against the intention of the demandant.*

The revised notion of conformity then goes beyond the use of the exact same interpretation for two dossiers, and consider the entailment relation as an enabling mechanism for positions.

Definition 14 (Revised)**.** *Let $\mathscr{D}_1, \mathscr{D}_2$ be two dossiers. We say that $\mathscr{D}_2$ conforms to $\mathscr{D}_1$, denoted $\mathscr{D}_1 \lhd \mathscr{D}_2$, iff every interpretation of $\mathscr{D}_2$ is either (a) an interpretation of $\mathscr{D}_1$ or (b) an interpretation ϕ entailed by an interpretation of $\mathscr{D}_1$ in such a way that $\mathscr{D}_1^I \cup \{\phi\}$ is internally coherent.*

In order to conform to a dossier $\mathscr{D}_1$, the exact same interpretations can be selected, or new interpretations that are entailed by $\mathscr{D}_1$ as long as it does not introduces a conflict, either in a direct way or through entailments. According then to Definition 14, in Example 8 a position that includes ϕ_a is able to conform to the dossier $\mathscr{D}_{EU}$ regarding the precedence of normative systems.

6 Related works

Several works in the literature of AI and Law explore how norms and their interpretation are models to improve the analysis of a specific legal domain. In this direction, the argumentation community address the interpretations of norms in a legal context from two perspectives: from an abstract point of view where norms and interpretations are abstract entities that interact in a certain way, or from a structured point of view based on logical language norms are studied at a higher level of description.

Kawasaki et al. in (Kawasaki, Moriguchi, and Takahashi 2018) preset a work where a transformation from the legally descriptive language PROLEG to a BAF. Thus, they create a bipolar model from a PROLEG program and present a semantic where the meaning of legal reasoning was preserved. To do that, first, the authors need the underlying PROLEG program providing a legal description of the domain. However, an abstract model that captures certain aspects like the provision with their possible interpretations and how they are linked is difficult to discern without the underlying logical description. In this sense, our work provides the tools to represent abstractly a legal scenario without a logical legal description about: provision and possible interpretation about such provision. Then, based on conceptual analysis, we identify permitted and mandatory interpretations specifying a specific legal position. Finally, the classical argumentation semantics are refined in the legal context, preserving some special properties.

In another direction, Malerba et al. in (Malerba, Rotolo, and Governatori 2016) present a logical formalism to treat with canons of interpretation coming from different legal systems. Thus, the authors defining a logic-based conceptual framework that could encompass the occurring interpretive interactions without neglecting the existing, broader normative background each legal system is nowadays part of. The spirit of this work is aligned with ours work, only that we treat the problem from an abstract point of view. Also, as future work, we intend to couple the theories from the possible worlds, where it would be possible to analyze how different legal systems can interact with each other according to a specific legal position.

From an abstract point of view, Bench-Capon and Modgil present in (Bench-Capon and Modgil 2009) a work where the capability of the extended abstract argumentation framework and the tools provided by the valued-based argumentation framework are combined to analyze a legal argumentation discussion. Briefly speaking, after considering the attacks between arguments and the attacks between attacks (giving legal mining about that), where arguments and attacks have assigned a preference order by an audience helping to resolve the conflicts, they arrive into a meta-level argumentation framework where the arguments have a legal value that they promote (social value, relevance level, ethical ideas, among other interpretations). Finally, valued-based semantics are applied to obtain an admissible set of arguments with the corresponding promoted value. In our work, the principal issue represents how provision can be interpreted, giving the place different kinds of conflict. More specifically, we see more inside the argument, splitting it according to provisions and the possible interpretation from each of them. However, it is interesting for future works to combine these research lines to obtain a set of admissible interpretations for a provision with the legal value that they promote, giving more information about the acceptance.

Finally, Walton et al. in (Walton, Sartor, and Macagno 2018b) carry out an in-depth study of how it is possible to interpret the arguments from the law. They argue that the justification of an interpretation can be regarded as an argumentation-based procedure in which the best interpretation is the one supported by the strongest or less defeasible set of arguments. Thus, to analyze an argument considering two points of view: the study of the possible interpretation associated with a provision and the argumentation scheme to study the argument strength. They show how the interpretation of provisions can be translated into argumentation schemes, and they distinguished two general macro-structures for positive and negative, total and partial provisions, under which various types of schemes and rebuttals can be classified. This classification was then used for modeling the interpretive arguments in a formal manner and integrating them into computational systems. Based on the above, our work is related to how interpretations are selected, conditioned, and analyzed to put a certain provision into context using our semi-structured argumentation framework. However, a way to improve our formalism is to consider the argumentation scheme (based on the expert opinion or cause-effect schemes) to specify another dimension of the provision interpretation quality or impact.

7 Conclusions and Future Work

In this work we proposed an abstract framework for semantic elaborations about provisions and interpretations. Two relations between interpretations are modelled: a conflict relation and an entailment relation. The former states that two interpretations are somehow incompatible and cannot be adopted simultaneously, while the latter establishes that some interpretations must be adopted as a consequence of other interpretations. The notion of legal dossier is introduced, as a set of provisions with some available interpretations. Using this structure, different qualities of positions (in the form of set of provisions) towards the dossier are introduced, such as sound and robust positions, and the relation to basic argumentation semantics are established. Later on, we explored the notion of mandatory and permitted interpretations, first for a stand-alone dossier and later under the use of another, second dossier as a referential legal system.

As stated before, this provides a general view of the argumentation-based behaviour of the interaction of interpretations applied to pieces of knowledge, showing how argumentation semantics can be applied to the basic quest of identifying rational standing positions. The abstract level is very high, inspired by classical abstract argumentation, by simply treating with the elemental relation knowledge-interpretation. Although legal reasoning is the leading field of study, the framework can be applied to different contexts, such as the analysis of detailed political platforms, newsfeeds, religion studies and any other situation in which potential conflictive interpretations can be applied to formalized knowledge. Legal reasoning is, however, a natural scenario for the consideration of concepts formalized in this article and the prime source of inspiration for the notion of abstract standing positions.

Future work has several directions. As one of our kindest reviewers mention, the work deals with some “incomplete argument” where parts of the support are missing and could be completed in different manners. In that sense, concepts as ”legal provision” and ”interpretation” seem to be related. Some discussion on those relations should be added in any case see (Black and Hunter 2012). We are interested in more semantic elaborations regarding positions across legal systems using dossiers as the basic structure by adding new relations between interpretations such as equivalence or preference order. We use entailment as positive relation among interpretations, but other forms of positive relations can be analyzed. We are also interested in the characterization of conflicts between norms, either by their intrinsic nature or due to conflictive interpretations. In order to achieve a proper level of detail, logic language could be used to represent provisions.

References

Bench-Capon, T., and Modgil, S. 2009. Case law in extended argumentation frameworks. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, 118–127.

Boella, G.; Governatori, G.; Rotolo, A.; and van der Torre, L. W. N. 2010. A logical understanding of legal interpretation. In Lin, F.; Sattler, U.; and Truszczynski, M., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010.* AAAI Press.

Broezk, B. 2013. Legal interpretation and coherence. In Araszkiewicz, M., and Savelka, J., eds., *Coherence: Insights from Philosophy, Jurisprudence and Artificial Intelligence*. Springer. 113–122.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In Godo, L., ed., *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 378–389. Berlin, Heidelberg: Springer Berlin Heidelberg.

Coste-Marquis, S.; Devred, C.; and Marquis, P. 2005. Symmetric argumentation frameworks. In Godo, L., ed., *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 317–328. Berlin, Heidelberg: Springer Berlin Heidelberg.

Greenberg, M. 2017. What makes a method of legal interpretation correct? legal standards vs fundamental determinants. In *Harvard Law Review Forum, Vol 130. Num 4*. 105–126.

Kawasaki, T.; Moriguchi, S.; and Takahashi, K. 2018. Transformation from proleg to a bipolar argumentation framework. In *SAFA@ COMMA*, 36–47.

Malerba, A.; Rotolo, A.; and Governatori, G. 2016. Interpretation across legal systems. In Bex, F., and Villata, S., eds., *Legal Knowledge and Information Systems - JURIX 2016: The Twenty-Ninth Annual Conference*, volume 294 of *Frontiers in Artificial Intelligence and Applications*, 83–92. IOS Press.

Maranhão, J. S. A. 2017. A logical architecture for dynamic legal interpretation. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, 129–138. New York, NY, USA: Association for Computing Machinery.

Rotolo, A.; Governatori, G.; and Sartor, G. 2015. Deontic defeasible reasoning in legal interpretation: two options for modelling interpretive arguments. In Sichelman, T., and Atkinson, K., eds., *Proceedings of the 15th International Conference on Artificial Intelligence and Law, ICAIL 2015, San Diego, CA, USA, June 8-12, 2015*, 99–108. ACM.

Walton, D.; Sartor, G.; and Macagno, F. 2018a. *Handbook of Legal Reasoning and Argumentation*. chapter Statutory interpretation as argumentation, 519–560.

Walton, D.; Sartor, G.; and Macagno, F. 2018b. Statutory interpretation as argumentation. In *Handbook of Legal Reasoning and Argumentation*. Springer. 519–560.

Autonomous Decision-Making with Incomplete Information and Safety Rules based on Non-Monotonic Reasoning

José-Luis Vilchis-Medina[1] , **Karen Godary-Déjean**[2] , **Charles Lesire**[3]

[1,3]ONERA/DTIS, University of Toulouse, France
[2]LIRMM/EXPLORE, University of Montpellier, France
{jose.vilchis_medina,charles.lesire}@onera.fr, karen.godary-dejean@umontpellier.fr

Abstract

In this article we propose a decision process integrating Non-Monotonic Reasoning (NMR), embedded in a deliberative architecture. The NMR process uses *Default Logic* to implement *goal reasoning*, managing partially observable or incomplete information, allowing the design of default behaviours completed by the handling of specific situations, in order to manage the current mission objective as well as safety rules. We illustrate our approach through an application of an underwater robot performing a marine biology mission.

1 Introduction

Over the past decades, autonomous robots have been used in environments where risk is high or access is difficult for humans. However, there is still work to be done when these robots have to integrate automated reasoning: autonomous robots will face unforeseen events (changes in the environment, uncertainty information, failures) and will then have to adapt their behaviour. More specifically, we are interested in giving to these robots *goal reasoning*. Goal reasoning will make the robot able to decide what should be its current objective according to the current situation, in order to ensure both the mission aims and safety constraints. Goal reasoning is a must-have for long-term autonomy in robotics (Ingrand and Ghallab 2017).

In this paper[1], we are more specifically interested in a marine biology application in which an autonomous underwater vehicle must film fishes along specific *transects* (Thanopoulou et al. 2018) defined by marine biologists. To perform a correct transect (w.r.t. the outcomes expected by the biologists), the robot must follow a straight line enforcing some specific constraints (Hereau et al. 2020). In this paper, we are not interested in the trajectory control of the robot, but rather on the *goal reasoning* process. Depending on the current situation, the goal reasoning process can decide either to fulfil mission objectives, i.e. transects defined by the biologists, or change the objective to respect the constraints and ensure the safety of the robot (e.g. going back to a home point, or urgently surfacing). Due to the intrinsic uncertainty of the robot environment, it is necessary for the goal reasoning process to manage uncertain information. While some automated planning methods allow to manage uncertainty, for instance using contingent approaches (Hoffmann and Brafman 2005) or probabilistic planning (Delamer, Watanabe, and Chanel 2021), they manage uncertainties related to the achievement of *one* objective (or optimizing *one* utility function), and do not allow to integrate goal reasoning, i.e. the ability to adapt the current objectives according to the situation.

Such goal reasoning capability generally roots in *Knowledge Reasoning* (KR) approaches. The KR approach that addresses incomplete and contradictory information is called *Non-Monotonic Reasoning* (NMR), in which the reasoning process can make some assumptions, and revise the conclusions according to further observations. In this paper, we are more specifically interested in *Default logic* (Reiter 1980), a non-monotonic logic in which we can *reason by default*, i.e. we can derive consequences only because of lack of evidence of the contrary. Such reasoning is indeed very relevant and flexible to handle autonomous robotic situations, in which we want to specify some *default* behaviours, except when observing specific situations where a specific reasoning should be applied, such as applying safety rules, or changing the current mission goal to adapt to environment changes.

In this paper, we propose a decision process for autonomous systems that uses NMR when incomplete or possibly contradictory information must be considered. The NMR process implements *goal reasoning*, determining which goals are relevant according to the observed situation. Then an automated planning algorithm computes an action plan to achieve this goal. Finally, the NMR process decides what action the robot should actually perform: either the first action of this plan, or another action imposed by a specific rule (typically a safety rule). In Sec. 2, we describe related works. Next, we remind the basic concepts of Default Logic. We then present our contribution in Sec. 4. We show some results in Sec. 5, and finally conclude.

2 Related Works

In Robotics, automated reasoning with logical or formal models is often based on techniques like model-checking or temporal logic. However the purpose of model-checking

[1] **This article has been and published in the IEEE Robotics and Automation Letters (RA-L) and accepted to be presented at the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021).**

is generally to verify properties on a discrete-event system model (Bride et al. 2021; Bensalem, Havelund, and Orlandini 2014). Other works used Linear Temporal Logic for under-actuated robots planning (Shoukry et al. 2017; Bolotov, Grigoriev, and Shangin 2007), describing behaviours of motion in a first-order language and using a theorem solver to obtain moves. Nevertheless, these approaches fail to capture the *non-monotonic* properties.

Knowledge Representation and Reasoning (KRR) has proposed languages to reason about actions and changes, which have been the basis of formal theory of actions (Gelfond and Lifschitz 1998) and dynamic modelling (Levesque et al. 1997; Jin and Thielscher 2004). More recently, a noticeable effort has been made in leveraging KRR processes for robotic applications, more specifically for human-robot interactions, in the KnowRob framework (Tenorth and Beetz 2013), where queries are done to a knowledge base to deduce information about the environment or the robot tasks. However, they still rely on a monotonic reasoning and do not include uncertainty or incompleteness at the logic level.

Robotics applications that integrate NMR generally use Answer Set Programming (ASP) (Kern, Kreijger, and Willcocks 2002). ASP is a declarative language based on a stable model paradigm. Among other applications, ASP has been used in robotics for human-robot collaboration through dialog to handle underspecification and further support knowledge accumulation (Chen et al. 2010), or for planning with time-bounded generation of actions (Schäpers et al. 2018). However, ASP generally has difficulties to reason on all classes of stable models related to a program (e.g. to make a choice of a model according to a user-defined criteria or to compare across the models), and as far as we know, no methodology based on ASP has been proposed for goal reasoning in robotics.

However, there are works that strive to solve problems through NMR based on *default reasoning*, e.g., for decision support in naval missions (Toulgoat, Siegel, and Doncescu 2011), and UAV control (Medina et al. 2018). Default logic is indeed a relevant reasoning framework for robotic missions in which we must handle safety rules and unknown environments. However, the latter works use default logic at the control level, without integration of long-term reasoning, nor providing a methodology for applying default logic.

In this context we propose a decision architecture that intensively uses *default reasoning*, to manage partial and/or contradictory observations and safety rules. We claim that default logic is an appropriate formal tool to design the *goal reasoning* that must take place in an autonomous robot, as it allows to define a default behaviour, and then to specialize this default behaviour by specific rules depending on the encountered (partially observed or incomplete) situation. This reasoning is done at a high-level of abstractions.

3 Default Logic

Default logic is one of the best known formalization for commonsense reasoning, introduced by Reiter (Reiter 1980). This kind of formalization allows to infer arguments based on partial and/or contradictory information as premises. A *default theory* Δ is a pair (D, W), where D is a set of defaults and W a set of formulas in First-Order Logic (FOL). A default $d \in D$ is defined by a quadruplet $X, A(X), B(X), C(X)$, with $X = (x_1, \ldots, x_n)$ a vector of (non-quantified) free variables, and $A(X), B(X), C(X)$ well-formed formulas (wffs) over X, and is represented as:

$$d = \frac{A(X) : B(X)}{C(X)} \tag{1}$$

$A(X)$ are the *prerequisites*, $B(X)$ the *justifications*, $C(X)$ the *consequences*. Intuitively a default means: "if $A(X)$ is true, and there is no evidence that $B(X)$ might be false, then $C(X)$ can be true".

The possible situations that can be derived from a default theory Δ are called *extensions*. An extension E^{Δ} can be seen as a set of believes of acceptable alternatives according to a theory Δ. Formally, an extension E^{Δ} is defined as a smallest fixed-point set for which the following property holds: "if d is a default of D, whose the prerequisite is in E^{Δ}, and the negation of its justification is not in E^{Δ}, then the consequence of d is in E^{Δ}", and defined as $E^{\Delta} = \bigcup_{i=0}^{\infty} E_i$ with (Reiter 1980):

$$E_0 = W \tag{2}$$

$$\forall i > 0,\ E_{i+1} = Th(E_i) \cup \Big\{C(X) \mid \frac{A(X) : B(X)}{C(X)} \in D, \\ A(X) \in E_i, \neg B(X) \notin E^{\Delta}\Big\} \tag{3}$$

where $Th(E_i)$ is the set of *closed wffs* (i.e. with no free variables) that are provable from E_i. However, extensions are difficult to compute in practice since condition $\neg B \notin E^{\Delta}$ (3) assumes that E^{Δ} is known, while E^{Δ} is not yet computed.

Normal default theories (Reiter 1980) is a specific class of default theories in which all defaults have the form $\frac{A(X) : C(X)}{C(X)}$, that can be read "if $A(X)$ is true, and there is no evidence that $C(X)$ might be false, then $C(X)$ can be true". The consequence of this formulation is that (3) can be rewritten (Reiter 1980):

$$\forall i > 0,\ E_{i+1} = Th(E_i) \cup \Big\{C(X) \mid \frac{A(X) : C(X)}{C(X)} \in D, \\ A(X) \in E_i,\ \neg C(X) \notin E_i\Big\} \tag{4}$$

Normal default theories have two main advantages: (1) at least one extension is always guaranteed to exist, and (2) computation of extensions using Horn clauses has a quasi-linear complexity (Marek, Nerode, and Remmel 1997; Dantsin et al. 2001).

4 Decision-Making with Default Logic

In order to handle incomplete or contradictory information in the goal reasoning process of an autonomous robot, we have proposed the decision architecture depicted in Fig. 1. This architecture is based on a ***perception - reasoning - action*** scheme, focusing here on the reasoning part.

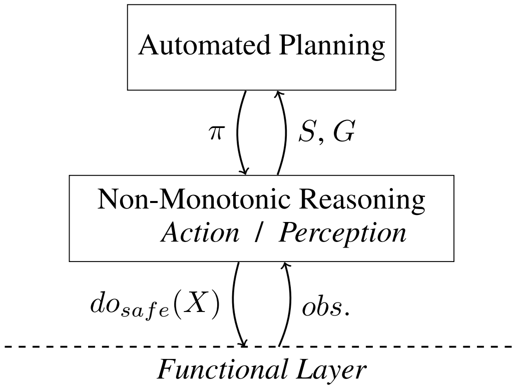


Figure 1: Decision layer based on NMR.

The NMR is made on some observations (*obs*) coming from the robot functional layer, being values of the internal states of the system, environment sensing, failures, etc. From these observations, we build and evaluate a default theory $\Delta = (D, W)$, where formulas in D have the form $\dfrac{A(X) : C(X)}{C(X)}$ and formulas in W have the form $A(X) \rightarrow C(X)$. Then an extension of Δ is computed, and two situations may occur:

- the extension contains a do_{safe} statement, which indicates that an action must be executed immediately, as a reactive response to the observations; in that case, the automated planning part is not called and the action is chosen to be executed;
- the extension does not contain a do_{safe} statement, but produces a current situation estimation S and a goal[2] G, which are given to the automated planning module. Then this module provides a plan π which indicates the next action(s) to be executed.

Once an action has been chosen to be executed, we evaluate again this action in Δ. The extension resulting from Δ must then define a do_{safe} statement, that corresponds to the action that will really be executed on the robot. In the following, we first briefly describe how we abstracted the functional layer (observations and actions) through a *skill* formal model, then we describe the design process of the default theory we used in the architecture, with examples from our marine biology application.

4.1 Functional Layer Abstraction

In order to formalize the interactions with the robot functional layer, we adopted a representation of the robot capabilities and features through a formal *skill model* (Lesire, Doose, and Grand 2020). The skills represent the actions in our reasoning model, that can be triggered through the do_{safe} predicate. These skill models have an executable semantics, described by required inputs, behaviours, expected outcomes, preconditions, etc. In addition to skills, the model also provides two complementary elements: *resources*, modeled as finite state-machine, and *data*. The skills toolchain includes code generators that provide: (1) a library to interface with the functional layer, in order to retrieve the resource and data values (used as observations for the NMR), and to trigger skill (i.e. action) execution, and (2) a PDDL formalization of skills used by the automated planning algorithm.

4.2 Non-Monotonic Reasoning Model

The first step of the deliberative scheme we propose consist in evaluating observations with respect to the default theory Δ. It has a relevant role because it allows to deduce conclusions from observable (functional layer data and resource states) and/or non-observable information, whose model is defined using defaults. Moreover, in addition to estimating non-observable information, Δ can adapt the mission objective to the current observed situation, either defining a new *goal* for the automated planning process, or directly performing an *emergency* action.

In this paper, we propose some *design patterns*, that can be seen as modeling guidelines to define such a default theory Δ for goal reasoning and safety management of a robotic system. We illustrate these guidelines through examples of logical rules for our marine application.

In order to structure Δ in a design perspective, we breakdown Δ in subsets which are specific to the type of information they deal with. First, Δ must contain a set of facts, summarized in a set W_{obs} (that can come either from *observations* of the functional layer, or be static information about the environment). Δ can also contain formulas that correspond to rules relative to three other classes: *state estimation* in Δ_{est} (that allows to infer values of unobserved states), *goal management* in Δ_{goal} and emergency *safety* rules in Δ_{safety}. Each of these sets is composed of a subset of defaults and a subset of FOL formulas. Thus we have:

$$D = \quad D_{est} \cup D_{goal} \cup D_{safety} \qquad (5)$$

$$W = \quad W_{obs} \cup W_{est} \cup W_{goal} \cup W_{safety} \qquad (6)$$

Modeling guidelines for each of these sets are given in the following paragraphs.

Observed Facts The propositions that are used in W_{obs} are directly deduced from the skill-based model of the functional layer, that correspond to the data that can be read from this layer, resource states, and skill execution statuses.

Let's look at the model we developed for our marine robot application. W_{obs} contains observed propositions, such as the robot position through the at predicate, the state of sensors, and the status of internal information, such as the precision level of the robot localization. For instance, a typical initial situation in our mission is when our robot is on the surface: so the GPS sensor could be captured, while the USBL (acoustic) sensor could not be. This situation is modeled by the formula:

$$\begin{aligned} &at(home) \wedge \neg usbl\_captured \wedge \\ &\quad gps\_captured \wedge on\_surface \end{aligned} \qquad (7)$$

[2] When several goals are computed in an extension, then a single one is selected thanks to specific rules.

Predicate Estimation $\Delta_{est} = (D_{est}, W_{est})$ is the part of the default theory that models formulas to infer the truth value of some *hidden* state variables, i.e. that are not directly observed from the functional layer.

Designing formulas in Δ_{est} is very specific to the application. The only guideline that we can enforce is to restrain the C part of the formulas (i.e., the consequences) to only rely on estimated propositions, and never on elements of W_{obs}.

In our application, this typically corresponds to the *localized* predicate, that models the fact that we consider the robot well localized enough to perform a specific task. We want to model the following informal behaviour:

1. we can generally assume that the robot is localized enough to navigate;
2. however, if the USBL signal has not been captured, and neither was the GPS signal, then the robot has never been geo-referenced and we consider that the localization is not good enough.

It results in the following model, where statement 1) is modelled as a default $d_{loc} \in D_{est}$ (8), while statement 2) is an exception to d_{loc}, modelled as classical logical formula $\varphi_{loc} \in W_{est}$ (9).

$$d_{loc} = \frac{\top : localized}{localized} \tag{8}$$

$$\varphi_{loc} = \neg usbl\_was\_captured \wedge \neg gps\_was\_captured \rightarrow \neg localized \tag{9}$$

Safety rules Emergency rules consists in situations where we want to directly apply an action instead of relying on an automated planning process. To design such safety rules of our model, we have defined a specific predicate, $do_{safe}(a)$, where a is a possible action of the functional layer. This predicate represents the fact that action a has to be performed as a consequence to the current situation.

In our application, safety rules correspond for example to situations when we observe a critical failure due to safety sensors of the robot (e.g., internal temperature or pressure, water ingress). In that case, we have either to shut down the robot to ensure its integrity, or to immediately surface.

Formulas in Δ_{safety} must then comply with the following guidelines: safety defaults pattern (D_{safety}) are given in eq. (10); FOL formulas patterns (W_{safety}) are given in eq. (11).

$$\frac{A(X) : do_{safe}(a)}{do_{safe}(a)} \tag{10}$$

$$A(X) \rightarrow do_{safe}(a) \text{ or } A(X) \rightarrow \neg do_{safe}(a) \tag{11}$$

where $A(X)$ is a formula over observed or estimated predicates (from W_{obs} and Δ_{est}), and a is an action. Note that formulas in W_{safety} can "cancel" the application of a safety action (when it is negated) as such formulas may correspond to exception to a default safety rule. In our robotic application, the model that corresponds to the loss of a safety sensor is:

$$\frac{safety\_sensor\_failure(X) : do_{safe}(shut\_down())}{do_{safe}(shut\_down())} \tag{12}$$

Note that the possibility to model *defaults* saves us from listing all the safety sensors, leading to a smaller number of rules. As an exception to this, we can model two formulas (one negative and one positive) that express that, for a specific sensor, instead of shutdown, we want to surface.

Goal Reasoning The formulas in Δ_{goal} aim at deducing the current mission objective. To express these formulas, we have introduced a new predicate, $goal(X)$, where X is a formula over the observable propositions, that correspond to elements of the functional layer (e.g., the robot position, the achievement of an action, the state of a sensor). Proposition $goal(X)$ then means that we want X to be achieved, i.e. to be the current mission objective.

Formulas in Δ_{goal} must then comply with the following guidelines: goal reasoning defaults patterns (D_{goal}) are given in eq. (13); FOL formulas patterns (W_{goal}) are given in eq. (14).

$$\frac{A(X) : goal(Y)}{goal(Y)} \text{ or } \frac{A(X) : \neg goal(Y)}{\neg goal(Y)} \tag{13}$$

$$A(X) \rightarrow goal(Y) \text{ or } A(X) \rightarrow \neg goal(Y) \tag{14}$$

where $A(X)$ is a formula over observable or estimated predicates,and Y a formula on observable predicates only. In our application, a goal reasoning can be informally described as: (1) the general mission objective is to perform a transect between p_A and p_B; (2) except if the localisation is too bad to do a transect. This behaviour is modeled through a default and an exception to this default:

$$\frac{\top : goal(transect\_done(p_A, p_B))}{goal(transect\_done(p_A, p_B))} \tag{15}$$

$$\neg localized \rightarrow \neg goal(transect\_done(X, Y)) \tag{16}$$

4.3 Non-Monotonic Reasoning Process

In the previous paragraphs, we have seen how to model the Default Theory Δ to integrate goal reasoning and safety management. In this paragraph, we will discuss the execution loop of the NMR process. In our application, we have implemented a periodic loop, where, at each period, we apply the steps described below. The decision architecture can then be seen as a *continuous planning* architecture.

Observation First, we get observations from the functional layer, and fill W_{obs} with the observed predicates.

Computing extensions We compute the set of extensions E^{Δ} corresponding to the current default theory.

Applying safety actions If the computed extension has any $do_{safe}(a)$ predicate, then we directly execute the action a (by triggering the corresponding skill in the functional layer), and wait for the next period.

Automated planning If there is no safety action in E^{Δ}, we split E_{Δ} in two sets: the set of goals $G = \{X, s.t.\ goal(X) \in E^{\Delta}\}$, and the set of states $S = E^{\Delta} - G$. We then use an automated planning algorithm to compute the plan π that leads to G from S. In our architecture, we have used the well-known FF algorithm (Hoffmann and Nebel 2001). Note that if G is empty, there is no current goal, and the mission is then finished.

Verifying the planned action Before executing the first action of plan π, we want to ensure that this action is not contradictory with the safety behaviours modeled in the NMR. We then add the first action a_0 of π to Δ, and compute a new extension E_{π}^{Δ}. This step is modeled through a specific predicate, $next\_action(a_0)$, and a default rule:

$$\frac{next\_action(a_0) : do_{safe}(a_0)}{do_{safe}(a_0)} \tag{17}$$

This equation models the fact that, if nothing prevents to execute a_0, then we can execute it. It is then possible to add in the knowledge database an exception to this default. For instance, in our application, we defined the formula:

$$\begin{aligned} next\_action(X) \wedge \neg enough\_energy(X) \\ \rightarrow do_{safe}(go\_boat) \end{aligned} \tag{18}$$

meaning that if the energy is not sufficient to perform the next planned action X, we decide to go back to the boat and abort the mission. E_{π}^{Δ} must then contain a do_{safe} predicate, indicating the action to execute, which will be most of the time the planned action to reach the current goal, except if a safety rule imposed an alternative action.

5 Results

We have implemented the proposed architecture using the ROS2 middleware, with the skill management layer generated from (Lesire, Doose, and Grand 2020), and a specific ROS2 node implementing the decision process in Python/Prolog. The skill model defines 3 data and 9 resources, leading to 12 observable state variables, and 10 skills/actions. Most of the behaviours we have modeled in the default theory have been defined based on a fault analysis of our robot (Hereau et al. 2021). We have also integrated several goal reasoning complementary behaviours. In the end, our default theory consists of 44 rules, including 17 defaults and 27 exceptions as FOL formulas.

To evaluate our approach, we made a set of simulations, activating the several goal reasoning and safety rules. In this section, we first present a simulation run in order to illustrate the approach, and then report an evaluation of computation times when the size of the models increases.

5.1 Simulated Scenario

Figure 2 shows the skills executions (from the functional layer perspective) during our simulated scenario. In this scenario, the robot must perform two transects. The successive actions performed by the robot are to move to a first location, then to dive, activate the video camera, and perform the first transect. Then, the robot moves to the start point of the second transect. During the second transect, the localization precision drops under a threshold, leading to the estimation of proposition $\neg localized$. As a consequence, the transect is cancelled, and the robot goes directly to its home point. In this simulation, the decision architecture ran at a period of 1 second. Figure 3 shows the computation times of the NMR process (to compute extensions) and the FF planning algorithm.

The computation times are quite stable all along the mission, and have quite low values, the total computation time being below 0.1 second. Note that the FF time is the time measured when calling FF as an external process, i.e. including file parsing.

5.2 Computation Time Evaluation

In the previous scenario, the mission objectives defined by the biologists included 2 transects, and the number of possible positions of the robot was restrained to the transects start and end points, as well as the home point and the boat position. While this situation correspond to an actual experiment specified by the marine biologists using our robot, the complexity of the problem is limited. In this section, we evaluated the number of inferences and the computation times when we increase the size of the problem (Fig. 4).

Figure 4a shows the evolution of the computation times when the number of possible positions increases up to 32 positions, which correspond to large problem: in the proposed architecture, we are not interested in trajectory planning, but only on goal reasoning, and the model must then only involve the positions that may have an impact on the mission objectives. We can see that the computation time of FF grows, but the absolute values stay reasonable. The computation time of the NMR is constant, which is expected as there is still only two transects to perform, and the NMR model only relies on the positions attached to the mission objectives. The evolution of the number of inferences (not shown due to lack of place) confirms the constant behaviour of the NMR.

Figure 4b shows the evolution of the number of logical inferences done during the computation of extensions, with respect to the number of mission objectives. In this setup, we fixed the number of positions to 16, and defined from 2 to 10 transects. We can notice that the number of inferences grows linearly, which is consistent with the theoretical complexity of Normal Default Theory (Reiter 1980; Marek, Nerode, and Remmel 1997). Figure 4c shows the evolution of the computation times. Even if the number of goals increases, we can notice that the computations times are almost static whatever the number of transects.

6 Conclusion

In this paper, we have proposed a new decision architecture based on a *non-monotonic reasoning*, more particularly *default reasoning*, that encompasses goal reasoning and safety management, two major features in long-term autonomy of

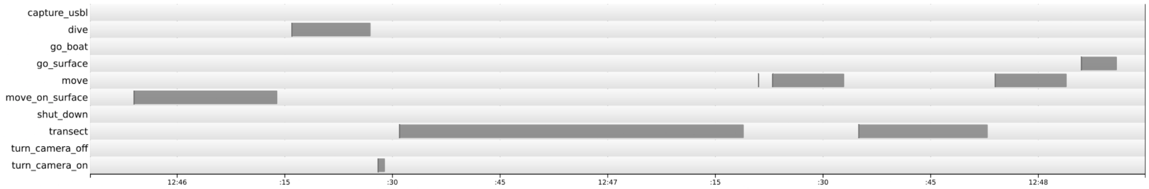


Figure 2: Timeline of the skill execution. Blue segments indicate successful skill executions, and gray segment shows a skill interruption.

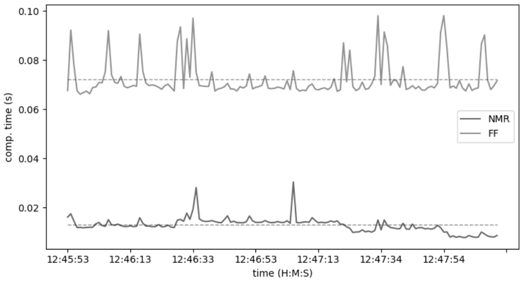


Figure 3: Evolution of the computation time of the NMR and Planning (FF) processes. The dashed lines indicate their respective mean values.

robotic systems. We presented the concept of the architecture, along with guidelines to model the several behaviours in default logic, relying on specific predicates to manage goals and emergency actions. The main decision-making process first gathers observations from the functional layer, then evaluates the default theory to compute an *extension*. This extension may include a do_{safe} statement, with an action to execute immediately, or a goal state to achieve. In the latter case, we use the FF algorithm to compute a plan of actions, and check the consistency of the first action w.r.t. to the default theory. We have illustrated the approach on a marine biology mission, and presented the results of simulations. This application and the associated results clearly show that the proposed method is a practicable approach to manage safety rules and goal reasoning for autonomous robots. Default logic is indeed very convenient and concise framework to model such behaviours, as it allows to define general defaults rules, and then only specify specific exceptions.

Based on this architecture, future work will address the implementation of an interactive decision process, to allow the biologists to modify the NMR rules *online* while the robot is doing a mission, in order to integrate new behaviours due to not modeled situations.

Acknowledgments

This work has been partly funded by the I-Site MUSE of the Univ. of Montpellier through ANR (the French National Research Agency) under the "Investissements d'avenir" program (PIA) with reference ANR-16-IDEX-0006.

References

Bensalem, S.; Havelund, K.; and Orlandini, A. 2014. Verification and validation meet planning and scheduling. *Int. J. on Software Tools for Technology Transfer* 16(1):1–12.

Bolotov, A.; Grigoriev, O.; and Shangin, V. 2007. Automated natural deduction for propositional linear-time temporal logic. In *TIME*.

Bride, H.; Dong, J. S.; Green, R.; Hóu, Z.; Mahony, B. P.; and Oxenham, M. 2021. GRAVITAS: A model checking based planning and goal reasoning framework for autonomous systems. *Eng. Appl. Artif. Intell.* 97:104091.

Chen, X.; Ji, J.; Jiang, J.; Jin, G.; Wang, F.; and Xie, J. 2010. Developing high-level cognitive functions for service robots. In *AAMAS*.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys (CSUR)* 33(3):374–425.

Delamer, J.-A.; Watanabe, Y.; and Chanel, C. P. C. 2021. Safe path planning for UAV urban operation under GNSS signal occlusion risk. *Robotics and Autonomous Systems* 142.

Gelfond, M., and Lifschitz, V. 1998. Action Languages. *Electronic Trans. on Artificial Intelligence* 2:193–210.

Hereau, A.; Godary-Dejean, K.; Guiochet, J.; Robert, C.; Claverie, T.; and Crestani, D. 2020. Testing an Underwater Robot Executing Transect Missions in Mayotte. In *TAROS*.

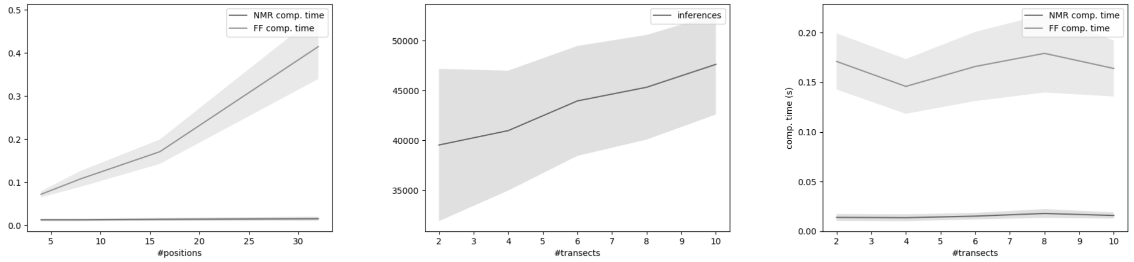
Hereau, A.; Godary-Dejean, K.; Guiochet, J.; ; and Crestani, D. 2021. A Fault Tolerant Control Architecture Based on Fault Trees for an Underwater Robot Executing Transect Missions. In *ICRA*.

Hoffmann, J., and Brafman, R. 2005. Contingent Planning via Heuristic Forward Search with Implicit Belief States. In *ICAPS*.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR* 14:253–302.

Ingrand, F., and Ghallab, M. 2017. Deliberation for autonomous robots: A survey. *Artificial Intelligence* 247:10–44.

Jin, Y., and Thielscher, M. 2004. Representing beliefs in the fluent calculus. In *ECAI*.



(a) Comp. times w.r.t. environment size

(b) Inferences w.r.t. number of goals

(c) Comp. times w.r.t. number of goals

Figure 4: Evaluation of the NMR architecture processes w.r.t. the size of the model. Plain curves represent the average value. Light areas indicate the standard deviation envelope.

Kern, T.; Kreijger, J.; and Willcocks, L. 2002. Exploring ASP as sourcing strategy: theoretical perspectives, propositions for practice. *J. of Strategic Information Systems* 11(2):153–177.

Lesire, C.; Doose, D.; and Grand, C. 2020. Formalization of robot skills with descriptive and operational models. In *IROS*.

Levesque, H. J.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. GOLOG: A logic programming language for dynamic domains. *The Journal of Logic Programming* 31(1-3):59–83.

Marek, V. W.; Nerode, A.; and Remmel, J. B. 1997. Complexity of recursive normal default logic. *Fundamenta Informaticae* 32(2):139–147.

Medina, J. L. V.; Siegel, P.; Risch, V.; and Doncescu, A. 2018. Intelligent and Adaptive System based on a Non-monotonic Logic for an Autonomous Motor-glider. In *ICARCV*.

Reiter, R. 1980. A logic for default reasoning. *Artificial intelligence* 13(1-2):81–132.

Schäpers, B.; Niemueller, T.; Lakemeyer, G.; Gebser, M.; and Schaub, T. 2018. Asp-based time-bounded planning for logistics robots. In *ICAPS*.

Shoukry, Y.; Nuzzo, P.; Balkan, A.; Saha, I.; Sangiovanni-Vincentelli, A. L.; Seshia, S. A.; Pappas, G. J.; and Tabuada, P. 2017. Linear temporal logic motion planning for teams of underactuated robots using satisfiability modulo convex programming. In *CDC*.

Tenorth, M., and Beetz, M. 2013. KnowRob: A knowledge processing infrastructure for cognition-enabled robots. *IJRR* 32(5):566–590.

Thanopoulou, Z.; Sini, M.; Vatikiotis, K.; Katsoupis, C.; Dimitrakopoulos, P. G.; and Katsanevakis, S. 2018. How many fish? Comparison of two underwater visual sampling methods for monitoring fish communities. *PeerJ* 6:e5066.

Toulgoat, I.; Siegel, P.; and Doncescu, A. 2011. Modelling of submarine navigation by nonmonotonic logic. In *Int. Conf. on Broadband and Wireless Computing, Communication and Applications*.

KLM-Style Defeasibility for Restricted First-Order Logic

Giovanni Casini[1,2], **Thomas Meyer**[1,2], **Guy Paterson-Jones**[2]
[1] ISTI–CNR, Italy
[2] University of Cape Town and CAIR, South Africa
giovanni.casini@isti.cnr.it, tmeyer@cs.uct.ac.za, PTRGUY002@myuct.ac.za

Abstract

We extend the KLM approach to defeasible reasoning to be applicable to a restricted version of first-order logic. We describe defeasibility for this logic using a set of rationality postulates, provide an appropriate semantics for it, and present a representation result that characterises the semantic description of defeasibility in terms of the rationality postulates. Based on this theoretical core, we then propose a version of defeasible entailment that is inspired by Rational Closure as it is defined for defeasible propositional logic and defeasible description logics. We show that this form of defeasible entailment is *rational* in the sense that it adheres to our rationality postulates. The work in this paper is the first step towards our ultimate goal of introducing KLM-style defeasible reasoning into the family of Datalog+/- ontology languages.

1 Introduction

The past 15 years have seen a flurry of activity to introduce defeasible-reasoning capabilities into languages that are more expressive than propositional logic (Casini and Straccia 2010, 2013; Casini et al. 2015; Giordano et al. 2013, 2015; Bonatti et al. 2015; Bonatti 2019; Pensel and Turhan 2018). Most of the focus has been on defeasibility for description logics (DLs), with much of it devoted to versions of the so-called KLM approach to defeasible reasoning initially advocated for propositional logic by Kraus, Lehmann, and Magidor (1990), and Lehmann and Magidor (1992). In DLs, knowledge is expressed as general concept inclusions of the form $C \sqsubseteq D$, where C and D are concepts, with the intended meaning that every instance of C is also an instance of D. Defeasible DLs allow, in addition, for defeasible concept inclusions of the form $C \mathrel{\underset{\sim}{\sqsubset}} D$ with the intended meaning that instances of C are *usually* instances of D. For instance, $\mathsf{Student} \mathrel{\underset{\sim}{\sqsubset}} \neg\exists\mathsf{pays.Tax}$ (students usually don't pay tax) is a defeasible version of $\mathsf{Student} \sqsubseteq \neg\exists\mathsf{pays.Tax}$ (students don't pay tax).

Given the tight formal relationship between DLs and the family of Datalog+/- ontology languages (Calì et al. 2010; Calì, Gottlob, and Lukasiewicz 2012), it is surprising that this form of defeasibility has not yet found its way into Datalog+/-. In this paper we take the first steps to fill that gap by providing the theoretical foundations for defeasibility in a restricted version of first-order logic. We refer to the classical version of the logic as RFOL and the defeasible extension as DRFOL. It suffices to to use Herbrand interpretations for the semantics of RFOL. However, the availability of non-unary predicates means that the definition of an appropriate semantics for DRFOL is a non-trivial exercise. This is because the intuition underlying KLM-style defeasibility generally depends on the type of language in which it is implemented. For propositional logics the intuition dictates a notion of typicality over possible worlds. The statement "birds usually fly", formalised as $\mathsf{bird} \mid\!\sim \mathsf{fly}$, is intended to convey that in the most typical worlds in which bird is true, fly is also true. In contrast, defeasibility in DLs invokes a form of typicality over individuals. The statement $\mathsf{Student} \mathrel{\underset{\sim}{\sqsubset}} \neg\exists\mathsf{pays.Tax}$ states that of all those individuals that are students, the most typical ones don't pay taxes. Consider, for instance, the following example of (Delgrande 1998):

Example 1. The following DRFOL knowledge base states that humans don't feed wild animals, that elephants are usually wild animals, that keepers are usually human, and that keepers usually feed elephants, but that Fred the keeper usually does not feed elephants (the connective $\rightsquigarrow$ refers to defeasible implication and variables are implicitly quantified).

$$\begin{aligned}\mathcal{K} = \{\ & \mathsf{wild\_animal(x)} \land \mathsf{human(y)} \rightarrow \neg\mathsf{feeds(y,x)},\\ & \mathsf{elephant(x)} \rightsquigarrow \mathsf{wild\_animal(x)},\\ & \mathsf{keeper(x)} \rightsquigarrow \mathsf{human(x)},\\ & \mathsf{elephant(x)} \land \mathsf{keeper(y)} \rightsquigarrow \mathsf{feeds(y,x)},\\ & \mathsf{elephant(x)} \land \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{feeds(fred,x)}\ \}\end{aligned}$$

Note that all statements, except for the first one, are defeasible. For any appropriate semantics, the knowledge base in the example should be satisfiable (given an appropriate notion of satisfiability). With this in mind it soon becomes clear that the propositional approach cannot achieve this. To see why, note that applying the propositional intuition to the example would result in $\mathsf{elephant(x)} \land \mathsf{keeper(y)} \rightsquigarrow \mathsf{feeds(y,x)}$ meaning that in the most typical worlds (Herbrand interpretations in this case) all keepers feed all elephants. This is in conflict with $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{feeds(fred,x)}$, which states that in the most typical Herbrand interpretations, keeper Fred does not feed any elephants. For any reasonable definition of satisfiability, this would render the knowledge base unsatisfiable.

The DL-based intuition of object typicality is also problematic. Under this intuition the statement $\mathsf{elephant(x)} \rightsquigarrow \mathsf{wild\_animal(x)}$ would mean that the most typical elephants are wild animals. Similarly, $\mathsf{keeper(x)} \rightsquigarrow \mathsf{human(x)}$ would mean that the most typical keepers are human. Combined with the first statement in the knowledge base, it would then follow that the most typical keepers (being humans) do not feed the most typical elephants (being wild animals). On the other hand, the fourth statement in the knowledge base explicitly states that the most typical keepers feed the most typical elephants, from which we obtain the counter-intuitive conclusion that typical elephants and typical keepers cannot exist simultaneously.

We resolve this matter with a semantics that is in line with the propositional intuition of a typicality ordering over worlds, but also includes aspects of the DL intuition of the typicality of individuals. We achieve the latter by enriching our semantics with a set of *typicality objects*, the elements of which are used to represent *typical* individuals. Thus, $\mathsf{elephant(x)} \wedge \mathsf{keeper(y)} \rightsquigarrow \mathsf{feeds(y,x)}$ means that in the most typical enriched Herbrand interpretations, all typical keepers feed all typical elephants, with the understanding that there may be exceptional keepers that don't feed some elephants. Note that the term *typical* is used here in two different, but related, ways.

The central theoretical result of the paper is a representation result (Theorems 2 and 3), showing that defeasible implication defined in this way can be characterised w.r.t. a set of KLM-style rationality postulates adapted for DRFOL. We show that DRFOL formally generalises propositional KLM-style defeasible reasoning in two ways. The cases where DRFOL, restricted to 0-ary predicates, or where n-ary predicates for any $n > 0$ are allowed, but with a restriction to variable-free statements, both reduce to propositional KLM-style defeasibility. A comparison with defeasible DLs is more complicated, but the semantics of defeasible DLs, for the most part, carries over to DRFOL. An important exception is that whereas a defeasible DL statement of the form $A \mathrel{\underset{\sim}{\sqsubset}} \bot$ is equivalent to its classical counterpart $A \sqsubseteq \bot$, it is possible to distinguish between the DRFOL version of the same statement, $A(x) \rightsquigarrow \bot$, and its classical counterpart $A(x) \rightarrow \bot$. In fact, the former is weaker than the latter.

Another important consequence of our representation result is that it provides the theoretical foundation for the definition of various forms of defeasible entailment for DRFOL. We present one such form of defeasible entailment in Section 5, and show that it can be viewed as the DRFOL analogue of Rational Closure, as originally defined for the propositional case (Kraus, Lehmann, and Magidor 1990; Lehmann and Magidor 1992).

The rest of the paper is structured as follows. Section 2 is a brief introduction to RFOL, as well as to KLM-style defeasible reasoning for propositional logics. Section 4 is the heart of the paper. It introduces DRFOL, describes an abstract notion of satisfaction w.r.t. a set of KLM-style postulates, provides a semantics, and proves a representation result, showing that the KLM-style postulates characterise the semantic construction. Section 5 presents a form of defeasible entailment for DRFOL that can be viewed as the DRFOL equivalent of the well-known propositional form of defeasible entailment known as Rational Closure. Section 6 compares defeasible reasoning in DRFOL with KLM-style defeasible reasoning in propositional logic and DLs. Section 7 provides an overview of related work, while Section 8 concludes and briefly discusses future work. The proofs can be found in an appendix: https://tinyurl.com/7472fn2a.

2 Background

We consider a restricted version of a first-order language, which we refer to as RFOL. The language of RFOL is defined by three disjoint sets of symbols: $\textsc{const}$, a finite set of constants; $\textsc{var}$, a countably infinite set of variable symbols; and $\textsc{pred}$, a finite set of predicate symbols. It has no function symbols. Associated with each predicate symbol $\alpha \in \textsc{pred}$ is an *arity*, denoted $\mathfrak{ar}(\alpha) \in \mathbb{N}$, which represents the number of terms it takes as arguments. We assume the existence of predicate symbols $\top$ and $\bot$, which we take to have arity 0. A *term* is an element of $\textsc{const} \cup \textsc{var}$. An *atom* is an expression of the form $\alpha(t_1, \ldots, t_{\mathfrak{ar}(\alpha)})$ where $\alpha \in \textsc{pred}$ and the t_i are terms. Observe that $\top$ and $\bot$ are atoms as well.

A *compound* is defined to be a boolean combination of atoms, i.e. an expression built out of atoms and the standard logical connectives $\neg$, $\wedge$, and $\vee$. An *implication* is defined to have the form $A(\vec{x}) \rightarrow B(\vec{y})$ where $A(\vec{x})$ and $B(\vec{y})$ are compounds, and where the terms occurring in $\vec{x}$ and $\vec{y}$ may overlap. A compound (respectively, implication) is said to be *ground* if all the terms contained in it are constants; otherwise it is *open*. In RFOL, the only formulas we permit are compounds and implications. When viewed as formulas, compounds and implications are understood to be implicitly universally quantified.

We adopt the following conventions for various kinds of formula. Constant symbols and variables will be written in lowercase English, with early letters used for constants $(a, b, \ldots)$ and later letters used for variables $(x, y, \ldots)$. Compounds will be written in uppercase English $(A, B, \ldots)$. A tuple of variables or constants will be written with overbars, such as $\vec{x}$ and $\vec{a}$ respectively, and $A(\vec{x})$ and $B(\vec{a})$ will be used as shorthand for compounds over their respective tuples of terms. We use lowercase greek $(\alpha, \beta, \ldots)$ to denote RFOL formulas.

We omit specifying the symbol sets under consideration, as they can be inferred from context. The set of all formulas (compounds and implications) is denoted by $\mathcal{L}$, and a *knowledge base* $\mathcal{K}$ is defined to be a finite subset of $\mathcal{L}$.

RFOL can be thought of as an extension of Datalog (Abiteboul, Hull, and Vianu 1995). In fact, we use *Herbrand interpretations* to specify the semantics of RFOL. The Herbrand universe $\mathbb{U}$ is the set of constant symbols $\textsc{const}$. The *Herbrand base* of $\mathbb{U}$, denoted $\mathbb{B}$, is the set of facts defined over $\mathbb{U}$. A *Herbrand interpretation* is a subset $\mathcal{H} \subseteq \mathbb{B}$.

Substitutions are defined to be functions $\varphi : \textsc{var} \rightarrow \textsc{var} \cup \textsc{const}$ assigning a term to each variable symbol. *Variable substitutions* are substitutions that assign only variables, and *ground substitutions* are substitutions that assign only constants. The application of a substitution φ to a compound $A(\vec{x})$ is denoted $A(\varphi(\vec{x}))$. RFOL knowledge bases are interpreted by Herbrand interpretations $\mathcal{H}$ as follows:

1. if $A(\vec{a})$ is a ground atom, then $\mathcal{H} \Vdash A(\vec{a})$ iff $A(\vec{a}) \in \mathcal{H}$.
2. if $A(\vec{a})$ and $B(\vec{b})$ are ground compounds (where $\vec{a}$ and $\vec{b}$ may overlap), then $\mathcal{H} \Vdash A(\vec{a})$ and $\mathcal{H} \Vdash A(\vec{a}) \to B(\vec{b})$ according to the usual laws of boolean connectives.
3. if $A(\vec{x})$ is an open compound, then $\mathcal{H} \Vdash A(\vec{x})$ iff $\mathcal{H} \Vdash A(\varphi(\vec{x}))$ for every ground substitution φ.
4. if $A(\vec{x}) \to B(\vec{y})$ is an open implication (where $\vec{x}$ and $\vec{y}$ may overlap), then $\mathcal{H} \Vdash A(\vec{x}) \to B(\vec{y})$ iff $\mathcal{H} \Vdash A(\varphi(\vec{x})) \to B(\varphi(\vec{y}))$ for every ground substitution φ.
5. If $\mathcal{K}$ is a knowledge base, then $\mathcal{H} \Vdash \mathcal{K}$ iff $\mathcal{H} \Vdash \alpha$ for every $\alpha \in \mathcal{K}$.

The set of Herbrand interpretations is denoted by $\mathscr{H}$. A Herbrand interpretation that satisfies a knowledge base $\mathcal{K}$ is a *Herbrand model* of $\mathcal{K}$.

3 Propositional Defeasible Reasoning

Kraus, Lehmann, and Magidor (1990) originally define $\mid\!\sim$ as a consequence relation over a propositional language, with statements of the form $\alpha \mid\!\sim \beta$ to be interpreted as the meta-statement "β is a defeasible consequence of α". Lehmann and Magidor (1992) subsequently shift to interpreting $\alpha \mid\!\sim \beta$ as the object-level statement "α defeasibly implies β", with $\mid\!\sim$ viewed as an object-level connective. An abstract notion of satisfaction can then be defined in terms of *satisfaction sets*. A satisfaction set $\mathcal{S}$ of statements of the form $\alpha \mid\!\sim \beta$ is said to be *rational* if it satisfies the well-known KLM properties below (Lehmann and Magidor 1992). Lehmann and Magidor did not refer to satisfaction sets, but our formulation here is equivalent to theirs for the propositional case:

$$(\textsc{Refl})\quad \alpha \mid\!\sim \alpha \in \mathcal{S}$$

$$(\textsc{Rw})\quad \frac{\alpha \mid\!\sim \beta \in \mathcal{S},\ \models \beta \to \gamma}{\alpha \mid\!\sim \gamma \in \mathcal{S}}$$

$$(\textsc{Lle})\quad \frac{\models \alpha \leftrightarrow \beta,\ \alpha \mid\!\sim \gamma \in \mathcal{S}}{\beta \mid\!\sim \gamma \in \mathcal{S}}$$

$$(\textsc{And})\quad \frac{\alpha \mid\!\sim \beta \in \mathcal{S},\ \alpha \mid\!\sim \gamma \in \mathcal{S}}{\alpha \mid\!\sim \beta \land \gamma \in \mathcal{S}}$$

$$(\textsc{Or})\quad \frac{\alpha \mid\!\sim \gamma \in \mathcal{S},\ \beta \mid\!\sim \gamma \in \mathcal{S}}{\alpha \lor \beta \mid\!\sim \gamma \in \mathcal{S}}$$

$$(\textsc{Rm})\quad \frac{\alpha \mid\!\sim \beta \in \mathcal{S},\ \alpha \mid\!\sim \neg\gamma \notin \mathcal{S}}{\alpha \land \gamma \mid\!\sim \beta \in \mathcal{S}}$$

A semantics for defeasible implications is provided by *ranked interpretations* $\mathscr{R}$, which are defined to be total preorders over a subset $U_{\mathscr{R}} \subseteq U$ of valuations. Valuations that are lower in the ordering are considered to be more typical, whereas valuations that are not in $U_{\mathscr{R}}$ are impossibly atypical. A defeasible statement $\alpha \mid\!\sim \beta$ is *satisfied in* $\mathscr{R}$ ($\mathscr{R} \Vdash \alpha \mid\!\sim \beta$) iff the $\mathscr{R}$-minimal models of α are also models of β, which formalises the intuition that β holds in the most typical situations in which α is true. A classical statement α is satisfied by $\mathscr{R}$ iff every valuation in $U_{\mathscr{R}}$ satisfies α.

Lehmann and Magidor (1992) prove the following correspondence between rational satisfaction sets and ranked interpretations:

Theorem 1. *(Lehmann and Magidor 1992). A set $\mathcal{S}$ of statements of the form $\alpha \mid\!\sim \beta$ is a rational satisfaction set iff there is a ranked interpretation $\mathscr{R}$ such that $\alpha \mid\!\sim \beta \in \mathcal{S}$ iff $\mathscr{R} \Vdash \alpha \mid\!\sim \beta$.*

To conclude this section, observe that $\mathscr{R} \Vdash \neg\alpha \mid\!\sim \bot$ iff $\mathscr{R}$ contains no models of $\neg\alpha$ (which are therefore viewed as impossible). In other words, $\mathscr{R} \Vdash \neg\alpha \mid\!\sim \bot$ iff $\mathscr{R} \Vdash \alpha$. We return to this property of propositional defeasible reasoning in Section 6.

4 Defeasible Restricted First-Order Logic

Defeasible Restricted First-Order Logic (DRFOL for short) extends the logic RFOL that was presented in Section 2 with *defeasible implications* of the form $A(\vec{x}) \leadsto B(\vec{y})$, where $A(\vec{x})$ and $B(\vec{y})$ are compounds, and where $\vec{x}$ and $\vec{y}$ may overlap. Observe that $\leadsto$ is intended to be the defeasible analogue of classical implication. That is, $A(\vec{x}) \leadsto B(\vec{y})$ is the defeasible analogue of the RFOL formula $A(\vec{x}) \to B(\vec{y})$. The set of defeasible implications is denoted $\mathcal{L}^{\leadsto}$, and a *DRFOL knowledge base* $\mathcal{K}$ is defined to be a subset of $\mathcal{L} \cup \mathcal{L}^{\leadsto}$. Note that DRFOL knowledge bases may include (classical) RFOL formulas.

As demonstrated in Example 1, defeasible implications are intended to model properties that *typically* hold, but which may have exceptions. In this example, for instance, $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \leadsto \neg\mathsf{feeds(x, fred)}$, is an exception to $\mathsf{elephant(x)} \land \mathsf{keeper(y)} \leadsto \mathsf{feeds(x, y)}$. A DRFOL knowledge base containing these statements ought to be satisfiable (for an appropriate notion of satisfaction). The same goes for the DRFOL knowledge base $\{\mathsf{bird(x)} \leadsto \mathsf{fly(x)}, \mathsf{bird(tweety)}, \neg\mathsf{fly(tweety)}\}$. To formalise these intuitions, we describe the intended behaviour of the defeasible connective $\leadsto$, and its interaction with (classical) RFOL formulas, in terms of a set of rationality postulates in the KLM style (Kraus, Lehmann, and Magidor 1990; Lehmann and Magidor 1992). These postulates are expressed via an abstract notion of satisfaction:

Definition 1. A *satisfaction set* is a subset $\mathcal{S} \subseteq \mathcal{L} \cup \mathcal{L}^{\leadsto}$.

We denote the classical part of a satisfaction set by $\mathcal{S}_C = \mathcal{S} \cap \mathcal{L}$. The first postulate we consider ensures that a satisfaction set respects the classical notion of satisfaction when restricted to classical formulas, where $\models$ refers to classical entailment:

$$(\textsc{ClassF})\quad \frac{\mathcal{S}_C \models A(\vec{x})}{A(\vec{x}) \in \mathcal{S}}$$

$$(\textsc{ClassR})\quad \frac{\mathcal{S}_C \models A(\vec{x}) \to B(\vec{y})}{A(\vec{x}) \to B(\vec{y}) \in \mathcal{S}}$$

Next, we consider the interaction between classical and defeasible implications. We expect the following supraclassicality postulate to hold:

$$(\textsc{SupR})\quad \frac{A(\vec{x}) \to B(\vec{y}) \in \mathcal{S}}{A(\vec{x}) \leadsto B(\vec{y}) \in \mathcal{S}}$$

A similar postulate for compounds then holds:

$$(\textsc{SupF})\ \frac{A(\vec{x}) \in \mathcal{S}}{\neg A(\vec{x}) \rightsquigarrow \bot \in \mathcal{S}}$$

Proposition 1. (SupF) *follows from* (ClassR) *and* (SupR).

We now consider the core of the proposal for defining rational satisfaction sets, the KLM rationality postulates, lifted to the DRFOL case, and expressed in terms of satisfaction sets:

$$(\textsc{Refl})\ A(\vec{x}) \rightsquigarrow A(\vec{x}) \in \mathcal{S}$$

$$(\textsc{Rw})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S},\ \models B(\vec{y}) \to C(\vec{z})}{A(\vec{x}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}}$$

$$(\textsc{Lle})\ \frac{A(\vec{x}) \rightsquigarrow C(\vec{z}) \in \mathcal{S},\ \models A(\vec{x}) \to B(\vec{y}),\ \models B(\vec{y}) \to A(\vec{x})}{B(\vec{y}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}}$$

$$(\textsc{And})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S},\ A(\vec{x}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}}{A(\vec{x}) \rightsquigarrow B(\vec{y}) \land C(\vec{z}) \in \mathcal{S}}$$

$$(\textsc{Or})\ \frac{A(\vec{x}) \rightsquigarrow C(\vec{z}) \in \mathcal{S},\ B(\vec{y}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}}{A(\vec{x}) \lor B(\vec{y}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}}$$

$$(\textsc{Rm})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S},\ A(\vec{x}) \rightsquigarrow \neg C(\vec{z}) \notin \mathcal{S}}{A(\vec{x}) \land C(\vec{z}) \rightsquigarrow B(\vec{y}) \in \mathcal{S}}$$

Next we consider *instantiations* of implications. To begin with, note that universal instantiation is *not* a desirable property for defeasible implications:

$$(\textsc{Duir})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S}}{A(\varphi(\vec{x})) \rightsquigarrow B(\varphi(\vec{y})) \in \mathcal{S}}$$

To see why, consider a satisfaction set $\mathcal{S}$ containing $\mathsf{elephant(x)} \land \mathsf{keeper(y)} \rightsquigarrow \mathsf{feeds(y,x)}$ and $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{feeds(y,fred)}$. From (Duir) we have $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \rightsquigarrow \mathsf{feeds(y,fred)} \in \mathcal{S}$, and hence by (And) and (Rw) that $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \rightsquigarrow \bot \in \mathcal{S}$ as well, which is in conflict with the intuition that exceptional cases (all elephants usually not being fed by keeper Fred) should be permitted to exist alongside the general case (all elephants usually being fed by all keepers).

Weaker forms of instantiation for defeasible implications are more reasonable. Consider $\mathsf{keeper(x)} \rightsquigarrow \mathsf{feeds(x,y)}$, which states that keepers typically feed everything. While we cannot conclude anything about instances of x, for the reasons discussed above, we should at least be able to conclude things about instances of y, since y only appears in the consequent of the implication. This motivates the following postulate, where ψ is a variable substitution and $\vec{x} \cap \vec{y} = \emptyset$:

$$(\textsc{Irr})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{x}, \vec{y}) \in \mathcal{S}}{A(\vec{x}) \rightsquigarrow B(\vec{x}, \psi(\vec{y})) \in \mathcal{S}}$$

There are some more subtle forms of defeasible instantiation that seem reasonable as well. Consider the following relation defined over $\mathcal{L}$:

Definition 2. $A(\vec{x})$ is *at least as typical* as $B(\vec{y})$ with respect to $\mathcal{S}$, denoted $A(\vec{x}) \preccurlyeq_{\mathcal{S}} B(\vec{y})$, iff $A(\vec{x}) \lor B(\vec{y}) \rightsquigarrow \neg A(\vec{x}) \notin \mathcal{S}$.

Intuitively, $A(\vec{x}) \preccurlyeq_{\mathcal{S}} B(\vec{y})$ states that typical instances of $A(\vec{x})$ are at least as typical as typical instances of $B(\vec{y})$. Note that $\preccurlyeq_{\mathcal{S}}$ does *not* partially order $\mathcal{L}$ in general, but is rather a partial ordering of the subset of *consistent* formulas of $\mathcal{L}$, i.e. $A(\vec{x}) \in \mathcal{L}$ such that $A(\vec{x}) \rightsquigarrow \bot \notin \mathcal{S}$.

For any variable substitution ψ, a typical instance of $A(\psi(\vec{x}))$ is always an instance of $A(\vec{x})$. Thus we should expect the following postulate to hold, where ψ is any variable substitution:

$$(\textsc{Typ})\ A(\vec{x}) \preccurlyeq_{\mathcal{S}} A(\psi(\vec{x}))$$

The last postulate we consider has to do with defeasibly impossible formulas. Suppose that $A(\varphi(\vec{x})) \rightsquigarrow \bot \in \mathcal{S}$ for all substitutions $\varphi : \textsc{var} \to \textsc{var} \cup \mathbb{U}$. This intuitively states that there are no typical instances of *any* specialisation of $A(\vec{x})$. Thus we should expect that there are in fact no instances of $A(\vec{x})$ at all:

$$(\textsc{Imp})\ \frac{A(\varphi(\vec{x})) \rightsquigarrow \bot \in \mathcal{S} \text{ for all } \varphi : \textsc{var} \to \textsc{var} \cup \mathbb{U}}{\neg A(\vec{x}) \in \mathcal{S}}$$

This puts us in a position to define the central construction of the paper: that of a *rational* satisfaction set.

Definition 3. A satisfaction set $\mathcal{S}$ is *rational* iff it satisfies (ClassF), (ClassR), (SupR), (Irr), (Typ), (Imp) and (Refl)-(Rm).

Note that rational satisfaction sets satisfy the following form of label invariance for defeasible implications, where the variable substitution ψ is a *permutation*:

$$(\textsc{Per})\ \frac{A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S}}{A(\psi(\vec{x})) \rightsquigarrow B(\psi(\vec{y})) \in \mathcal{S}}$$

Proposition 2. (Per) *follows from* (Refl)-(Rm)*,* (Irr) *and* (Typ).

4.1 Semantics

We now proceed to define an appropriate semantics for defeasible implications. The first step is to enrich the Herbrand universe with a set $\mathcal{T}$ of *typicality objects*. Typicality objects represent the individuals that aren't explicitly mentioned in a given knowledge base, and are used to interpret defeasible implications in a ranking of (enriched) Herbrand interpretations.

Definition 4. The *enriched Herbrand universe* is defined to be the set $\mathbb{U}_{\mathcal{T}} = \mathbb{U} \cup \mathcal{T}$. An *enriched Herbrand interpretation* (or EHI) $\mathcal{E}$ is a Herbrand interpretation over the enriched Herbrand universe.

Observe that every enriched Herbrand interpretation $\mathcal{E}$ restricts to a unique Herbrand interpretation $\mathcal{H}^{\mathcal{E}}$ over $\mathbb{U}$, defined by $\mathcal{H}^{\mathcal{E}} = \mathcal{E} \cap \mathbb{B}$. The set of EHIs over $\mathcal{T}$ is denoted by $\mathscr{H}_{\mathcal{T}}$. To interpret defeasible implications, we make use of preference rankings over $\mathscr{H}_{\mathcal{T}}$.

Definition 5. A *ranked interpretation* is a function $rk : \mathscr{H}_{\mathcal{T}} \to \Omega \cup \{\infty\}$, for some linear poset Ω, satisfying the following properties, where we define $\mathscr{H}_{\mathcal{T}}^{rk} = \{\mathcal{E} \in \mathscr{H}_{\mathcal{T}} : rk(\mathcal{E}) \neq \infty\}$ to be the set of possible EHIs w.r.t. *rk*, and $\mathscr{H}_{\mathcal{T}}^{rk}(A(\vec{x})) = \{\mathcal{E} \in \mathscr{H}_{\mathcal{T}}^{rk} : \mathcal{E} \Vdash A(\varphi(\vec{x})) \text{ for some } \varphi : \textsc{var} \to \mathcal{T}\}$ to be the set of possible EHIs w.r.t. *rk* satisfying some typical instance of $A(\vec{x}) \in \mathcal{L}$:

1. if $rk(\mathcal{E}) = x < \infty$, then for every $y \leq x$ there is some $\mathcal{E}' \in \mathscr{H}_{\mathcal{T}}$ such that $rk(\mathcal{E}') = y$.

2. for all $A(\vec{x}) \in \mathcal{L}$, $\mathscr{H}^{rk}_{\mathcal{T}}(A(\vec{x}))$ is either empty or has an element that is an *rk*-minimal model of $A(\vec{x})$. This is *smoothness* (Kraus, Lehmann, and Magidor 1990).

The set of all ranked interpretations over $\mathcal{T}$ is denoted $\mathcal{R}_{\mathcal{T}}$.

Definition 6. For $A(\vec{x}), B(\vec{y}) \in \mathcal{L}$:

1. $rk \Vdash A(\vec{x})$ iff $\mathcal{E} \Vdash A(\vec{x})$ for all $\mathcal{E} \in \mathscr{H}^{rk}_{\mathcal{T}}$.
2. $rk \Vdash A(\vec{x}) \rightarrow B(\vec{y})$ iff $\mathcal{E} \Vdash A(\vec{x}) \rightarrow B(\vec{y})$ for all $\mathcal{E} \in \mathscr{H}^{rk}_{\mathcal{T}}$.
3. $rk \Vdash A(\vec{x}) \rightsquigarrow B(\vec{y})$ iff $\mathcal{E} \Vdash A(\varphi(\vec{x})) \rightarrow B(\varphi(\vec{y}))$ for all $\mathcal{E} \in \min_{rk} \mathscr{H}^{rk}_{\mathcal{T}}(A(\vec{x}))$ and all $\varphi : \text{VAR} \rightarrow \mathcal{T}$.

Thus, compounds and classical implications are true in a ranked interpretation *rk* if they are true in all possible EHIs w.r.t. *rk*, while a defeasible implication is true in *rk* if its classical counterpart, with variables substituted for typicality objects, are true in all minimal EHIs (possible w.r.t. *rk*) in which the antecedent of the defeasible implication is true. A ranked interpretation in which a statement is true is a *ranked model* of the statement.

Example 2. This is an example proposed by Delgrande (1998). The following DRFOL knowledge base states that elephants usually like keepers, that elephants usually *don't* like the keeper Fred, and that the elephant Clyde usually *does* like Fred:

$$\begin{aligned}\mathcal{K} = \{&\mathsf{elephant(x)} \wedge \mathsf{keeper(y)} \rightsquigarrow \mathsf{likes(x,y)},\\ &\mathsf{elephant(x)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{likes(x,fred)},\\ &\mathsf{elephant(clyde)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \mathsf{likes(clyde,fred)}\}.\end{aligned}$$

Let $\mathcal{T} = \{\mathsf{t}_1, \ldots\}$ be the set of typicality objects. For readability we abbreviate $\mathsf{elephant}$ by e, keeper by k and likes by l.

Consider the EHIs $\mathcal{E}_1 = \{\mathsf{e(t_1)}, \mathsf{k(t_2)}, \mathsf{l(t_1,t_2)}, \mathsf{e(t_2)}, \mathsf{e(clyde)}, \mathsf{k(fred)}, \mathsf{l(clyde,fred)}\}$, $\mathcal{E}_2 = \{\mathsf{e(t_1)}, \mathsf{k(t_2)}, \mathsf{l(t_1,t_2)}, \mathsf{k(t_3)}, \mathsf{l(t_1,t_3)}, \mathsf{e(clyde)}, \mathsf{k(fred)}, \mathsf{l(clyde,fred)}\}$, and $\mathcal{E}_3 = \{\mathsf{e(t_1)}, \mathsf{k(t_2)}, \mathsf{e(t_2)}, \mathsf{e(clyde)}, \mathsf{k(fred)}, \mathsf{l(clyde,fred)}\}$.

Let $rk_1(\mathcal{E}_1) = rk_1(\mathcal{E}_2) = 0$, $rk_1(\mathcal{E}_3) = 0$, and $rk_1(\mathcal{E}) = \infty$ for all other EHIs. Then rk_1 is a ranked model of the knowledge base above. Let $rk_2(\mathcal{E}_1) = rk_2(\mathcal{E}_3) = 0$, $rk_2(\mathcal{E}_2) = 1$, and $rk_2(\mathcal{E}) = \infty$ for all other EHIs. Then rk_2 is not a ranked model of $\mathsf{elephant(x)} \wedge \mathsf{keeper(y)} \rightsquigarrow \mathsf{likes(x,y)}$, but is a ranked model of $\mathsf{elephant(x)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{likes(x,fred)}$ and $\mathsf{elephant(clyde)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \mathsf{likes(clyde,fred)}$.

4.2 A Representation Result

In this section we show that ranked interpretations precisely characterise rational satisfaction sets.

Definition 7. The satisfaction set $\mathcal{S}^{rk}$ corresponding to a ranked interpretation *rk* is defined as: $\mathcal{S}^{rk} = \{\alpha \in \mathcal{L} \cup \mathcal{L}^{\rightsquigarrow} : rk \Vdash \alpha\}$.

Our representation result is obtained by showing that all ranked interpretations generate rational satisfaction sets (Theorem 2), and that every rational satisfaction set $\mathcal{S}$ can be realised as the satisfaction set corresponding to some ranked interpretation (Theorem 3).

Theorem 2. *For every ranked interpretation rk, $\mathcal{S}^{rk}$ is a rational satisfaction set.*

To show the converse of Theorem 2, we adapt the representation proof of Lehmann and Magidor Lehmann and Magidor (1992) to the DRFOL setting. The main idea is to show that the defeasible implications in a given rational satisfaction set can be completely characterised by *normal EHIs*, which are EHIs that satisfy all of the defeasible consequences of some compound $A(\vec{x})$. By ranking these normal EHIs over an appropriate linear poset, we can capture the satisfaction set exactly.

Definition 8. For a rational satisfaction set $\mathcal{S}$, the compounds $A(\vec{x}), B(\vec{y})$ are *equally typical* w.r.t. $\mathcal{S}$ (denoted $A(\vec{x}) \equiv_{\mathcal{S}} B(\vec{y})$) iff $A(\vec{x}) \preccurlyeq_{\mathcal{S}} B(\vec{y})$ and $B(\vec{y}) \preccurlyeq_{\mathcal{S}} A(\vec{x})$.

We denote the equivalence class of a compound $A(\vec{x}) \in \mathcal{L}$ with respect to $\equiv_{\mathcal{S}}$ by $[A(\vec{x})]_{\mathcal{S}}$. As predicates can have arbitrarily high arity in general, it is necessary in what follows to assume that $\mathcal{T}$ is a countably infinite set of typicality objects.

Definition 9. Let $\mathcal{S}$ be a rational satisfaction set. Then $\mathcal{E} \in \mathscr{H}_{\mathcal{T}}$ is *normal* for a formula $A(\vec{x}) \in \mathcal{L}$ w.r.t. $\mathcal{S}$ iff the following properties hold:

1. $\mathcal{E} \Vdash \alpha$ for all $\alpha \in \mathcal{S}_C$.
2. $\mathcal{E} \Vdash A(\varphi(\vec{x}))$ for some $\varphi : \text{VAR} \rightarrow \mathcal{T}$.
3. for all $B(\vec{y}) \in [A(\vec{x})]_{\mathcal{S}}$ and $\varphi : \text{VAR} \rightarrow \mathcal{T}$, $B(\vec{y}) \rightsquigarrow C(\vec{z}) \in \mathcal{S}$ implies that $\mathcal{E} \Vdash B(\varphi(\vec{y})) \rightarrow C(\varphi(\vec{z}))$.

The set of normal EHIs for $A(\vec{x})$ is denoted $\text{norm}_{\mathcal{S}}(A(\vec{x}))$. For the rest of this section, we will consider a *fixed* rational satisfaction set $\mathcal{S}$, and sketch the construction of a ranked interpretation $rk : \mathscr{H}_{\mathcal{T}} \rightarrow \Omega \cup \{\infty\}$ such that $\mathcal{S} = \mathcal{S}^{rk}$. First, we show that normal EHIs completely characterise the defeasible implications in a given rational satisfaction set:

Lemma 1. *$A(\vec{x}) \rightsquigarrow B(\vec{y}) \in \mathcal{S}$ iff for every $\mathcal{E} \in$ $norm_{\mathcal{S}}(A(\vec{x}))$ and substitution $\varphi : \text{VAR} \rightarrow \mathcal{T}$ we have $\mathcal{E} \Vdash A(\varphi(\vec{x})) \rightarrow B(\varphi(\vec{y}))$.*

Corollary 1. *$A(\vec{x})$ has a normal EHI iff $A(\vec{x})$ is consistent with respect to $\mathcal{S}$, i.e. $A(\vec{x}) \rightsquigarrow \bot \notin \mathcal{S}$.*

Let $\Omega^* = \{\langle A(\vec{x}), \mathcal{E}\rangle : A(\vec{x}) \in \mathcal{L},\ \mathcal{E} \in \text{norm}_{\mathcal{S}}(A(\vec{x}))\}$. We order elements of Ω^* using the relation $\preccurlyeq_{\mathcal{S}}$ as follows:

$$\langle A(\vec{x}), \mathcal{E}^A\rangle \leq \langle B(\vec{y}), \mathcal{E}^B\rangle \text{ iff } A(\vec{x}) \preccurlyeq_{\mathcal{S}} B(\vec{y})$$

Proposition 3. *$\leq$ is reflexive, transitive and total over Ω^*.*

Let $\Omega = \Omega^*/\sim$ be the quotient of Ω^* with respect to its equivalence classes, which we denote by $[\alpha]_{\leq}$ for $\alpha \in \Omega^*$. By Proposition 3, Ω is a linear poset, though in general it is not well-ordered. We now show that any given EHI is contained in at most one equivalence class:

Lemma 2. *For any $\mathcal{E} \in \mathscr{H}_{\mathcal{T}}$, the following set is either empty or contains a single element:*

$$\Omega(\mathcal{E}) = \{[\langle A(\vec{x}), \mathcal{E}\rangle]_{\leq} : \langle A(\vec{x}), \mathcal{E}\rangle \in \Omega^*\}.$$

This lets us construct a ranking function $rk : \mathscr{H}_{\mathcal{T}} \rightarrow \Omega \cup \{\infty\}$ as follows:

$$rk(\mathcal{E}) = \begin{cases} [\langle A(\vec{x}), \mathcal{E}\rangle]_{\leq} & \text{if } \Omega(\mathcal{E}) = \{[\langle A(\vec{x}), \mathcal{E}\rangle]_{\leq}\} \\ \infty & \text{if } \Omega(\mathcal{E}) = \emptyset \end{cases}$$

Proposition 4. *The ranking function rk* : $\mathscr{H}_{\mathcal{T}} \rightarrow \Omega \cup \{\infty\}$ *is a ranked interpretation.*

Finally, we have the following result relating normal EHIs to minimal elements in *rk*:

Lemma 3. *For any formula* $A(\vec{x}) \in \mathcal{L}$*, we have that* $\min_{rk} \mathscr{H}^{rk}_{\mathcal{T}}(A(\vec{x})) = norm_{\mathcal{S}}(A(\vec{x}))$.

Lemmas 1 and 3 prove the converse to Theorem 2.

Theorem 3. *For every rational satisfaction set* $\mathcal{S}$ *there exists a ranked interpretation rk, over an infinite set of* $\mathcal{T}$ *of typicality objects, such that* $\mathcal{S} = \mathcal{S}^{rk}$.

4.3 Finite Sets of Typicality Objects

Theorem 3 has some limitations in that it requires an infinite set of typicality objects to be true in general. In this section we detail some ways ranked interpretations can be restricted to *finite* sets of typicality objects, which will be useful for defining a basic notion of entailment for DRFOL knowledge bases.

First, consider a fixed finite set $\mathcal{T}' \subset \mathcal{T}$. Note that the set of EHIs over $\mathcal{T}'$ is finite, as there are only finitely many possible atoms over the extended Herbrand base $\mathbb{B}_{\mathcal{T}'}$. Furthermore, given any such $\mathcal{E} \in \mathscr{H}_{\mathcal{T}'}$, we can define a *characteristic compound* for $\mathcal{E}$ that parallels the notion of characteristic formula for a propositional valuation:

Definition 10. Let $\mathcal{E} \in \mathscr{H}_{\mathcal{T}'}$ be an EHI over $\mathcal{T}'$, and $\pi : \mathcal{T}' \rightarrow \textsc{var}$ any injective function. Then the *characteristic compound* for $\mathcal{E}$, denoted $\text{ch}_\pi(\mathcal{E})$, is defined as follows:

$$\text{ch}_\pi(\mathcal{E}) = \bigwedge_{A(\vec{c},\vec{t}) \in \mathbb{B}_{\mathcal{T}'}} \pm A(\vec{c}, \pi(\vec{t}))$$

Here, $\vec{c}$ is a tuple of constants, $\vec{t}$ is a tuple of typicality objects, and $\pm A(\vec{x}, \pi(\vec{t}))$ means $A(\vec{c}, \pi(\vec{t}))$ if $\mathcal{E} \Vdash A(\vec{c}, \vec{t})$, or $\neg A(\vec{c}, \pi(\vec{t}))$ otherwise.

Note that, while $\text{ch}_\pi(\mathcal{E})$ depends on π, the characteristic formula is nevertheless unique up to relabelling of variables and the order of clauses. For this reason we will omit defining π explicitly where we refer to it. The important fact about characteristic formulas is that they reflect satisfaction properties of the underlying EHI $\mathcal{E}$:

Lemma 4. *Let* $\mathcal{E} \in \mathscr{H}_{\mathcal{T}}$ *and* $\mathcal{E}' \in \mathscr{H}_{\mathcal{T}'}$ *be any two EHIs over* $\mathcal{T}$ *and* $\mathcal{T}'$ *respectively such that* $\mathcal{E} \Vdash \varphi(ch_\pi(\mathcal{E}'))$ *for some* $\varphi : \textsc{var} \rightarrow \mathcal{T}$*. Then for any compound* $A(\vec{x})$ *and substitution* $\psi : \textsc{var} \rightarrow \mathcal{T}'$*,* $\mathcal{E}' \Vdash A(\psi(\vec{x}))$ *iff* $\mathcal{E} \Vdash A(\varphi \circ \pi \circ \psi(\vec{x}))$.

The number of typicality objects required to model a defeasible formula depends on the number of variables in the formula. With this in mind, we define the *order* of a formula $A(\vec{x})$ to be the length of the tuple $\vec{x}$.

Definition 11. For any ranked interpretation $rk \in \mathcal{R}_{\mathcal{T}}$, the *restriction of rk to* $\mathcal{E}'$, denoted $rk^* \in \mathcal{R}_{\mathcal{T}'}$, is defined by $rk^*(\mathcal{E}) = \min_{rk} \mathscr{H}^{rk}_{\mathcal{T}}(\text{ch}_\pi(\mathcal{E}))$.

The following lemma proves that *rk** and *rk* agree for formulas of small enough order:

Lemma 5. *rk** *satisfies the following properties, where* $n = |\mathcal{T}'|$ *is the number of typicality objects in* $\mathcal{T}'$*:*

1. *for all classical formulas* $\alpha \in \mathcal{L}$*, rk*$^* \Vdash \alpha$ *iff rk* $\Vdash \alpha$.
2. *for all defeasible formulas* $\alpha \in \mathcal{L}^{\rightsquigarrow}$ *of order* $\leq n$*, rk*$^* \Vdash \alpha$ *iff rk* $\Vdash \alpha$.

This lets us define approximations to any given ranked interpretation using a finite subset of typicality objects. In particular, if one only cares about satisfaction for formulas of bounded order, then a finite set suffices to model them. Defining the order of a knowledge base to be the maximum order of any formula contained within it, we have the following corollary:

Corollary 2. *Let* $\mathcal{K} \subseteq \mathcal{L} \cup \mathcal{L}^{\rightsquigarrow}$ *be any knowledge base of order* n*. Then* $\mathcal{K}$ *has a ranked model iff it has a ranked model over a set* $\mathcal{T}'$ *of typicality objects where* $|\mathcal{T}'| = n$.

5 Defeasible Entailment

A central question that we have left unaddressed until now is *entailment*. That is, given a DRFOL knowledge base $\mathcal{K}$, when are we justified in asserting that a DRFOL formula α follows defeasibly from $\mathcal{K}$? In this section we provide one answer to this question by defining a semantic version of *Rational Closure* (Lehmann and Magidor 1992) for DRFOL. It is, by now, well-established that systems for defeasible reasoning are amenable to multiple forms of defeasible entailment, and the work we present in this section should therefore be viewed as the first step in a larger investigation into defeasible entailment.

Rational Closure is a well-known framework for non-monotonic reasoning that can be viewed as one of the core forms of defeasible entailment in KLM-style reasoning. Due to the so-called *drowning effect* (Benferhat et al. 1993), it is considered inferentially too weak for some application domains. Despite that, it is a semantic construction that can be extended to obtain other interesting entailment relations (Lehmann 1995; Casini and Straccia 2013; Casini et al. 2014; Giordano and Gliozzi 2019). It has gained attention in the framework of DLs (Casini and Straccia 2010; Britz et al. 2020; Giordano et al. 2015; Bonatti et al. 2015). An equivalent semantic construction, System Z (Pearl 1990), has been considered for unary first-order logic (Kern-Isberner and Beierle 2015; Beierle et al. 2016, 2017). Several equivalent definitions of Rational Closure can be found in the literature. Here we refer to the one due to Booth and Paris (1998).

Let a knowledge base $\mathcal{K}$ be a set of propositional defeasible implications $\alpha \mathrel{|\!\sim} \beta$ (see Section 3). Booth and Paris provide a construction with the following two immediate consequences:

1. Given all the ranked models of $\mathcal{K}$ there is a model $\mathscr{R}^*$ of $\mathcal{K}$, that we can call the *minimal* one, which is such that it assigns to every propositional valuation v the *minimal* rank assigned to it by any of the ranked models of $\mathcal{K}$.
2. Propositional Rational Closure can be characterised using $\mathscr{R}^*$. That is, $\alpha \mathrel{|\!\sim} \beta$ is in the (propositional) Rational Closure of $\mathcal{K}$ iff $\mathscr{R}^* \Vdash \alpha \mathrel{|\!\sim} \beta$. The intuition behind the use of the ranked model $\mathscr{R}^*$ for the definition of entailment is that it formalises the *presumption of typicality* (Lehmann 1995): assigning to each valuation the lowest possible rank, we model a reasoning pattern in which

we assume that we are in one of the most typical situations that are compatible with our knowledge base.

Based on Corollary 2 and the other results in Section 4.3, we can define an analogous construction for DRFOL:

Definition 12. Let $\mathcal{K} \subseteq \mathcal{L} \cup \mathcal{L}^{\leadsto}$ be a DRFOL knowledge base of order n, and take $\mathcal{T}' \subset \mathcal{T}$ to be a finite set of typicality objects of cardinality n. Then the *minimal ranked interpretation* of $\mathcal{K}$, which we denote by $rk_{\mathcal{K}} : \mathscr{H}_{\mathcal{T}'} \to \mathbb{N} \cup \{\infty\}$, is defined as follows:

$$rk_{\mathcal{K}}(\mathcal{E}) = \min\{rk(\mathcal{E}) : rk \in \mathcal{R}_{\mathcal{T}'} \text{ and } rk \Vdash \mathcal{K}\}$$

Note that we take $\min \emptyset = \infty$ by convention, and that $rk_{\mathcal{K}}$ is a ranked interpretation over $\mathcal{T}'$, hence $rk_{\mathcal{K}} \in \mathcal{R}_{\mathcal{T}'}$. Intuitively, $rk_{\mathcal{K}}$ is what you get if you let every EHI rank as low as possible amongst the models of $\mathcal{K}$. This minimal ranked interpretation can be used to define a defeasible entailment relation for DRFOL:

Definition 13. For any DRFOL knowledge base $\mathcal{K}$ and formula α, we say that α is in the *Rational Closure* of $\mathcal{K}$, denoted $\mathcal{K} \mid\!\approx_{rc} \alpha$, iff $rk_{\mathcal{K}} \Vdash \alpha$.

Example 3. Consider the knowledge base $\mathcal{K}$ from Example 2. We add the unary predicate $\mathsf{purple(x)}$ to PRED . The order of $\mathcal{K}$ is 2, so we build our minimal model $rk_{\mathcal{K}}$ using the set of EHIs $\mathscr{H}_{\mathcal{T}'}$, where the set of typical constants is $\mathcal{T}' = \{t_1, t_2\}$. Since $\mathcal{K}$ does not contain classical formulas, there are no EHIs of infinite rank. All the EHIs satisfying $\mathcal{K}$ will be assigned rank 0. That is, all the EHIs in which if t_i is an elephant and t_j is a keeper ($i, j \in \{1, 2\}$), t_i likes t_j but, if fred is a keeper, t_i does not like fred. Also, if fred is a keeper and clyde is an elephant, clyde likes fred. All the other EHIs will be assigned rank 1. For example, the EHI $\mathcal{E}_1$ from Example 2 would have rank 0, while $\mathcal{E}_3$ would have rank 1, since it does not satisfy the formula $\mathsf{elephant(x)} \land \mathsf{keeper(y)} \leadsto \mathsf{likes(x,y)}$ ($\mathcal{E}_2$ is not considered in $rk_{\mathcal{K}}$, since it uses the constant t_3).

It then follows that a desirable form of constrained monotonicity, formalised by (RM), holds. Note that all the EHIs at rank 0 in the minimal model $rk_{\mathcal{K}}$ would either satisfy $\mathsf{purple}(t_i)$ ($i \in \{1, 2\}$) or not, since it is irrelevant to the satisfaction of $\mathcal{K}$. The outcome would be that, while of course satisfying the formula $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \leadsto \neg\mathsf{likes(x,fred)}$, since it is in $\mathcal{K}$, $rk_{\mathcal{K}}$ would not satisfy $\mathsf{elephant(x)} \land \mathsf{keeper(fred)} \leadsto \neg\mathsf{purple}(x)$, while it would satisfy $\mathsf{elephant(x)} \land \mathsf{purple}(x) \land \mathsf{keeper(fred)} \leadsto \neg\mathsf{likes(x,fred)}$.

More generally, Rational Closure, in the propositional and DL cases, satisfies a number of attractive properties:

$$\begin{array}{ll} (\text{INCL}) & \alpha \in \mathcal{K} \text{ implies } \mathcal{K} \mid\!\approx_{rc} \alpha \\ (\text{SMP}) & \mathcal{S} = \{\alpha : \mathcal{K} \mid\!\approx_{rc} \alpha\} \text{ is rational} \end{array}$$

It is straightforward that our definition of $\mid\!\approx_{rc}$ carries over to these properties:

Theorem 4. *The entailment relation* $\mid\!\approx_{rc}$ *satisfies* (INCL) *and* (SMP).

It is worthwhile delving a bit deeper into each of these properties. The first one, (INCL), also known as Inclusion, simply requires that statements in $\mathcal{K}$ also be defeasibly entailed by $\mathcal{K}$. It is a meta-version of the (REFL) rationality postulate for propositional logic (described in Section 2) and for DRFOL (described in Section 4). While the property itself might seem self-evident, it is instructive to view it in concert with the definition of $rk_{\mathcal{K}}$. From this it follows that $rk_{\mathcal{K}}$, which essentially defines Rational Closure, is the ranked interpretation in which EHIs are assigned a ranking that is truly as low (i.e., as typical) as possible, subject to the constraint that $rk_{\mathcal{K}}$ is a model of $\mathcal{K}$. This aligns with the intuition of propositional Rational Closure which requires of propositional valuations in a ranked interpretation to be as typical as possible.

(SMP) requires the set of statements corresponding to the Rational Closure of knowledge base $\mathcal{K}$ to be rational (in the sense of Definition 3). By virtue of Theorem 3, this requires defeasible entailment to be characterised by a *single* ranked interpretation. This accounts for the fact that the property is also referred as the Single Model Property.

6 Comparison

Given that KLM-style defeasible reasoning started off as a propositional endeavour, it makes sense to begin this section with a formal comparison to the propositional case. Note firstly that, when restricted to 0-ary predicates, the language of RFOL reduces to a propositional one. In this case the Herbrand universe becomes superfluous, the Herbrand base is the set of 0-ary predicates (propositional atoms), and a Herbrand interpretation is a subset of propositional atoms. Clearly then, Herbrand interpretations reduce to propositional valuations. For DRFOL we work with enriched Herbrand interpretations in which typicality objects are added to the Herbrand universe. But since the Herbrand universe plays no role in the semantics of 0-ary predicates, it is redundant. The ranked interpretations for DRFOL (Definition 5) then reduce to propositional ranked interpretations (described in Section 3, from which it follows that defeasible implication in DRFOL reduces to propositional defeasible implication (represented by the symbol $\mid\!\sim$ in Section 3). More specifically, consider a defeasible propositional language generated from a set of atoms, and take this set to be the 0-ary predicates of a DRFOL language. It follows that for every propositional ranked interpretation $\mathscr{R}$ there is a DRFOL ranked interpretation rk such that for all propositional statements α, β constructed from $\neg, \land$ and $\lor$, $rk \Vdash \alpha \leadsto \beta$ iff $\mathscr{R} \Vdash \alpha \mid\!\sim \beta$ and $rk \Vdash \alpha$ iff $\mathscr{R} \Vdash \alpha$. Conversely, for every DRFOL ranked interpretation rk, there is a propositional ranked interpretation $\mathscr{R}$ such that for all propositional statements α, β constructed from $\neg, \land$ and $\lor$, $rk \Vdash \alpha \leadsto \beta$ iff $\mathscr{R} \Vdash \alpha \mid\!\sim \beta$ and $rk \Vdash \alpha$ iff $\mathscr{R} \Vdash \alpha$.

A similar result holds when DRFOL is restricted to compounds, implications, and defeasible implications that are all ground. Considering RFOL first, observe that, unlike the case discussed above, the Herbrand universe is used to construct the Herbrand base here, and it is therefore used in the definition of Herbrand interpretations. But since we only consider ground statements, each ground atom in a Herbrand interpretation effectively functions like a propositional atom, which again means that Herbrand interpretations reduce to propositional valuations (for the propositional language with the ground atoms as its set of propositional atoms). Moving on to DRFOL we note that since we are restricted to ground statements, the substitutions referred to in Definition 6 do not play any role, which means that the typicality objects in enriched Herbrand interpretations are redundant. In summary, consider a defeasible propositional language generated from the ground atoms of a language of DRFOL. It follows that for every propositional ranked interpretation $\mathscr{R}$ there is a DRFOL ranked interpretation *rk* such that for all propositional statements α, β constructed from $\neg, \wedge$ and $\vee$, $rk \Vdash \alpha \rightsquigarrow \beta$ iff $\mathscr{R} \Vdash \alpha \mid\!\sim \beta$ and $rk \Vdash \alpha$ iff $\mathscr{R} \Vdash \alpha$. And conversely, for every DRFOL ranked interpretation *rk*, there is a propositional ranked interpretation $\mathscr{R}$ such that for all propositional statements α, β constructed from $\neg, \wedge$ and $\vee$, $rk \Vdash \alpha \rightsquigarrow \beta$ iff $\mathscr{R} \Vdash \alpha \mid\!\sim \beta$ and $rk \Vdash \alpha$ iff $\mathscr{R} \Vdash \alpha$.

Space considerations prevent us from providing a detailed comparison of DRFOL with $\mathcal{DALC}$, the defeasible version of the DL $\mathcal{ALC}$ (Britz et al. 2020). Suffice it to note that when $\mathcal{DALC}$ is stripped of existential and value restrictions and confined to Tbox statements, and when DRFOL is restricted to unary predicates and open implications (defeasible and classical), every concept C in $\mathcal{DALC}$ can be mapped to a compound $C(x)$ in DRFOL, and vice versa. It is then possible to obtain a result that is analogous to the propositional cases above, with one exception: a defeasible implication of the form $C(x) \rightsquigarrow \bot$ has a meaning that is different than $C \mathrel{\sqsubset\!\!\!\sim} \bot$, its $\mathcal{DALC}$ counterpart.

This marks an important distinction between DRFOL and both the propositional KLM framework and $\mathcal{DALC}$, in which classical statements are equivalent to certain defeasible implications. In the propositional case α is equivalent to $\neg\alpha \mid\!\sim \bot$ ($\mathscr{R} \Vdash \alpha$ iff $\mathscr{R} \Vdash \neg\alpha \mid\!\sim \bot$ for all $\mathscr{R}$) while, for $\mathcal{DALC}$, $C \sqsubseteq \bot$ is equivalent to $C \mathrel{\sqsubset\!\!\!\sim} \bot$. But in DRFOL, defeasible implications *cannot* inform us about compounds or classical implications. Formally, rational satisfaction sets do *not* necessarily satisfy the following postulate:

$$(\textsc{Sub})\ \ \frac{A(\vec{x}) \rightsquigarrow \bot \in \mathcal{S}}{A(\vec{x}) \rightarrow \bot \in \mathcal{S}}$$

One way this difference manifests itself is in the way our framework handles the finitary Lottery Paradox (Poole 1991). Consider the DRFOL knowledge base $\mathcal{K} = \{\mathsf{penguin(x)} \rightarrow \mathsf{bird(x)}, \mathsf{cuckoo(x)} \rightarrow \mathsf{bird(x)}, \mathsf{bird(x)} \rightarrow \mathsf{cuckoo(x)} \vee \mathsf{penguin(x)}, \mathsf{bird(x)} \rightsquigarrow \mathsf{flies(x)} \wedge \mathsf{nests(x)}, \mathsf{cuckoo(x)} \rightarrow \neg\mathsf{nests(x)}, \mathsf{penguin(x)} \rightarrow \neg\mathsf{flies(x)}\}$. This can also be modelled as a propositional defeasible knowledge base and as a $\mathcal{DALC}$ knowledge base.

In all three cases KLM rationality dictates that being a bird defeasibly implies a contradiction: $\mathsf{bird(x)} \rightsquigarrow \bot$ in the case of DRFOL, $\mathsf{bird} \mid\!\sim \bot$ in the propositional defeasible case, and $\mathsf{Bird} \mathrel{\sqsubset\!\!\!\sim} \bot$ in the case of $\mathcal{DALC}$. In the defeasible propositional case this means there are no birds ($\mathsf{bird} \mid\!\sim \bot$ is equivalent to $\neg\mathsf{bird}$). Similarly for $\mathcal{DALC}$, where $\mathsf{Bird} \mathrel{\sqsubset\!\!\!\sim} \bot$ is equivalent to $\mathsf{Bird} \sqsubseteq \bot$. In DRFOL, however, $\mathsf{bird(x)} \rightsquigarrow \bot$ is *not* equivalent to $\mathsf{bird(x)} \rightarrow \bot$. Rather than stating that there are no birds, $\mathsf{bird(x)} \rightsquigarrow \bot$ means that there are no *typical* birds. This leaves open the possibility of there being only atypical birds, something that is not possible in the propositional and DL cases.

Example 4. Let $\textsc{const} = \{\mathsf{tweety}\}$, $\textsc{var} = \{x\}$, $\textsc{pred} = \{\mathsf{bird}, \mathsf{penguin}, \mathsf{cuckoo}, \mathsf{flies}, \mathsf{nests}\}$, with $\mathcal{T} = \{\mathsf{t_1}, \ldots\}$ the set of typicality objects. Let *rk* be the ranked interpretation for which $rk(\mathcal{E}) = 0$ and $rk(\mathcal{E}') = \infty$ for all other EHIs, where $\mathcal{E} = \{\mathsf{bird(tweety)}, \mathsf{penguin(tweety)}\}$. It is easily verified that *rk* satisfies all statements in the DRFOL knowledge base $\mathcal{K}$ above, and also satisfies $\mathsf{bird(x)} \rightsquigarrow \bot$, since $rk \not\Vdash \mathsf{bird(t_i)}$ for all i. But *rk* does not satisfy $\mathsf{bird(x)} \rightarrow \bot$.

We regard this as a significant advantage of DRFOL over previous KLM-style defeasible formalisms.

As a final remark, observe that this distinction is not in conflict with the claim that DRFOL is a proper generalisation of propositional defeasible logic. For a ground compound α (including those containing 0-ary predicates) it is indeed the case that $\alpha \rightsquigarrow \bot$ is equivalent to $\alpha \rightarrow \bot$. It is when α is an open compound that ($\textsc{Sub}$) need not hold.

7 Related Work

Defeasible reasoning is part of a broader research programme on conditional reasoning (Arlo-Costa 2019), most of which was developed for propositional logic. This paper falls in the class of approaches aimed at moving beyond propositional expressivity. We pointed out the connection with defeasible DLs (Casini and Straccia 2010, 2013; Casini et al. 2015; Giordano et al. 2013, 2015; Bonatti et al. 2015; Bonatti 2019; Pensel and Turhan 2018) in Section 6, but there have also been proposals to extend this approach to first-order logic. Most of these define a preferential order over the elements of the first-order domain (Schlechta 1995; Brafman 1997; Delgrande and Rantsoudis 2020), in line with some of the DL proposals (Giordano et al. 2015; Britz et al. 2020), and present rationality postulates, but they do not provide characterisations in terms of rationality postulates. Others (Delgrande 1998; Kern-Isberner and Thimm 2012) are formally closer to our work in that they use preference orders over interpretations.

Delgrande (1998) proposes a semantics that is closer to the intuitions behind *circumscription* (McCarthy 1980), giving preference to interpretations that minimise the counterexamples to defeasible conditionals. On the other hand, Kern-Isberner and Thimm (2012) propose a technical solution that is much closer to the work we present here. Like ours, their semantics is based on Herbrand interpretations. They define *ordinal conditional functions* over the set of Herbrand interpretations, obtaining a structure that is very close to our ranked interpretations. They identify some individuals as *representatives* of a conditionals. This is done to formalise the same intuition (or, at least, an intuition that is very similar) that underlies our decision to introduce typicality objects. Apart from other formal differences (e.g. the expressivity of their language is slightly different), their work focuses on the definition of a notion of entailment based on a specific semantic construction carried over from the propositional framework known as *c-representations* of a conditional knowledge base (Kern-Isberner 2001, 2004). In contrast, our focus in this paper is on getting the theoretical foundations of defeasible reasoning for restricted first-order logics in place. Thus, our work here is centred around a representation result that provides a characterisation of the semantics in terms of structural properties. And while we present some results on defeasible entailment in Section 5, we have left a more in-depth study of this important topic as future work. Indeed, it is our conjecture that the foundations we have put in place in this paper will allow for the definition of more than one form of defeasible entailment. At the same time, a more in-depth comparison with the proposal of Kern-Isberner and Thimm is also necessary. We leave that for future work.

Kern-Isberner and Beierle (2015); Beierle et al. (2016, 2017) use the same semantic approach of Kern-Isberner and Thimm (2012) to develop an extension of Pearl's System Z (1990) for first-order logic, but they restrict their attention to unary predicates. System Z is a form of entailment that is very close to the approach we introduce in Section 5.

Brafman (1997) suggests that preference orders over the domain should result in forms of reasoning quite different from the use of preference orders over interpretations, comparable to the difference between statistical and subjective readings of probabilities. We leave a proper investigation of the differences between these two different modelling solutions as future work.

As mentioned, the final goal of our investigation is the development of a defeasible extension of Datalog+/-. To the best of out knowledge there is no research on the introduction of defeasible implication in Datalog+/-. Of course, there is a longstanding tradition of non-monotonic extensions of Disjunctive Datalog with an Answer Set semantics (Leone et al. 2006). Although there are some connections between conditional reasoning (of which defeasible reasoning is a special case) and negation-as-failure (Makinson 1994, 2005), these two approaches are different. Answer Set Programming is a popular solution to model the closed-world assumption, while conditional reasoning is focused on reasoning with the potential conflicts resulting from defeasible pieces of information.

8 Conclusion and Future Work

In this paper we have laid the theoretical groundwork for KLM-style defeasible Datalog (DRFOL). Our primary contribution is a set of rationality postulates describing the behaviour of defeasibility in DRFOL, a typicality semantics for interpreting defeasibility in DRFOL, and a representation result, proving that the proposed postulates characterise the semantic behaviour precisely.

With the theoretical core in place, we then proceeded to define a form of defeasible entailment for DRFOL that can be viewed as the DRFOL equivalent of the propositional form of defeasible entailment known as Rational Closure.

There are at least three important avenues for future research. The first one relates to a more detailed investigation of defeasible entailment for DRFOL knowledge bases. While Rational Closure for DRFOL is on par with the analogous notions for propositional logic and DLs (restricted to Tboxes), it is not able to fully manage reasoning about individuals. Going back to Example 3, assume that we add a constant bob to CONST. Since we are not informed of anything atypical about bob, we would like to be able to infer the statement $\mathsf{elephant(bob)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{likes(bob, fred)}$. But Rational Closure does not sanction this, since the formula $\mathsf{elephant(x)} \wedge \mathsf{keeper(fred)} \rightsquigarrow \neg\mathsf{likes(x, fred)}$ is evaluated only on the typicality constants, and whether bob behaves in a typical way or not is irrelevant w.r.t. the satisfaction of the knowledge base. Consequently, on rank 0 of $rk_{\mathcal{K}}$ there are EHIs in which $\mathsf{elephant(bob)}$ behaves like an atypical elephant. Rational Closure would therefore need to be refined to model the inferences about individuals properly.

Next we discuss a more general point about defeasible entailment. Based on the theoretical basics we have put in place and the preliminary work on Rational Closure for DRFOL, we conjecture that all appropriate forms of DRFOL defeasible entailment will satisfy the (SMP) property, thereby ensuring that all forms of defeasible entailment are rational. This will be similar to the propositional case (Lehmann 1995; Booth and Paris 1998; Giordano et al. 2015), and unlike the case for DLs (Casini and Straccia 2010; Casini et al. 2013).

With a suitable definition (or definitions) of DRFOL defeasible entailment in place, the next step is to investigate algorithms for computing DRFOL defeasible entailment. Here we plan to draw inspiration from both the propositional and DL cases, where defeasible entailment can be reduced to a series of classical entailment checks, sometimes in polynomial time and with a polynomial number of classical entailment checks (Casini, Straccia, and Meyer 2019; Giordano et al. 2015; Casini, Meyer, and Varzinczak 2019).

Finally, in line with our stated aim in Section 1, the basic theoretical framework presented in this paper places us in a position to see whether the work on DRFOL can be extended to Datalog +/-.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

Arlo-Costa, H. 2019. The Logic of Conditionals. In *The Stanford Encyclopedia of Philosophy*. Summer 2019 edition.

Beierle, C.; Falke, T.; Kutsch, S.; and Kern-Isberner, G. 2016. Minimal Tolerance Pairs for System Z-Like Ranking Functions for First-Order Conditional Knowledge Bases. In *Proc. of FLAIRS 2016*, 626–631. AAAI Press.

Beierle, C.; Falke, T.; Kutsch, S.; and Kern-Isberner, G. 2017. System Z^{FO}: Default reasoning with system Z-like ranking functions for unary first-order conditional knowledge bases. *Int. J. Approx. Reason.* 90: 120–143.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency Management and Prioritized Syntax-based Entailment. In *Proc. of IJCAI-93*, 640–645. Morgan Kaufmann Publishers Inc.

Bonatti, P. A. 2019. Rational closure for all description logics. *Artificial Intelligence* 274: 197–223.

Bonatti, P. A.; Faella, M.; Petrova, I. M.; and Sauro, L. 2015. A new semantics for overriding in description logics. *Artificial Intelligence* 222: 1–48.

Booth, R.; and Paris, J. B. 1998. A Note on the Rational Closure of Knowledge Bases with Both Positive and Negative Knowledge. *J. Log. Lang. Inf.* 7(2): 165–190.

Brafman, R. I. 1997. A First-order Conditional Logic with Qualitative Statistical Semantics. *J. Log. Comput.* 7(6): 777–803.

Britz, K.; Casini, G.; Meyer, T.; Moodley, K.; Sattler, U.; and Varzinczak, I. 2020. Principles of KLM-Style Defeasible Description Logics. *ACM T. Comput. Log.* 22(1).

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general Datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics* 14(C): 57–83.

Calì, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; and Pieris, A. 2010. Datalog+/-: A Family of Logical Knowledge Representation and Query Languages for New Applications. In *Proc. of LICS 2010*, 228–242. IEEE.

Casini, G.; Meyer, T.; Moodley, K.; and Nortjé, R. 2014. Relevant Closure: A New Form of Defeasible Reasoning for Description Logics. In *Proc. of JELIA 2014*, volume 8761 of *LNCS*, 92–106. Springer.

Casini, G.; Meyer, T.; Moodley, K.; Sattler, U.; and Varzinczak, I. 2015. Introducing Defeasibility into OWL Ontologies. In *Proc. of ISWC 2015*, volume 9367 of *LNCS*, 409–426. Springer.

Casini, G.; Meyer, T.; Moodley, K.; and Varzinczak, I. 2013. Nonmonotonic reasoning in Description Logics: Rational Closure for the ABox. In *Proceedings of DL-13*, 600–615. CEUR-WS.org.

Casini, G.; Meyer, T.; and Varzinczak, I. 2019. Taking Defeasible Entailment Beyond Rational Closure. In *Proc. of JELIA 2019*, volume 11468 of *LNCS*, 182–197. Springer.

Casini, G.; and Straccia, U. 2010. Rational Closure for Defeasible Description Logics. In *Proc. of JELIA 2010*, volume 6341 of *LNCS*, 77–90. Springer-Verlag.

Casini, G.; and Straccia, U. 2013. Defeasible Inheritance-Based Description Logics. *Journal of Artificial Intelligence Research* 48: 415–473.

Casini, G.; Straccia, U.; and Meyer, T. 2019. A Polynomial Time Subsumption Algorithm for Nominal Safe $\mathcal{ELO}_\bot$ under Rational Closure. *Information Sciences* 501: 588–620.

Delgrande, J. P. 1998. On first-order conditional logics. *Artificial Intelligence* 105(1): 105 – 137.

Delgrande, J. P.; and Rantsoudis, C. 2020. A Preference-Based Approach for Representing Defaults in First-Order Logic. In *Proc. of NMR 2020*, 120–129.

Giordano, L.; and Gliozzi, V. 2019. Strengthening the Rational Closure for Description Logics: An Overview. In *Proc. of CILC 2019*, 68–81. CEUR-WS.org.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2013. A non-monotonic Description Logic for reasoning about typicality. *Artificial Intelligence* 195: 165–202.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Art. Int.* 226: 1–33.

Kern-Isberner, G. 2001. *Conditionals in Nonmonotonic Reasoning and Belief Revision - Considering Conditionals as Agents*, volume 2087 of *LNCS*. Springer.

Kern-Isberner, G. 2004. A Thorough Axiomatization of a Principle of Conditional Preservation in Belief Revision. *Ann. Math. Artif. Intell.* 40(1-2): 127–164.

Kern-Isberner, G.; and Beierle, C. 2015. A System Z-like Approach for First-Order Default Reasoning. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, volume 9060 of *LNCS*, 81–95. Springer.

Kern-Isberner, G.; and Thimm, M. 2012. A Ranking Semantics for First-Order Conditionals. In *Proc. of ECAI 2012*, 456–461. IOS Press.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44: 167–207.

Lehmann, D. 1995. Another perspective on default reasoning. *Ann. Math. Artif. Intell.* 15(1): 61–82.

Lehmann, D.; and Magidor, M. 1992. What does a conditional knowledge base entail? *Art. Intell.* 55: 1–60.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006. The DLV System for Knowledge Representation and Reasoning. *ACM T. Comput. Log.* 7(3): 499–562.

Makinson, D. 1994. General Patterns in Nonmonotonic Reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. III*, 35–110. Clarendon Press.

Makinson, D. 2005. *Bridges from Classical to Nonmonotonic Logic*. King's College Publications.

McCarthy, J. 1980. Circumscription, a form of nonmonotonic reasoning. *Art. Intell.* 13(1-2): 27–39.

Pearl, J. 1990. System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proc. of TARK 1990*.

Pensel, M.; and Turhan, A.-Y. 2018. Reasoning in the Defeasible Description Logic $\mathcal{EL}_\bot$ - computing standard inferences under rational and relevant semantics. *Int. J. Approx. Reason.* 103: 28 – 70.

Poole, D. 1991. The Effect of Knowledge on Belief: Conditioning, Specificity and the Lottery Paradox in Default Reasoning. *Art. Intell.* 49(1-3): 281–307.

Schlechta, K. 1995. Defaults as Generalized Quantifiers. *Journal of Logic and Computation* 5(4): 473–494.

On the Cognitive Adequacy of Non-monotonic Logics

Sara Todorovikj[1,2] , **Gabriele Kern-Isberner**[3] , **Marco Ragni**[1,2]
[1]Cognitive Computation Lab, Technical Faculty, University of Freiburg
[2]Danish Institute for Advanced Study, University of Southern Denmark
[3]Technical University Dortmund
{satod, ragni}@sdu.dk, gabriele.kern-isberner@cs.uni-dortmund.de

Abstract

Humans have the ability to reason conditionally despite the existence of disablers. They have the capability to consider content and background knowledge and they are prototypical non-monotonic reasoners. So far most research has focused on explaining an "average" reasoner and neglected the individual reasoning process. Towards identifying the specifics of human reasoning, we investigate the inference mechanism for conditional reasoning considering experimental data presented by a previous psychological study. The experimental material included a range of different problems and contents with varying amounts of disablers and alternatives. We consider individual inference patterns and explain them by a ranking on worlds and ordinal conditional functions. We investigate: (i) Do effects found on aggregate level still hold on the individual level, and if yes - to which extent? (ii) How can possible disablers and alternatives change the inference pattern? (iii) How do individuals differ among each other and are there any common patterns? With this analysis we show how non-monotonic logic provides a suitable tool to express and explain the specifics of human reasoning formally in a more coherent way than classical logic.

1 Introduction

You are given the following information (Singmann, Klauer, and Beller 2016):

> If a balloon is pricked with a needle, then it will pop.
> A balloon is pricked with a needle.

Then, you are asked to answer the following question with an *endorsement* in the form of a *probability value* between 0% and 100%:

> How likely is it that it will pop?

Given the information you are provided with, and no reason to believe otherwise, your answer would most likely tend towards 100%. However, in this world of balloons and needles, consider the following information:

> The balloon is without air, i.e., empty.

If you mentally consider such situations where the balloon would *not* pop, then your endorsement will most likely be lower than 100%. States like this are called *disablers*. On the other hand, there can be additional cases, e.g.:

> The balloon is pricked with a pen.

that are called *alternatives*.

Depending on the different scenarios, in the form of disablers and alternatives, that an individual knows about and can think of, their endorsements can vary to a great extent. E.g., in the balloon scenario, after considering the information that the balloon might be without air, your answer might be 95% instead of 100%. Another person, due to their own personal background, might consider that information as more influential, so they would answer with e.g., 80%. In human reasoning literature many have focused on aggregating over an experiment's participants and just explaining the most frequently given answers. However, examples like this point to the need for an analysis shift to the individual level.

Table 1: Conditional Inference Forms

Premise	MP	AC	DA	MT
Major	X→Y	X→Y	X→Y	X→Y
Minor	X	Y	¬X	¬Y
Conclusion	Y	X	¬Y	¬X

Conditionals are statements usually of the form "If X then Y", or equivalently "Y, if X" (also written as X→Y, where X is the *antecedent* and Y the *consequent*). Conditionals are relevant in everyday life and science to describe causal, counterfactual, and other forms of relations between two propositions X and Y. By combining a conditional (also called a *major premise*) with a current state of a proposition (also called a *minor premise*), a *conclusion* can be inferred about the state of the other proposition. There are four major inference forms: *modus ponens* (MP), *modus tollens* (MT), *affirming the consequent* (AC) and *denying the antecedent* (DA), as shown in Table 1. Humans systematically deviate from interpreting conditionals as material implication (Ragni, Kola, and Johnson-Laird 2018). Despite more than 50 years of research there is still no cognitive theory that can fully explain human conditional reasoning processes and effects recognized through experimental data (Ragni, Dames, and Johnson-Laird 2019). Learning more about how individuals interpret different conditional reasoning tasks is crucial in order to take a step forward towards understanding human reasoning.

As motivated earlier, in this paper, we turn to the individual participant. We are interested in the inference process of the individual, we want to investigate which conclusion depending on the inference types this individual endorses and which not and how this individual differs in applying these inference mechanisms.

Eichhorn, Kern-Isberner, and Ragni (2018) propose an inference analysis approach using *inference patterns* based upon conditional logic. Contrary to previous research that largely focuses on the inference forms individually, they joined all inferences into one tuple. Here, we propose an enhancement that allows for inference patterns to be applied on *probabilistic* experimental data, by also taking into account the relationships between the inference form *endorsements*. Similarly, here the rationality of an inference pattern is assessed based on a plausibility semantics derived from preferential models (Makinson 1994) respectively Ordinal Conditional Functions (OCF, (Spohn 1988)), allowing for a deviation from following logical inference rules. Following this idea, we propose total preorders over possible worlds as *preferential mental models*, serving as cognitive models for reasoning of humans when they are presented with a conditional reasoning task.

In this way, we combine basic approaches from nonmonotonic reasoning and cognitive science on a deep methodological level to set up a formal framework of human reasoning that goes beyond classical logic, but does not need quantifications via, e.g., probabilities in the first place. These preferential mental models can be applied on the level of individuals, as well as on an aggregated level, to reveal basic structures of reasoning.

A *mental model* consists of the true states of the propositions in a premise. Given a conditional premise "If A then B", its mental model representation would consider the states of the propositions A and B. One of the most prominent reasoning theories that uses mental models is the Mental Model Theory (Johnson-Laird and Byrne 1991; Johnson-Laird and Byrne 2002). It assumes that when presented with a conditional, individuals start with an initial model where both propositions are true:

A B
...

Once the initial mental model is created, it triggers the recollection of relevant facts and background knowledge related to the conditional premise (Johnson-Laird and Byrne 1991). With that the initial model is either confirmed as correct or it stimulates the individual to engage in search for counterexamples, which would lead to the so-called *fleshed-out* representation, consisting of all states for which the conditional *holds*:

A B
¬A ¬B
¬A B

However, an interesting question is – given that there are limited cognitive capacities, which models are constructed and which are neglected? Is it possible to reverse-engineer the underlying rank of models and identify the preferred mental models? Which influence do alternatives or disablers have? Is it possible to formally found the Mental Model Theory (Johnson-Laird and Byrne 2002)? This will be investigated in the paper.

The paper is structured as follows. In the next section we will provide the empirical bases, an experimental study conducted by Singmann et al. (2016). Then we introduce some formal preliminaries and introduce plausible reasoning. In Section 5 we introduce the formal foundation for inference patterns and in Section 6 how they can be extended towards endorsements. In Section 7 we explain the empirical results with the formal framework we have developed. Section 8 discusses and concludes the article.

2 Experimental Data

Singmann et al. (2016) present four experiments in which they studied endorsement rates of the respective conclusions for the four inference forms. In three of them they use contents with a varying amount of disablers and alternatives. The fourth experiment manipulates the speaker expertise and differs from the others, hence, we do not consider it here.

The experimental data by Singmann et al. (2016)[1] considered here is from Experiments 1, 3a and 3b. In all three experiments, participants gave endorsements for the four inference forms. The contents are the same in all three experiments and they vary in the amounts of disablers and alternatives associated with them, quantified with 'Few' and 'Many', as shown in Table 2. Moreover, in Experiments 3a and 3b, participants were divided in three groups. In two of them they are given additional information in the form of disablers and alternatives, whereas in the last group participants received only the conditional task. Participants were asked to endorse the conclusion as a probability in the range 0 - 100%.

Each content is presented as a reduced inference (no major premise), e.g., for MP:

$$\frac{\text{A balloon is pricked with a needle.}}{\text{How likely is it that it will pop?}}$$

and additionally as a full conditional inference.

In the original study, Singmann et al. (2016) aggregate the participants from all three experiments, which is the approach that we also follow here. The number of participants in Exp. 1 is N = 31, in Exp. 3a is N = 77 and Exp. 3b is N = 91, making the total N = 199.

Table 3 presents the average endorsement values among participants for each inference form for all contents in both conditional presentation forms (reduced and full inference).

3 Formal Preliminaries

Building up on Eichhorn et al. (2018), we base our formal modeling approach on propositional logic with a language set up from a finite set of propositional atoms $\Sigma = \{V_1, \ldots, V_m\}$ which can be interpreted to be *true* (v_i) or *false* ($\overline{v}_i$). The propositional language $\mathfrak{L}$ is composed from Σ with the logical connectives *and* ($\wedge$), *or* ($\vee$), and

[1] The data can be found at https://osf.io/zcdfq

Table 2: Contents used in Singmann et al. (2016) experiments. *Note:* This is a translation of the contents in English as provided by the authors. The experiment has been conducted in German.

Keyword	Content	Disablers	Alternatives
Predator	If a predator is hungry, then it will search for prey.	Few	Few
Balloon	If a balloon is pricked with a needle, then it will pop.	Few	Many
Girl	If a girl has sexual intercourse, then she will be pregnant.	Many	Few
Coke	If a person drinks a lot of coke, then the person will gain weight.	Many	Many

Table 3: Average inference form endorsements for each task in each conditional presentation form from Singmann et al.'s (2016) experimental data. ('Red.' - Reduced Inference, 'Full' - Full Inference, 'Dis' - Disablers, 'Alt' - Alternatives, 'F' - Few, 'M' - Many)

Form	Task	Dis/Alt	MP	AC	DA	MT
Red.	Predator	F/F	91	84	74	81
	Balloon	F/M	89	64	74	81
	Girl	M/F	32	88	85	44
	Coke	M/M	64	52	57	60
Full	Predator	F/F	92	87	80	85
	Balloon	F/M	93	77	76	86
	Girl	M/F	62	87	83	62
	Coke	M/M	78	64	63	73

not ($\neg$), as usual. For simplicity, the symbol $\wedge$ might be omitted and the conjunction would be written by juxtaposition. Additionally, the negation ($\neg A$) would be abbreviated by ($\overline{A}$). The set of possible worlds over Σ will be called Ω, we often use the 1-1 association between worlds and complete conjunctions, that is, conjunctions of literals $\dot{v}_i \in \{v_i, \overline{v}_i\}$ where every variable $V_i \in \Sigma$ appears exactly once. A formula $A \in \mathfrak{L}$ is evaluated under a world ω according to the classical logical rules, that is, $[\![A]\!]_\omega = \mathit{true}$ if and only if $\omega \models A$ if and only if $\omega \in \mathit{Mod}(A)$, that is, ω is an element of the classical models $\mathit{Mod}(A)$ of A. The set of classical consequences of a set of formulas $\mathcal{A} \subseteq \mathfrak{L}$ is $\mathit{Cn}(\mathcal{A}) = \{B | \mathcal{A} \models B\}$. The deductively closed set of formulas which has exactly a subset $\mathcal{W} \subseteq \Omega$ as models is called the *formal theory* of $\mathcal{W}$ and defined as $\mathit{Th}(\mathcal{W}) = \{A \in \mathfrak{L} \mid \omega \models A \text{ for all } \omega \in \mathcal{W}\}$. The material implication "From A it (always) follows that B" is, as usual, equivalent to $\overline{A} \vee B$ and written as $A \Rightarrow B$.

We introduce the binary operator $|$ to obtain the set $(\mathfrak{L}|\mathfrak{L})$ of *conditionals* written as $(B|A)$. Conditionals are three-valued logical entities with the evaluation (DeFinetti 1974)

$$[\![(B|A)]\!]_\omega = \begin{cases} \mathit{true} & \text{iff } \omega \models AB \text{ (verification)} \\ \mathit{false} & \text{iff } \omega \models A\overline{B} \text{ (falsification)} \\ \mathit{undefined} & \text{iff } \omega \models \overline{A} \quad \text{(neutrality).} \end{cases}$$

A *(conditional) knowledge base* is a finite set of conditionals $\Delta = \{(B_1|A_1), \ldots, (B_n|A_n)\} \subseteq (\mathfrak{L} \mid \mathfrak{L})$. To give appropriate semantics to conditionals and knowledge bases, we need richer structures like epistemic states in the sense of (Halpern 2005), most commonly being represented as probability distributions, possibility distributions (Dubois and Prade 2015) or Ordinal Conditional Functions (Spohn 1988; Spohn 2012). A knowledge base is *consistent* if and only if there is (a representation of) an epistemic state that accepts (all conditionals in) the knowledge base.

4 Plausible Reasoning

Similarly to Eichhorn et al. (2018), we will implement non-monotonic inferences by plausibility relations on possible worlds by instantiating preferential models (Makinson 1994) with total preorders resp. Ordinal Conditional Functions (OCF, (Spohn 1988; Spohn 2012)) which we derive from the statistical data of experiments via inference patterns.

4.1 Preferential Inference

For non-monotonic inference and the modeling of epistemic states, total preorders $\preccurlyeq$ on possible worlds expressing plausibility are of crucial importance. If $\omega_1 \preccurlyeq \omega_2$, ω_1 is deemed as at least as plausible as ω_2. Such a preorder can be lifted to the level of formulas by stating that $A \preccurlyeq B$ if for each model of B, there is a model of A that is at least as plausible. As usual, the relations $\prec$ and $\approx$ are derived from $\preccurlyeq$ by $A \prec B$ if and only if $A \preccurlyeq B$ and not $B \preccurlyeq A$, and $A \approx B$ if and only if both $A \preccurlyeq B$ and $B \preccurlyeq A$. Non-monotonic inference can then be easily realized as a form of preferential entailment of high logical quality (Makinson 1994): $A {\mid\!\sim}_{\preccurlyeq} B$ if and only if $AB \prec A\overline{B}$, i.e., from A, B can be plausibly inferred if in the context of A, B is more plausible than $\overline{B}$. Hence total preorders provide convenient epistemic structures for plausible reasoning, and epistemic states Ψ can be represented by such a total preorder $\preccurlyeq_\Psi$. The belief set, i.e., the most plausible beliefs that an agent with epistemic state Ψ holds, is defined to be the set of all formulas which are satisfied by all most plausible worlds: $\mathit{Bel}(\Psi) = \mathit{Th}(\min(\preccurlyeq_\Psi))$, where $\min(\preccurlyeq_\Psi)$ is the set of all minimal worlds according to $\preccurlyeq$. Conditionals can then be integrated smoothly into this reasoning framework by defining $\Psi \models (B|A)$ if and only if $A {\mid\!\sim}_{\preccurlyeq} B$, i.e., conditionals can encode non-monotonic inferences on the object level. We illustrate this with the following example, for more details, we refer to, e.g., Kern-Isberner and Eichhorn (2014).

Example 1. *We illustrate this inference using the 'Balloon' content from Singmann et al.'s (2016) experiments – "If a balloon is pricked with a needle, then it will pop". Let N indicate that a balloon is pricked with a needle (n), or not ($\overline{n}$), and P indicate that the balloon has popped (p), or not ($\overline{p}$). Here the possible worlds are $\{np, n\overline{p}, \overline{n}p, \overline{n}\,\overline{p}\}$. We define* *the epistemic state Ψ to be represented by the preorder*

$$np \approx_\Psi \overline{n}\,\overline{p} \prec_\Psi n\overline{p} \approx_\Psi \overline{n}p.$$

Applying preferential inference we obtain that, for instance, $n \mid\!\sim_{\preccurlyeq} p$ because $np \prec n\overline{p}$, thus $\Psi \models (p|n)$. Here, $\min(\preccurlyeq_\Psi) = \{np, \overline{n}\,\overline{p}\}$*, thus $Bel(\Psi) = Th(\{np, \overline{n}\,\overline{p}\}) = Cn(n \Leftrightarrow p)$.*

4.2 Ordinal Conditional Functions

Ordinal conditional functions (Spohn 2012) are specific implementations of such epistemic states that assign to each level of plausibility a degree of (im)plausibility. Also called a *ranking function*, an Ordinal Conditional Function (OCF, (Spohn 1988; Spohn 2012)) is a function $\kappa : \Omega \to \mathbb{N}_0 \cup \{\infty\}$ that assigns to each world ω an implausibility rank $\kappa(\omega)$ such that the higher $\kappa(\omega)$, the less plausible ω is. Given a normalization constraint, there are worlds that are maximally plausible, that is, the pre-image $\kappa^{-1}(0)$ cannot be empty. The rank of a formula $A \in \mathfrak{L}$ is the minimal rank of all worlds that satisfy A, and the rank of a conditional is the rank of the verification of the conditional normalized by the rank of the premise, so we have $\kappa(A) = \min\{\kappa(\omega)|\omega \models A\}$ and $\kappa(B|A) = \kappa(AB) - \kappa(A)$.

A ranking function *accepts* a conditional (written $\kappa \models (B|A)$) if and only if its verification is more plausible than its falsification, and a formula B is κ-inferred from a formula A (written $A \mid\!\sim_\kappa B$) if and only if κ accepts the conditional $(B|A)$, if and only if $\kappa \models (B|A)$, if and only if $\kappa(AB) < \kappa(A\overline{B})$, in accordance with preferential inference as defined above. An OCF is *admissible with respect to a knowledge base* (written $\kappa \models \Delta$) if and only if it accepts all conditionals in Δ.

Example 2. *We continue Example 1 to illustrate OCF. A ranking function that induces $\preccurlyeq_\Psi$ is the OCF $\kappa(np) = \kappa(\overline{n}\,\overline{p}) = 0$, $\kappa(n\overline{p}) = \kappa(n\overline{p}) = 1$. With κ we have $\kappa(np) < \kappa(n\overline{p})$, and thus $\kappa \models (p|n)$ and also $n \mid\!\sim_\kappa p$.*

5 Inference Patterns

Eichhorn et al. (2018) proposed an approach to combine all four inference rules into tuples called *inference patterns* in order to classify psychological findings. Their initial point are the inference rules and their respective inferences as shown in Table 4, followed by a formalization of what it means that it is *plausible* to draw conclusions according to these rules, as (re-)introduced in the following.

Definition 1 (Inference Pattern). *An* inference pattern ϱ *is a 4-tuple of inference rules that for each inference rule MP, MT, AC, and DA indicates whether the rule is used (positive rule, e.g., MP) or not used (negated rule, e.g., ¬MP) in an inference scenario. The set of all 16 inference patterns is called $\mathcal{R}$.*

To draw plausible inferences with respect to an inference rule, a plausibility preorder $\preccurlyeq$ has to be defined on the set of worlds, see Section 4. For instance, we have MP if any only if for a statement "If A then B" the inference $A \mid\!\sim B$ is drawn. This is the case if and only if the worlds are ordered such that for each world violating the statement (each $\omega' \models A\overline{B}$) there is a world that verifies the statement ($\omega \models AB$) which is more plausible than ω' ($\omega \prec \omega'$), that is, if and only if $AB \prec A\overline{B}$. Table 5 gives all of the plausibility constraints which are equivalent to using the inference rules.

Table 4: Overview of the inferences drawn or not drawn from "From A it (usually) follows that B" with respect to application of the inference rules.

Rule	Inference	Rule	Inference
MP	$A \mid\!\sim B$	¬MP	$A \not\mid\!\sim B$
MT	$\overline{B} \mid\!\sim \overline{A}$	¬MT	$\overline{B} \not\mid\!\sim \overline{A}$
AC	$B \mid\!\sim A$	¬AC	$B \not\mid\!\sim A$
DA	$\overline{A} \mid\!\sim \overline{B}$	¬DA	$\overline{A} \not\mid\!\sim \overline{B}$

Table 5: Constraints on the plausibility relation on worlds in order to satisfy inference rules.

Rule	Plausibility constraint	Rule	Plausibility constraint
MP	$A\,B \prec A\,\overline{B}$	¬MP	$A\,\overline{B} \preccurlyeq A\,B$
MT	$\overline{A}\,\overline{B} \prec A\,\overline{B}$	¬MT	$A\,\overline{B} \preccurlyeq \overline{A}\,\overline{B}$
AC	$AB \prec \overline{A}\,B$	¬AC	$\overline{A}\,B \preccurlyeq A\,B$
DA	$\overline{A}\,\overline{B} \prec \overline{A}\,B$	¬DA	$\overline{A}\,B \preccurlyeq \overline{A}\,\overline{B}$

To satisfy an inference pattern, the plausibility relation has to satisfy each of the constraints given in Table 5. So each reasoning pattern $\varrho \in \mathcal{R}$ imposes a set of constraints on the plausibility relation, which in the following is called $\mathcal{C}(\varrho)$; $\mathcal{C}(\varrho)$ is *satisfiable* if and only if there is a plausibility relation $\prec$ and hence an epistemic state that satisfies all constraints in $\mathcal{C}(\varrho)$.

For instance, to satisfy the pattern (MP, MT, ¬AC, DA) (which occurs as an individual pattern in the balloon example, see Table 9), the worlds have to be ordered such that all four constraints given in Table 6 are satisfied.

If for a given pattern ϱ, there is a plausibility relation $\preccurlyeq$ that satisfies $\mathcal{C}(\varrho)$, that is, there is a total preorder on the worlds which is in accordance with plausible reasoning, ϱ can be deemed to be rational. Therefore, we call an inference pattern *rational* if and only if there is a plausibility relation $\preceq$ that satisfies the inference pattern. Note that, similar to more classical approaches, rationality is understood in terms of compliance with logic. However, here we use non-monotonic logics and its preferential models as norms for rational reasoning behavior.

Inspecting all $\varrho \in \mathcal{R}$ we obtain that only two patterns, namely (MP, ¬MT, ¬AC, DA) and (¬MP, MT, AC, ¬DA), are irrational: For the first pattern, the constraints impose the unrealizable ordering $\overline{A}\,\overline{B} \prec \overline{A}B \preccurlyeq AB \prec A\overline{B} \preccurlyeq \overline{A}\,\overline{B}$, for the second, the constraints impose the unrealizable ordering $\overline{A}\,\overline{B} \prec A\overline{B} \preccurlyeq AB \prec \overline{A}B \preccurlyeq \overline{A}\,\overline{B}$.

Eichhorn et al. (2018) used this approach to analyze the combination of inference rules in an experiment. We will perform a similar analysis on the experimental data presented in Section 2, however, since now we are dealing with probabilistic endorsements, in order to enable such analysis,

Table 6: Constraints for the inference pattern $(\mathrm{MP}, \mathrm{MT}, \neg\mathrm{AC}, \mathrm{DA})$.

	$\{AB \prec A\overline{B}, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B \preccurlyeq AB, \overline{A}\,\overline{B} \prec \overline{A}B\}$
yields	$\overline{A}\,\overline{B} \prec \overline{A}B \preccurlyeq AB \prec A\overline{B}$

we will propose an enhancement of the inference patterns in the following section.

6 Inference Patterns with Endorsement

A probabilistic endorsement of a conclusion describes the degree of subjective belief an individual has in that world, in the range 0-100%. We consider endorsements that are $\geq 50\%$ as $true$, i.e., the inference form has been applied and otherwise $false$– the inference form has not been applied.

Definition 2. *For a conditional* $(B|A)$ *and an ordinal conditional function* κ, *we say* κ *accepts* $(B|A)$ *with strength* s *if* $\kappa \models (B|A)$ *and* $s = \kappa(A\overline{B}) - \kappa(AB)$. *We call* $s = s_\kappa((B|A))$ *the* κ-strength *of* $(B|A)$.

Definition 3. *An inference rule* r *is more* endorsed *than an inference rule* r' *with respect to a ranking function* κ *if* $s_\kappa(\varphi_r) > s_\kappa(\varphi_{r'})$ *holds for the associated conditionals* $\varphi_r, \varphi_{r'}$.

Following Definition 3, we examined the relationships between endorsements in Singmann et al.'s (2016) experimental data on both aggregate *and* individual level. We also specified a *difference tolerance* of 5, meaning that two endorsements (in the range 0-100%) will be considered as equal if the difference between them is $\leq 5\%$.

Example 3 (Ranking of endorsements)**.** *Let us consider the task 'Girl' in reduced inference (Table 3). Given the endorsements MP: 32%, MT: 44%, AC: 88%, DA: 85%, the respective rankings would then be ¬MP ≻ ¬ MT (false, not applied) and AC = DA (true, applied). Additionally, the corresponding (non-enhanced) inference pattern would be* $(\neg MP, \neg MT, AC, DA)$.

Note that for negated inference rules, the preorder $\succeq$ is reversed, i.e., in the example above, both MP and MT are not applied (i.e. false), but the endorsement of MP is lower than MT. This results in MP ≻ MT.

The derived rankings of the average inference form endorsements are presented in Table 7. With them, we can now *enhance* the inference patterns from Eichhorn et al. (2018) by statements about the strengths of inference rules.

Definition 4 (Extended Inference Pattern)**.** *An* extended inference pattern ϱ *is a 4-tuple of inference rules that for each inference rule MP, MT, AC, and DA indicates whether the rule is used (positive rule, e.g., MP) or not used (negated rule, e.g., ¬MP) in an inference scenario, possibly together with statements about the ranking of endorsements of these (negated) inference rules.*

Example 4. *In Ex. 3, we obtain the extended inference pattern* $(\neg MP, \neg MT, AC, DA; \neg MP \succ \neg MT, AC = DA)$. *It shows that MP and MT have not applied, AC and DA have been applied, MP is* less *endorsed than MT, and AC and DA are endorsed equally.*

7 Explaining Human Inferences

In the introduction we have briefly introduced the theory of mental models (Johnson-Laird and Byrne 2002). This theory argues that people represent possibilities (we call them here possible worlds) that can depend on "knowledge, pragmatics, and semantics". As this theory can represent even the case $A\overline{B}$ the question arises, which worlds are preferred over others. The state of art in psychological research implicitly suggests that there are some orders on worlds.

Another psychological experiment (Barrouillet, Grosset, and Lecas 2000) suggests the order $AB \prec \overline{A}\,\overline{B} \prec \overline{A}B$. This has been so far identified experimentally only on the aggregate level (i.e., the mean of answers), but it has not yet been shown if this order holds for the individual reasoner. This is, however, most important as modeling each individual is the preferred goal of cognitive modeling, since models for the aggregate can distort theories (Fisher, Medaglia, and Jeronimus 2018). In the following, we introduce the necessary definitions and analyses to support or reject the claimed order and to analyze the inference patterns.

Definition 5. *A* preferential mental model *is a set of possible worlds together with a total preorder.*

As Eichhorn et al. (2018) explained, inference patterns can be realized by preferential mental models. Now, together with the endorsements, we are able to refine these preferential mental models. For that, we make use of ordinal conditional functions to be able to use arithmetics for the comparisons. However, in order to only make use of arithmetics on an intuitive level, we restrict the exploitation of these comparisons to basic cases. For instance, via ordinal conditional functions, the statement MP ≻ MT translates into $\kappa(A\overline{B}) - \kappa(AB) > \kappa(A\overline{B}) - \kappa(\overline{A}\,\overline{B})$, which is equivalent to $\kappa(AB) < \kappa(\overline{A}\,\overline{B})$. Note that $\kappa(A\overline{B})$ occurs in both differences which allows for an easy comparison by basic arithmetics. In this way, we obtain the following results for qualitative comparisons among the endorsements of inference rules:

$MP \succ MT$	$AB \prec \overline{A}\,\overline{B}$
$MP \succ AC$	$\overline{A}B \prec A\overline{B}$
$MT \succ DA$	$\overline{A}B \prec A\overline{B}$
$AC \succ DA$	$AB \prec \overline{A}\,\overline{B}$

Regarding disablers and alternatives, we translate their influence on the acceptance/endorsement of inference forms into these schemata, so that we are able to identify them in the preferential mental models. The presence of many disablers reduces the degree of belief in the logically valid MP and MT, and alternatives reduce the endorsement of AC and DA (Byrne 1989; Singmann, Klauer, and Beller 2016).

- Few disablers make the antecedent very informative for the consequent, similarly as in classical implications. Therefore, the logically valid MP and MT inference rules should be strong. So, we characterize this scenario by $MP \succeq AC$, which results in $\overline{A}B \preceq A\overline{B}$. Note that $MT \succeq DA$ yields the same constraint.
- Consequently, many disablers are modeled by $A\overline{B} \prec \overline{A}B$.

Table 7: Derived rankings of the average inference form endorsements for each task in each conditional presentation form, corresponding preferential mental models and scenarios. Scenarios that coincide with expected scenarios are marked in bold. In the rankings, inference forms that are True (applied, endorsement ≥ 50) are preceded with a 'T', ones that are False (not applied, endorsement < 50) with a 'F'. The average values of the inference form endorsements are presented in Table 3. ('Red.' - Reduced Inference, 'Full' - Full Inference, 'Dis' - Disablers, 'Alt' - Alternatives, 'Sc.' - Scenario)

Form	Task	Dis / Alt	Ranking of Endorsements	Preferential Mental Model	Sc. (Dis / Alt)
Red.	Predator	Few/Few	T: $MP \succ AC = MT \succ DA$	$AB \prec \overline{A}\,\overline{B} \prec \overline{A}B \prec A\overline{B}$	**Few/Few**
	Balloon	Few/Many	T: $MP \succ MT \succ DA \succ AC$	(no preferential mental model)	
	Girl	Many/Few	T: $AC = DA$; F: $MP \succ MT$	$A\overline{B} \preceq \overline{A}\,\overline{B} \prec AB \prec \overline{A}B$	Many/Many
	Coke	Many/Many	T: $MP = MT \succ DA \succ AC$	$\overline{A}\,\overline{B} \prec AB \prec \overline{A}B \prec A\overline{B}$	Few/Many
Full	Predator	F / F	T: $MP = AC \succ DA$ $AC = MT; MP \succ MT$	$AB \succ \overline{A}\,\overline{B} \succ A\overline{B}, \overline{A}B$	**Few/Few**
	Balloon	Few/Many	T: $MP \succ MT \succ AC = DA$	$AB \prec \overline{A}\,\overline{B} \prec \overline{A}B \prec A\overline{B}$	Few/Few
	Girl	Many/Few	T: $AC = DA \succ MP = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B} \prec \overline{A}B$	**Many/Few**
	Coke	Many/Many	T: $MP = MT \succ AC = DA$	$AB, \overline{A}\,\overline{B} \prec \overline{A}B \prec A\overline{B}$	Few/Few

Table 8: Number of individuals out of 199 that have the same ranking that is found on the aggregate level (shown in Table 7).

Task	# Individuals	
	Reduced	Full
Predator	3 (1.51%)	0 (0.0%)
Balloon	3 (1.51%)	5 (2.51%)
Girl	27 (13.57%)	11 (5.53%)
Coke	0 (0.0%)	4 (2.01%)

- Few alternatives make the antecedent very plausible when observing the consequent, so particularly AC should be strong. We model this via $AC \succeq DA$ which gives us $AB \preceq \overline{A}\,\overline{B}$.
- Consequently, many alternatives are modeled by $\overline{A}\,\overline{B} \prec AB$.

Please keep in mind that artifacts that contradict these schematic classifications may arise due to the general plausibility of A and B in the background knowledge of the individuals.

Aside from the inference patterns, Table 7 also presents the corresponding preferential mental models and the respective scenarios.

If we look at the aggregate case (Table 7) for the reduced inference presentation form, the scenarios respective to the induced preferential mental models do not necessarily correspond to the original quantification of disablers and alternatives associated with the tasks' contents. E.g., the 'Coke' task, leads to the question whether alternatives are more influential than disablers when a content has 'Many' of both associated with it. Additionally, when looking into the 'Balloon' task, the inference pattern derived from aggregate data is inconsistent, i.e. it induced no preferential mental models. This may happen due to too divergent views of the individuals. In their analysis, Singmann et al. (2016) showed that when presented with a reduced inference, individuals tend to rely more on their background knowledge and have a stronger influence by the corresponding disablers and alternatives, in contrast to the full inference. That effect can also be seen here, as in the full inference presentation form, the scenarios identify 'Few' disablers and alternatives even when there are 'Many', meaning that individuals were not integrating their background knowledge as much. An exception is the 'Girl' content, where disablers seem to be exceptionally influential, which is also visible when looking at the average inference forms endorsements (Table 3).

However, to which extent do these findings on the aggregate level hold for the individual reasoner? We looked into each participant's endorsements and found out for how many participants the ranking derived from the aggregate data holds. The numbers are shown in Table 8. We can immediately see that the number of individuals that are captured by the aggregate rankings is exceptionally low. This was an expected outcome (Fisher, Medaglia, and Jeronimus 2018), and strongly confirms our need for individual analysis. Therefore, we also derived the rankings, the induced preferential models and scenarios for each individual separately.

The different rankings found for at least 5% of participants are shown in Table 9. It can be seen that individuals are not divided in only a few largely populated groups, but there are multiple different rankings found across the 199 participants. For each pattern the size of the participant group is also presented in the table. In most cases, the scenarios identify the same quantities of disablers and alternatives as originally associated with the respective tasks. This is extremely important, as we can see that groups of individuals *do* interpret the conditionals as intended. If we only focused on the aggregate analysis, we would most likely dismiss the tasks in the reduced inference case due to the lack of correspondence between derived scenarios and expected ones.

Additionally, the inference patterns whose preferential mental models do not identify the same quantities are still present among a larger group of participants, which points to possible differences in conditional interpretation and different knowledge bases. E.g., the third most frequent pattern

Table 9: Individual rankings, preferential mental models and corresponding scenarios for all contents in both conditional presentation forms. The rankings and preferential models are listed in descending order of the frequencies of the appertaining extended inference patterns, i.e., the first line corresponds to the most frequent extended inference pattern. Only inference patterns that were found for at least 5% of participants are taken into consideration and the exact number of individuals for each ranking is presented. Scenarios that correspond to the expected scenarios are marked in bold. In the rankings, inference forms that are True (applied, endorsement ≥ 50) are preceded with a 'T', ones that are False (not applied, endorsement < 50) with a 'F'. ('Red.' - Reduced Inference, 'Full' - Full Inference, 'Dis' - Disablers, 'Alt' - Alternatives, 'Ind.' - Individuals)

Task	Dis / Alt	Form	Ranking	Preferential Mental Model	Scenario (Dis / Alt)	# Ind.
Predator	Few/Few	Red. Full	T: $MP = AC = DA = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B$	**Few/Few**	49 59
Balloon	Few/Many	Red.	T: $DA = MP = MT \succ AC$	$\overline{A}\,\overline{B} \prec AB \prec A\overline{B}, \overline{A}B$	**Few/Many**	16
			T: $MP = MT \succ DA$; F: AC	$\overline{AB} \prec \overline{A}B \preceq AB \prec A\overline{B}$	**Few/Many**	15
			T: $MP = AC = DA = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B$	Few/Few	12
			T: $MP = MT \succ DA \succ AC$	$\overline{A}\,\overline{B} \prec AB \prec \overline{A}B \prec A\overline{B}$	**Few/Many**	10
		Full	T: $MP = AC = DA = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B$	Few/Few	46
			T: $MP = MT \succ DA$; F: AC	$\overline{A}\,\overline{B} \prec \overline{A}B \preceq AB \prec A\overline{B}$	**Few/Many**	13
			T: $MP = MT \succ AC = DA$	$AB, \overline{A}\,\overline{B} \prec \overline{A}B \prec A\overline{B}$	**Few/Many**	13
			T: $DA = MP = MT \succ AC$	$\overline{A}\,\overline{B} \prec AB \prec A\overline{B}, \overline{A}B$	**Few/Many**	12
Girl	Many/Few	Red.	T: $AC = DA$; F: $MP \succ MT$	$A\overline{B} \preceq \overline{A}\,\overline{B} \prec AB \prec \overline{A}B$	Many/Many	27
			T: $AC = DA$; F: $MP = MT$	$A\overline{B} \preceq AB, \overline{A}\,\overline{B} \prec \overline{A}B$	**Many/Few**	19
			T: $AC = DA \succ MT$; F: MP	$\overline{A}\,\overline{B} \prec A\overline{B} \preceq AB \prec \overline{A}B$	Many/Many	17
			T: $AC = DA$; F: $MT \succ MP$	$A\overline{B} \preceq AB \prec \overline{A}\,\overline{B} \prec \overline{A}B$	**Many/Few**	15
			T: $AC = DA \succ MP$; F: MT	$AB \prec A\overline{B} \preceq \overline{A}\,\overline{B} \preceq \overline{A}B$	**Many/Few**	12
			T: $AC = DA \succ MP = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B} \prec \overline{A}B$	**Many/Few**	11
		Full	T: $MP = AC = DA = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B$	Few/Few	29
			T: $AC = DA$; F: $MP = MT$	$A\overline{B} \preceq AB, \overline{A}\,\overline{B} \prec \overline{A}B$	**Many/Few**	13
			T: $AC = DA \succ MP = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B} \prec \overline{A}B$	**Many/Few**	11
			T: $AC = DA$; F: $MP \succ MT$	$A\overline{B} \preceq \overline{A}\,\overline{B} \prec AB \prec \overline{A}B$	Many/Many	10
			T: $AC = DA \succ MP$; F: MT	$AB \prec A\overline{B} \preceq \overline{A}\,\overline{B} \preceq \overline{A}B$	**Many/Few**	10
			T: $AC = DA \succ MT$; F: MP	$\overline{A}\,\overline{B} \prec A\overline{B} \preceq AB \prec \overline{A}B$	Many/Many	10
Coke	Many/Many	Red.	(no consistent ranking found)	–	–	–
		Full	T: $MP = AC = DA = MT$	$AB, \overline{A}\,\overline{B} \prec A\overline{B}, \overline{A}B$	Few/Few	23
			T: $MP = MT = AC \succ DA$	$AB, \overline{A}\,\overline{B} \prec \overline{A}B \prec A\overline{B}$	Few/Few	11

for the 'Balloon' content in the reduced inference case does not suggest that those individuals were able to incorporate 'Many' alternatives when reasoning.

When comparing reduced inference with full inference, it can be seen that patterns that induce a 'Few/Few' scenario even if there are 'Many' disablers and alternatives present are rather frequent, meaning that the influence of background knowledge has been suppressed. However, there is still a significant amount of individuals who nevertheless successfully integrate information about the disablers/alternatives when reasoning. A conclusion about the influence of the full inference in contrast to the reduced one on the effect of disablers and alternatives should absolutely *not* be derived based on aggregate data. As we show here, individuals and their interpretations *differ*.

It is also noteworthy that even though there are different inference form endorsement combinations present among individuals, after deriving the corresponding preferential mental models, they all suggest the same interpretation. The presence of a certain amount of disablers and alternatives can be modeled and endorsed in various ways by different people.

8 Discussion and Conclusion

We followed the inference evaluation approach by applying logic based on conditionals and plausible reasoning proposed by Eichhorn et al. (2018) and extended it towards *probabilistic endorsements*. We do not only take into consideration whether an inference form has been applied or not, but also look into the relationships between the subjective degrees of belief in said inference forms for various contents. Using OCFs (Spohn 2012) a plausibility relation on possible worlds was defined in order to obtain a preferential entailment.

The beauty of this interdisciplinary field is that different formalisms can be used in analysis and with that we can get insight into human reasoning from many different perspectives which can be joined to get an even better understanding of reasoning processes. Given the preferential character of probabilistic endorsements, ranks are a natural approach to consider. Our contribution is to show how ranking functions can be applied to probabilistic conditional reasoning experimental data and what we can learn from them.

The extended inference patterns reveal in an abstract way how people *understood* the task they were given. Our focus was on tasks with a varying amount of disablers and alternatives – events that make humans diverge from logical reasoning. They are especially influential in a reduced inference presentation form, when individuals are not bound by a conditional rule but can rather integrate their personal background knowledge on a higher level. Our approach is flexible enough to be able to show the impact of disablers and alternatives. We are already familiar with the fact that humans deviate from classical logic when reasoning, so shifting the focus of research to everyday contents is important. Understanding how background knowledge, personal and cultural differences influence reasoning is of our interest.

As illustrated by the large variety in the derived inference patterns we can see the effect of individual differences (e.g. substantial cultural differences) and how human reasoning can be very diverse. Therefore, an aggregate analysis approach might not always be appropriate to get a better understanding of inference mechanisms, but the individual differences play a big role and should be taken into consideration.

Additionally, by performing individual analysis we can also learn whether certain experimental content managed to achieve its goal. For example, the 'Coke' content is supposed to have 'Many' disablers and alternatives associated with it, which as true as it might be, it does not seem to be understood that way by the participants. In Table 9 we see that in the reduced inference case, where the background knowledge should be dominating, no consistent pattern was found. That means that there was not a single group of individuals formed by at least 5% of the total participants that understood the task in the same way, which points to the need for reconsideration of the chosen content.

The inference pattern derived from aggregate data for the 'Balloon' task in the reduced inference case is inconsistent. Contradictory patterns show irrationality when found on an individual level, e.g. the three individual participants that had the same endorsement ranking reasoned irrationally, which is, of course, a common occasional human trait, and our approach can account for this! However, having found an inconsistent pattern on the aggregate level means that the individuals' perspectives and interpretation are diverging too much in order to be aggregated consistently. This supports the idea that an *individual* analysis approach is necessary. Moreover, it also indicates a potential requirement to reconsider whether such content is suitable to test human reasoning. Naturally, in order to determine proper task contents a significant increase in various experimental data is required.

To conclude, we analyzed the same experimental data on two levels – aggregate and individual. In many ways we showed that the focus undoubtedly needs to be switched to the *individual*. Humans are diverse, their personal experiences lead to diverging background knowledge and interpretation abilities. In order to make a larger leap forward towards understanding the human reasoning processes, these differences have to be taken into consideration and modeling approaches should be able to account for the deviations between individuals. We found preferential mental models on the individual level whose interpretations give us insight into how the experimental content manipulation affected (or not) the participants' reasoning. Generally, the fact that many different individual inference patterns induce the same mental models points to a significant strength of our approach – using mental models is more fundamental than representation forms that heavily rely on statistics.

The next step in this work would be to look more into the specifics of the relevant background knowledge and its influence on reasoning. Additionally, an even larger focus on the individual would be of interest. For instance, how do the individual's subjective believes and inference mechanism change between different contents or conditional presentation forms? The theoretical foundation of our approach would allow for gaining more insight into such questions and aid in getting a better understanding of the individual reasoning processes.

References

Barrouillet, P.; Grosset, N.; and Lecas, J.-F. 2000. Conditional reasoning by mental models: Chronometric and developmental evidence. *Cognition* 75(3):237–266.

Byrne, R. M. 1989. Suppressing valid inferences with conditionals. *Cognition* 31:61–83.

DeFinetti, B. 1974. *Theory of Probability: A Critical Introductory Treatment (Translated by A. Machi and A. Smith)*, volume 1 & 2. UK: Wiley.

Dubois, D., and Prade, H. 2015. Possibility theory and its applications: Where do we stand? In Kacprzyk, J., and Pedrycz, W., eds., *Springer Handbook of Computational Intelligence*. Berlin, DE: Springer. 31–60.

Eichhorn, C.; Kern-Isberner, G.; and Ragni, M. 2018. Rational inference patterns based on conditional logic. In *AAAI Conference on Artificial Intelligence*, volume 32, 1827–1834.

Fisher, A. J.; Medaglia, J. D.; and Jeronimus, B. F. 2018. Lack of group-to-individual generalizability is a threat to human subjects research. *Proceedings of the National Academy of Sciences* 115(27):E6106–E6115.

Halpern, J. Y. 2005. *Reasoning About Uncertainty*. Cambridge, MA, USA: MIT Press.

Johnson-Laird, P. N., and Byrne, R. M. 1991. *Deduction*. Lawrence Erlbaum Associates, Inc.

Johnson-Laird, P. N., and Byrne, R. M. 2002. Conditionals: a theory of meaning, pragmatics, and inference. *Psychological review* 109(4):646.

Kern-Isberner, G., and Eichhorn, C. 2014. Structural Inference from Conditional Knowledge Bases. In Unterhuber, M., and Schurz, G., eds., *Logic and Probability: Reasoning in Uncertain Environments*, number 102 (4) in Studia Logica. Dordrecht, NL: Springer Science+Business Media. 751–769.

Makinson, D. 1994. General Patterns in Nonmonotonic Reasoning, vol. 3. In Gabbay, D. M.; Hogger, C. J.; and Robinson, J. A., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford Uni. Press. 35–110.

Ragni, M.; Dames, H.; and Johnson-Laird, P. N. 2019. A meta-analysis of conditional reasoning. In Stewart, T. C., ed., *Proceedings of the 17th International Conference on Cognitive Modeling*, 151–156. Waterloo, Canada: University of Waterloo: https://iccm-conference.neocities.org/2019/proceedings/index.html.

Ragni, M.; Kola, I.; and Johnson-Laird, P. N. 2018. On selecting evidence to test hypotheses: a theory of selection tasks. *Psychological Bulletin* 144(8):779–796.

Singmann, H.; Klauer, K. C.; and Beller, S. 2016. Probabilistic conditional reasoning: Disentangling form and content with the dual-source model. *Cognitive Psychology* 88:61.

Spohn, W. 1988. Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. In *Causation in Decision, Belief Change and Statistics: Proceedings of the Irvine Conference on Probability and Causation*, volume 42 of *The Western Ontario Series in Philosophy of Science*, 105–134. Dordrecht, NL: Springer Science+Business Media.

Spohn, W. 2012. *The Laws of Belief: Ranking Theory and Its Philosophical Applications*. Oxford, UK: Oxford University Press.

From Common Sense Reasoning to Neural Network Models through Multiple Preferences: an Overview

Laura Giordano[1] , **Valentina Gliozzi**[2] , **Daniele Theseider Dupré**[1]

[1] DISIT - Università del Piemonte Orientale, Italy

[2] Center for Logic, Language and Cognition & Dipartimento di Informatica, Università di Torino, Italy

{laura.giordano, dtd}@uniupo.it, valentina.gliozzi@unito.it

Abstract

In this paper we discuss the relationships between conditional and preferential logics and neural network models, based on a multi-preferential semantics. We propose a concept-wise multipreference semantics, recently introduced for defeasible description logics to take into account preferences with respect to different concepts, as a tool for providing a semantic interpretation to neural network models. This approach has been explored both for unsupervised neural network models (Self-Organising Maps) and for supervised ones (Multilayer Perceptrons), and we expect that the same approach might be extended to other neural network models. It allows for logical properties of the network to be checked (by model checking) over an interpretation capturing the input-output behavior of the network. For Multilayer Perceptrons, the deep network itself can be regarded as a conditional knowledge base, in which synaptic connections correspond to weighted conditionals. The paper describes the general approach, through the cases of Self-Organising Maps and Multilayer Perceptrons, and discusses some open issues and perspectives.

1 Introduction

Preferential approaches (Kraus, Lehmann, and Magidor 1990; Pearl 1990; Lehmann and Magidor 1992) to common sense reasoning, having their roots in conditional logics (Lewis 1973; Nute 1980), have been recently extended to description logics, to deal with inheritance with exceptions in ontologies, allowing for non-strict forms of inclusions, called *typicality or defeasible inclusions* (namely, conditionals), with different preferential semantics (Giordano et al. 2007; Britz, Heidema, and Meyer 2008) and closure constructions (Casini and Straccia 2010; Casini et al. 2013; Giordano et al. 2015; Pensel and Turhan 2018), allowing for defeasible or typicality inclusions, e.g., of the form $\mathbf{T}(C) \sqsubseteq D$, meaning "the typical Cs are Ds" or "normally Cs are Ds", corresponding, in the propositional case, to the conditionals $C \mathrel{|\!\sim} D$ in Kraus, Lehmann and Magidor's (KLM) preferential approach (1990; 1992). Description logics allow for a limited first-order language. A first-order extension of system Z has also been explored by Bierle et al. (2017).

In this paper we consider a "concept-wise" multi-preferential semantics, recently introduced by Giordano and Theseider Dupré (2020a) to capture preferences with respect to different aspects (concepts) in ranked $\mathcal{EL}$ knowledge bases, and describe how it has been used as a semantics for some neural network models. We have considered both an unsupervised model, Self-Organising Maps, and a supervised one, Multilayer Perceptrons.

Self-organising maps (SOMs) are psychologically and biologically plausible neural network models (Kohonen, Schroeder, and Huang 2001) that can learn after limited exposure to positive category examples, without need of contrastive information. They have been proposed as possible candidates to explain the psychological mechanisms underlying category generalisation. Multilayer Perceptrons (MLPs) (Haykin 1999) are deep networks. Learning algorithms in the two cases are quite different but, in this work, we only aim to capture, through a semantic interpretation, the behavior of the network resulting after training and not to model learning. We will see that this can be accomplished in both cases in a similar way, based on a multi-preferential semantics.

The result of the training phase is represented very differently in the two models: for SOMs it is given by a set of units spatially organized in a grid (where each unit u in the map is associated with a weight vector w_u of the same dimensionality as the input vectors); for MLPs, as a result of training, the weights of the synaptic connections have been learned. In both cases, considering the domain of all input stimuli presented to the network during training (or in the generalization phase), one can build a semantic interpretation describing the input-output behavior of the network as a multi-preference interpretation, where preferences are associated to concepts. For SOMs, the learned categories are regarded as concepts $C_1, \ldots, C_n$ so that a preference relation (over the domain of input stimuli) is associated to each category. In case of MLPs, each neuron in the deep network (including hidden neurons) can be associated to a concept and a preference relation can be associated to it.

In both cases, the preferential model resulting from the network after training describes the input-output behavior of the network on the input stimuli considered, and the preference relations define a notion of typicality (with respect to different concepts/categories) on the domain of input stimuli. For instance, given two input stimuli x and y, the model can assign to x a degree of typicality which is higher than the degree of typicality of y with respect to some category $Horse$, so that x is regarded as a being more typical than y as a horse ($x <_{Horse} y$), while vice-versa y can be regarded as a being more typical than x as a zebra ($y <_{Zebra} x$). The preferential interpretation can be used for checking properties like: are the instances of a category C_1 also instances of category C_2? Are typical instances of a category C_1 also instances of category C_2? This verification can be done by *model-checking* given multipreference interpretation describing the input-output behavior of the network (Giordano, Gliozzi, and Theseider Dupré 2021).

This kind of construction establishes a strong relationship between the logics of commonsense reasoning and the neural network models, as the first ones are able to reason about the properties of the second ones. The relationship can be made even stronger in some cases, e.g., for MLPs, when the neural network itself can be seen as a conditional knowledge base. In (Giordano and Theseider Dupré 2021b), the concept-wise multipreference semantics has been adapted to deal with weighted knowledge bases, where typicality inclusions have a weight, a real (positive or negative) number, representing the plausibility of the typicality inclusions. It has been proven that Multilayer Perceptrons can be regarded as weighted conditional knowledge bases under a fuzzy extension of the multipreference semantics. The multipreference interpretation which can be built over the set of input stimuli to describe the input-output behavior of the deep network can be proven to be a coherent fuzzy multipreference model of such a knowledge base (under some condition on the activation functions).

This approach raises several issues, from the standpoint of knowledge representation, from the standpoint of neuro-symbolic integration, as well as from the standpoint of explainable AI (Adadi and Berrada 2018; Guidotti et al. 2019; Arrieta et al. 2020). We will discuss some of these issues in the paper after describing the approach in some detail.

2 A concept-wise multi-preference semantics

In this section we shortly describe an extension of $\mathcal{ALC}$ with typicality based on the same language as the typicality logics (Giordano et al. 2007; Giordano et al. 2015), but on a different concept-wise multipreference semantics, first introduced for $\mathcal{EL}^{+}_{\bot}$ (Giordano and Theseider Dupré 2020a).

We consider the description logic $\mathcal{ALC}$. Let N_C be a set of concept names, N_R a set of role names and N_I a set of individual names. The set of $\mathcal{ALC}$ *concepts* can be defined as follows: $C \;:= A \mid \top \mid \bot \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C$, where $a \in N_I$, $A \in N_C$ and $r \in N_R$. A knowledge base (KB) K is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox. The TBox $\mathcal{T}$ is a set of *concept inclusions* (or subsumptions) of the form $C \sqsubseteq D$, where C, D are concepts. The ABox $\mathcal{A}$ is a set of assertions of the form $C(a)$ and $r(a, b)$ where C is a concept, $r \in N_R$, and $a, b \in N_I$.

In addition to standard $\mathcal{ALC}$ inclusions $C \sqsubseteq D$ (called *strict* inclusions in the following), the TBox $\mathcal{T}$ also contains typicality inclusions of the form $\mathbf{T}(C) \sqsubseteq D$, where C and D are $\mathcal{ALC}$ concepts and $\mathbf{T}$ is a new concept constructor ($\mathbf{T}(C)$ is called a typicality concept). A typicality inclusion $\mathbf{T}(C) \sqsubseteq D$ means that "typical Cs are Ds" or "normally Cs are Ds" and corresponds to a conditional implication $C \mathrel{|\!\sim} D$ in Kraus, Lehmann and Magidor's (KLM) preferential approach (1990; 1992). Such inclusions are defeasible, i.e., admit exceptions, while strict inclusions must be satisfied by all domain elements.

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a set of distinguished $\mathcal{ALC}$ concepts. For each concept $C_i \in \mathcal{C}$, we introduce a modular preference relation $<_{C_i}$ which describes the preference among domain elements with respect to C_i. Each preference relation $<_{C_i}$ has the same properties of preference relations in KLM-style ranked interpretations (Lehmann and Magidor 1992), i.e., it is a modular and well-founded strict partial order (an irreflexive and transitive relation), where: $<_{C_i}$ is *well-founded* if, for all $S \subseteq \Delta$, if $S \neq \emptyset$, then $min_{<_{C_i}}(S) \neq \emptyset$; and $<_{C_i}$ is *modular* if, for all $x, y, z \in \Delta$, if $x <_{C_j} y$ then ($x <_{C_j} z$ or $z <_{C_j} y$).

Definition 1. *A* multipreference interpretation *is a tuple* $\mathcal{M} = \langle \Delta, <_{C_1}, \ldots, <_{C_k}, \cdot^I \rangle$*, where:*

(a) Δ *is a non-empty domain;*

(b) $<_{C_i}$ *is an irreflexive, transitive, well-founded and modular relation over* Δ*;*

(c) $\cdot^I$ *is an interpretation function, as in an $\mathcal{ALC}$ interpretation, that maps each concept name $C \in N_C$ to a set $C^I \subseteq \Delta$, each role name $r \in N_R$ to a binary relation $r^I \subseteq \Delta \times \Delta$, and each individual name $a \in N_I$ to an element $a^I \in \Delta$. It is extended to complex concepts as usual: $\top^I = \Delta$, $\bot^I = \emptyset$, $(\neg C)^I = \Delta \backslash C^I$, $(C \sqcap D)^I = C^I \cap D^I$ and $(C \sqcup D)^I = C^I \cup D^I$, $(\exists r.C)^I = \{x \in \Delta \mid \exists y.(x, y) \in r^I$ and $y \in C^I\}$ and $(\forall r.C)^I = \{x \in \Delta \mid \forall y.(x, y) \in r^I \rightarrow y \in C^I\}$.*

The preference relation $<_{C_i}$ allows the set of prototypical C_i-elements to be defined as the C_i-elements which are minimal with respect to $<_{C_i}$, i.e., $min_{<_{C_i}}(C_i^I)$. As a consequence, the multipreference interpretation above is able to single out the typical C_i-elements, for all distinguished concepts $C_i \in \mathcal{C}$.

The multipreference structures above are at the basis of the semantics for ranked $\mathcal{EL}$ knowledge bases (Giordano and Theseider Dupré 2020a), which have been inspired by Brewka's framework of basic preference descriptions (Brewka 2004). While we refer to (Giordano and Theseider Dupré 2020a) for the construction of the preference relations $<_{C_i}$'s from a ranked knowledge base K, in the following we shortly recall the notion of concept-wise multipreference interpretation which can be obtained by *combining* the preference relations $<_{C_i}$ into a global preference relation $<$. This is needed for reasoning about typicality for arbitrary $\mathcal{ALC}$ concepts C, which do not belong to the set of distinguished concepts $\mathcal{C}$. For instance, we may want to verify whether typical employed students are young, or whether they have a boss, starting from a ranked KB containing inclusions $\mathbf{T}(\mathit{Stud}) \sqsubseteq \mathit{Young}$, $\mathbf{T}(\mathit{Emp}) \sqsubseteq \mathit{Has\_Boss}$, $\mathbf{T}(\mathit{Emp}) \sqsubseteq \mathit{NonYoung}$, and $\mathit{Young} \sqcap \mathit{NonYoung} \sqsubseteq \bot$. To answer the query above both preference relations $<_{\mathit{Emp}}$ and $<_{\mathit{Stud}}$ are relevant, and they might be conflicting as, for instance, Tom is more typical than Bob as a student ($\mathit{tom} <_{\mathit{Stud}} \mathit{bob}$), but more exceptional as an employee ($\mathit{bob} <_{\mathit{Emp}} \mathit{tom}$). By *combining* the preference relations $<_{C_i}$ into a single *global preference* relation $<$ we can exploit $<$ for interpreting the typicality operator, which may be applied to arbitrary concepts, and verify, for instance, whether $\mathbf{T}(Stud \sqcap Emp) \sqsubseteq Has\_Boss$.

A natural definition of the notion of global preference $<$ exploits the Pareto combination of the relations $<_{C_1}, \ldots, <_{C_k}$, as follows:

$$x < y \text{ iff } \quad (i)\ x <_{C_i} y, \text{ for some } C_i \in \mathcal{C}, \text{ and} \\ (ii) \text{ for all } C_j \in \mathcal{C},\ x \leq_{C_j} y$$

where $\leq_{C_i}$ is the non-strict preference relation associated with $<_{C_i}$ ($\leq_{C_i}$ is a total preorder). A slightly more sophisticated notion of preference combination, which exploits a modified Pareto condition taking into account the specificity relation among concepts (such as, for instance, the fact that concept $PhdStudent$ is more specific than concept $Student$), has been considered for ranked knowledge bases (Giordano and Theseider Dupré 2020a) to allow a form of overriding which relates to (Bonatti et al. 2015).

The addition of the global preference relation allows for defining a notion of *concept-wise multipreference interpretation* $\mathcal{M} = \langle \Delta, <_{C_1}, \ldots, <_{C_k}, <, \cdot^I \rangle$, where a typicality concept $\mathbf{T}(C)$ is interpreted as the set of the $<$-minimal C elements, i.e., $(\mathbf{T}(C))^I = min_<(C^I)$, where $Min_<(S) = \{u : u \in S \text{ and } \nexists z \in S \text{ s.t. } z < u\}$.

The notions of cw^m-model of a ranked $\mathcal{EL}^+_\bot$ knowledge base K, and of cw^m-entailment can be easily extended to $\mathcal{ALC}$. For $\mathcal{EL}^+_\bot$ ranked knowledge bases, cw^m-entailment has been proven to be Π^p_2-complete and to satisfy the KLM postulates of a preferential consequence relation (Giordano and Theseider Dupré 2020a).

3 A multi-preferential interpretation of Self-organising maps

In this section, we report about the multi-preferential semantics for SOMs, originally introduced to support the plausibility of a semantics with multiple perferences, (Giordano, Gliozzi, and Theseider Dupré 2020), and later extended by the same authors to consider fuzzy interpretations and probabilistic interpretations (2021).

Self-organising maps, introduced by Kohonen (Kohonen, Schroeder, and Huang 2001), are particularly plausible neural network models that learn in a human-like manner. In this section we shortly describe the architecture of SOMs and report about Gliozzi and Plunkett's similarity-based account of category generalization based on SOMs (Gliozzi and Plunkett 2019).

SOMs consist of a set of neurons, or units, spatially organized in a grid (Kohonen, Schroeder, and Huang 2001). Each map unit u is associated with a world representation, given by a weight vector w_u of the same dimensionality as the input vectors. At the beginning of training, all weight vectors are initialized to random values, outside the range of values of the input stimuli. During training, the input elements are sequentially presented to all neurons of the map. After each presentation of an input x, the *best-matching unit* (BMU$_x$) is selected: this is the unit i whose weight vector w_i is closest to the stimulus x (i.e. $i = \arg\min_j \|x - w_j\|$).

The weights of the best matching unit and of its surrounding units are updated in order to maximize the chances that the same unit (or its surrounding units) will be selected as the best matching unit for the same stimulus or for similar stimuli on subsequent presentations. In particular, it reduces the distance between the best matching unit's weights (and its surrounding neurons' weights) and the incoming input. The learning process is incremental: after the presentation of each input, the map's representation of the input (in particular the representation of its best-matching unit) is updated in order to take into account the new incoming stimulus. At the end of the whole process, the SOM has learned to organize the stimuli in a topologically significant way: similar inputs (with respect to Euclidean distance) are mapped to close by areas in the map, whereas inputs which are far apart from each other are mapped to distant areas of the map.

Once the SOM has learned to categorize, to assess category generalization, Gliozzi and Plunkett (Gliozzi and Plunkett 2019) define the map's disposition to consider a new stimulus y as a member of a known category C as a function of the *distance* of y from the *map's representation* of C. They use BMU_C to refer to the map's representation of category C and define category generalization as depending on the distance of the new stimulus y with respect to the category representation *compared to* the maximal distance from that representation of all known instances of the category. This is captured by the following notion of *relative distance* (*rd* for short) (Gliozzi and Plunkett 2019) :

$$rd(y, C) = \frac{min\|y - BMU_C\|}{max_{x \in C}\|x - BMU_x\|} \tag{1}$$

where $min\|y - BMU_C\|$ is the (minimal) Euclidean distance between y and C's category representation, and $max_{x \in C}\|x - BMU_x\|$ expresses the *precision* of category representation, and is the (maximal) Euclidean distance between any known member of the category and the category representation.

By judging a new stimulus as belonging to a category by comparing the distance of the stimulus from the category representation to the precision of the category representation, Gliozzi and Plunkett demonstrate (Gliozzi and Plunkett 2019) that the Numerosity and Variability effects of category generalization, described by Griffiths and Tenenbaum (Tenenbaum and Griffiths 2001), and usually explained with Bayesian tools, can be accommodated within a simple and psychologically plausible similarity-based account. Their notion of relative distance can as well be used as a basis for a logical semantics for SOMs.

3.1 Relating self-organising Maps and multi-preference models

Once the SOM has learned to categorize, we can regard the result of the categorization as a multipreference interpretation. Let X be the set of input stimuli from different categories, $C_1, \ldots, C_k$, which have been considered during the learning process. For each category C_i, we let BMU_{C_i} be the ensemble of best-matching units corresponding to the input stimuli of category C_i, i.e., $BMU_{C_i} = \{BMU_x \mid x \in X \text{ and } x \in C_i\}$. We regard the learned categories $C_1, \ldots, C_k$ as being the concept names (atomic concepts) in the description logic and we let them constitute our set of distinguished concepts $\mathcal{C} = \{C_1, \dots, C_k\}$.

To construct a multi-preference interpretation, first we fix the *domain* Δ^s to be the space of all possible stimuli; then, for each category (concept) C_i, we define a preference relation $<_{C_i}$, exploiting the notion of relative distance of each stimulus y from the map's representation of C_i. Finally, we define the interpretation of concepts.

Let Δ^s be the set of all the possible stimuli, including all input stimuli ($X \subseteq \Delta^s$) as well as the best matching units of input stimuli (i.e., $\{BMU_x \mid x \in X\} \subseteq \Delta^s$). For simplicity, we will assume the space of input stimuli to be finite.

Once the SOM has learned to categorize, the notion of relative distance $rd(x, C_i)$ of a stimulus x from a category C_i can be used to build a binary preference relation $<_{C_i}$ among the stimuli in Δ^s w.r.t. category C_i as follows: for all $x, x' \in \Delta^s$,

$$x <_{C_i} x' \text{ iff } rd(x, C_i) < rd(x', C_i) \qquad (2)$$

Each preference relation $<_{C_i}$ is a strict modular partial order relation on Δ^s. The relation $<_{C_i}$ is also well-founded, as we have assumed Δ^s to be finite.

We exploit this notion of preference to define a concept-wise multipreference interpretation associated with the SOM. We restrict to the boolean fragment of $\mathcal{ALC}$ with no individual names and no roles.

Definition 2 (multipreference-model of a SOM)**.** *The* multipreference-model of the SOM *is a multipreference interpretation* $\mathcal{M}^s = \langle \Delta^s, <_{C_1}, \dots, <_{C_k}, \cdot^I \rangle$ *such that:*

(i) Δ^s *is the set of all the possible stimuli, as above;*

(ii) for each $C_i \in \mathcal{C}$*,* $<_{C_i}$ *is the preference relation defined by equivalence (2).*

(iii) the interpretation function $\cdot^I$ *is defined for concept names (i.e. categories)* C_i *as:*

$$C_i^I = \{y \in \Delta^s \mid rd(y, C_i) \leq rd_{max, C_i}\}$$

where rd_{max,C_i} *is the maximal relative distance of an input stimulus* $x \in C_i$ *from category* C_i*, that is,* $rd_{max,C_i} = max_{x \in C_i}\{rd(x, C_i)\}$*. The interpretation function* $\cdot^I$ *is extended to complex concepts in the fragment of* $\mathcal{LC}$ *according to Definition 1.*

Informally, we interpret as C_i-elements those stimuli whose relative distance from category C_i is not larger than the relative distance of any input exemplar belonging to category C_i. Given $<_{C_i}$, we can identify the most typical C_i-elements wrt $<_{C_i}$ as the C_i-elements whose relative distance from category C_i is minimal, i.e., the elements in $min_{<_{C_i}}(C_i^I)$. Observe that the best matching unit BMU_x of an input stimulus $x \in C_i$ is an element of Δ^s. As, for $y = BMU_x$, $rd(y, C_i)$ is 0, $BMU_{C_i} \subseteq min_{<_{C_i}}(C_i^I)$.

3.2 Evaluation of concept inclusions by model checking

We have defined a multipreference interpretation $\mathcal{M}^s$ where, in the domain Δ^s of the possible stimuli, we are able to identify, for each category C_i, the C_i-elements as well as the most typical C_i-elements wrt $<_{C_i}$. We can exploit $\mathcal{M}^s$ to verify which inclusions are satisfied by the SOM by *model checking*, i.e., by checking the satisfiability of inclusions over model $\mathcal{M}^s$. This can be done both for strict concept inclusions of the form $C_i \sqsubseteq C_j$ and for defeasible inclusions of the form $\mathbf{T}(C_i) \sqsubseteq C_j$, where C_i and C_j are concept names (i.e., categories), by exploiting a notion of maximal relative distance of BMU_{C_i} from C_j, defined as $rd(BMC_{C_i}, C_j) = max_{x \in C_i}\{rd(BMU_x, C_j)\}$. We refer to (Giordano, Gliozzi, and Theseider Dupré 2020; Giordano, Gliozzi, and Theseider Dupré 2021) for details. Let us observe that checking the satisfiability of strict or defeasible inclusions on the SOM may be non trivial, depending on the number of input stimuli that have been considered in the learning phase, although from a logical point of view, this is just model checking. Gliozzi and Plunkett have considered self-organising maps that are able to learn from a limited number of input stimuli, although this is not generally true for all self-organising maps (Gliozzi and Plunkett 2019).

Note also that the multipreference interpretation $\mathcal{M}^s$ introduced in Definition 2 allows to determine the set of C_i-elements for all learned categories C_i and to define the most typical C_i-elements, exploiting the preference relation $<_{C_i}$. Although we are not able to define, for instance, the most typical $C_i \sqcap C_j$-elements just using single preferences, starting from $\mathcal{M}^s$, we can construct a concept-wise multipreference interpretation $\mathcal{M}^{som}$ that combines the preferential relations in $\mathcal{M}^s$ into a global preference relation $<$, and provides an intepretation to all typicality concepts as $\mathbf{T}(C_i \sqcap C_j)$. The interpretation $\mathcal{M}^{som}$ can be constructed from $\mathcal{M}^s$ according to the definition of the global preference in Section 2.

As an alternative to a multipreference semantics for SOMs, a fuzzy semantics has also been considered (Giordano, Gliozzi, and Theseider Dupré 2021), based on fuzzy Description Logics (Lukasiewicz and Straccia 2009), as well as a related probabilistic account exploiting Zadeh's probability of fuzzy events (Zadeh 1968).

Our work has focused on the multipreference interpretation of a self-organising map after the learning phase. However, the state of the SOM during the learning phase can as well be represented as a multipreference model (in the same way). During training, the current state of the SOM corresponds to a model representing the beliefs about the input stimuli considered so far (beliefs concerning the category of the stimuli). One can regard the category generalization process as a model building process and, in a way, as a belief change process. For future work, it would be interesting to study the properties of this notion of change and compare it with the notions of change studied in the literature (Gardenförs 1988; Gardenfors and Rott 1995; Katsuno and Mendelzon 1989; Katsuno and Sato 1991).

4 A multi-preferential interpretation of multilayer perceptrons

Let us first shortly introduce multilayer perceptrons. We first recall from (Haykin 1999) the model of a *neuron* as an information-processing unit in an (artificial) neural network. The basic elements are the following:

- a set of *synapses* or *connecting links*, each one characterized by a *weight*. We let x_j be the signal at the input of synapse j connected to neuron k, and w_{kj} the related synaptic weight;

- the adder for summing the input signals to the neuron, weighted by the respective synapses weights: $\sum_{j=1}^{n} w_{kj} x_j$;
- an *activation function* for limiting the amplitude of the output of the neuron (typically, to the interval $[0, 1]$ or $[-1, +1]$).

The sigmoid, threshold and hyperbolic-tangent functions are examples of activation functions. A neuron k can be described by the following pair of equations:

$$u_k = \sum_{j=1}^{n} w_{kj} x_j$$

$$y_k = \varphi(u_k + b_k)$$

where $x_1, \ldots, x_n$ are the input signals and $w_{k1}, \ldots, w_{kn}$ are the weights of neuron k; b_k is the bias, φ the activation function, and y_k is the output signal of neuron k. By adding a new synapse with input $x_0 = +1$ and synaptic weight $w_{k0} = b_k$, one can write: $u_k = \sum_{j=0}^{n} w_{kj} x_j$, and $y_k = \varphi(u_k)$, where u_k is called the *induced local field* of the neuron.

A neural network can then be seen as "a directed graph consisting of nodes with interconnecting synaptic and activation links" (Haykin 1999). Nodes in the graph are the neurons (the processing units) and the weight w_{ij} on the edge from node j to node i represents "the strength of the connection [..] by which unit j transmits information to unit i" (McLeod, Plunkett, and Rolls 1998). MLPs are classified by their synaptic connection topology. In a *feedforward* network the architectural graph is acyclic, while in a *recurrent* network it contains cycles. In a feedforward network neurons are organized in layers. In a *single-layer* network there is an input-layer of source nodes and an output-layer of computation nodes. In a *multilayer feedforward* network there is one or more hidden layer, whose computation nodes are called *hidden neurons* (or hidden units). The source nodes in the input-layer supply the activation pattern (*input vector*) providing the input signals for the first layer computation units, and so on, up to the final output layer of the network, which provides the overall response of the network to the activation pattern. In a recurrent network at least one feedback exists.

4.1 A two-valued multipreference interpretation of multilayer perceptrons

In the following, we consider a deep network $\mathcal{N}$ after training, when the synaptic weights w_{kj} have been learned. We associate a concept name $C_i \in N_C$ to any unit i in $\mathcal{N}$ (including input units and hidden units) and construct a multi-preference interpretation over a (finite) *domain* Δ of input stimuli, the input vectors considered so far, for training and generalization. In case the network is not feedforward, we assume that, for each input vector v in Δ, the network reaches a stationary state (Haykin 1999), in which $y_k(v)$ is the activity level of unit k. In essence, we are not considering the transient behavior of the network, but rather it behavior at stationary states.

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a subset of concepts in N_C, the concepts associated to the units we are focusing on (e.g., $\mathcal{C}$ might be associated to a subset of output units, or to all units). We associate to $\mathcal{N}$ and Δ a (two-valued) concept-wise multipreference interpretation over the boolean fragment of $\mathcal{ALC}$ (with no roles or individual names).

Definition 3. *The* cw^{m}interpretation $\mathcal{M}_{\mathcal{N}}^{\Delta} = \langle \Delta, <_{C_1}, \ldots, <_{C_n}, <, \cdot^I \rangle$ over Δ for network $\mathcal{N}$ *wrt* $\mathcal{C}$ *is a cw^m-interpretation where:*

- *the interpretation function $\cdot^I$ is defined for named concepts $C_k \in N_C$ as: $x \in C_k^I$ if $y_k(x) \neq 0$, and $x \notin C_k^I$ if $y_k(x) = 0$;*
- *for $C_k \in \mathcal{C}$, relation $<_{C_k}$ is defined for $x, x' \in \Delta$ as: $x <_{C_k} x'$ iff $y_k(x) > y_k(x')$, where $y_k(x)$ is the output signal of unit k for input vector x.*

The relation $<_{C_k}$ is a strict partial order, and $\leq_{C_k}$ and $\sim_{C_k}$ are defined as usual. In particular, $x \sim_{C_k} x'$ for $x, x' \notin C_k^I$. Clearly, the boundary between the domain elements which are in C_k^I and those which are not could be defined differently, e.g., by letting $x \in C_k^I$ if $y_k(x) > 0.5$, and $x \notin C_k^I$ if $y_k(x) \leq 0.5$. This would require only a minor change in the definition of the $<_{C_k}$.

This model provides a multipreference interpretation of the network $\mathcal{N}$, based on the input stimuli considered in Δ. For instance, when the neural network is used for categorization and a single output neuron is associated to each category, each concept C_h associated to an output unit h corresponds to a learned category. If $C_h \in \mathcal{C}$, the preference relation $<_{C_h}$ determines the relative typicality of input stimuli wrt category C_h. This allows to verify typicality properties concerning categories, such as $\mathbf{T}(C_h) \sqsubseteq D$ (where D is a boolean concept built from the named concepts in N_C), by *model checking* on the model $\mathcal{M}_{\mathcal{N}}^{\Delta}$.

Evaluating properties involving hidden units might be of interest, although their meaning is usually unknown. In the well known Hinton's family example (Hinton 1986), one may want to verify whether, normally, given an old Person 1 and relationship Husband, Person 2 would also be old, i.e., $\mathbf{T}(Old_1 \sqcap Husband) \sqsubseteq Old_2$ is satisfied. Here, concept Old_1 (resp., Old_2) is associated to a (known, in this case) hidden unit for Person 1 (and Person 2), while Husband is associated to an input unit.

4.2 From a two-valued to a fuzzy preferential interpretation of multilayer perceptrons

The definition of a fuzzy model of a neural network $\mathcal{N}$, under the same assumptions as in the previous section, is straightforward. In a fuzzy DL interpretation $I = \langle \Delta, \cdot^I \rangle$ (Lukasiewicz and Straccia 2009) concepts can be interpreted as fuzzy sets, and the fuzzy interpretation function $\cdot^I$ assigns to each concept $C \in N_C$ a function $C^I : \Delta \rightarrow [0, 1]$. For a domain element $x \in \Delta$, $C^I(x)$ represents the degree of membership of x in concept C.

Let N_C be the set containing a concept name C_i for each unit i in $\mathcal{N}$, including hidden units. We restrict to the boolean fragment of $\mathcal{ALC}$ with no individual names and no roles. A *fuzzy interpretation* $I_{\mathcal{N}} = \langle \Delta, \cdot^I \rangle$ *for* $\mathcal{N}$ (Giordano and Theseider Dupré 2021b) is defined as follows:

(i) Δ is a (finite) set of input stimuli;

(ii) the interpretation function $\cdot^I$ is defined for named concepts $C_k \in N_C$ as: $C_k^I(x) = y_k(x), \forall x \in \Delta$; where $y_k(x)$ is the output signal of neuron k, for input vector x.

The verification that a fuzzy axiom $\langle C \sqsubseteq D \geq \alpha \rangle$ is satisfied in the model $I_{\mathcal{N}}$, can be done based on satisfiability in fuzzy DLs, according to the choice of the t-norm and implication function. It requires $C_k^I(x)$ to be recorded for all $k = 1, \ldots, n$ and $x \in \Delta$. Of course, one could restrict N_C to the concepts associated to input and output units in $\mathcal{N}$, so to capture the input/output behavior of the network.

The fuzzy interpretation $I_{\mathcal{N}}$ above, induces a preference relation over the domain Δ as, for all $x, x' \in \Delta$, $x <_{C_k} x'$ iff $y_k(x) > y_k(x')$. Based on this idea, a fuzzy multipreference interpretation $\mathcal{M}_{\mathcal{N}}^{f,\Delta} = \langle \Delta, <_{C_1}, \ldots, <_{C_n}, \cdot^I \rangle$ over $\mathcal{C}$ can be associated to the network $\mathcal{N}$ starting from $I_{\mathcal{N}}$. In a fuzzy multipreference interpretation a typicality concept $\mathbf{T}(C)$ can be interpreted as a crisp concept having the value 1 for the minimal C-elements in the domain with respect to the preference relation $<_C$, and 0 otherwise. This relation is well-founded if we restrict to finite models (as we do), or to witnessed models, as usual in fuzzy DLs (Lukasiewicz and Straccia 2009).

5 Multilayer perceptrons as weighted conditional knowledge bases

The three interpretations considered above for MLPs describe the input-output behavior of the network, and allow for the verification of properties by model-checking. The last one, $\mathcal{M}_{\mathcal{N}}^{f,\Delta}$ is, in essence, a combination of the first two, and can be proved to be a model of the neural network $\mathcal{N}$ when regarded as a weighted conditional knowledge base.

In this section, we report the notion of a weighted conditional knowledge base for $\mathcal{ALC}$ from (Giordano and Theseider Dupré 2021b), and we describe how a weighted conditional knowledge base $K_{\mathcal{N}}$ can be associated to a deep network $\mathcal{N}$. We give some hint about its two-valued and fuzzy multipreference semantics, and we refer to (Giordano and Theseider Dupré 2021b) for a detailed description.

5.1 Weighted conditional knowledge bases

Weighted $\mathcal{ALC}$ knowledge bases are $\mathcal{ALC}$ knowledge bases in which defeasible or typicality inclusions of the form $\mathbf{T}(C) \sqsubseteq D$ are given a positive or negative weight (a real number).

A *weighted $\mathcal{ALC}$ knowledge base K*, over a set $\mathcal{C} = \{C_1, \ldots, C_k\}$ of distinguished $\mathcal{ALC}$ concepts, is a tuple $\langle \mathcal{T}, \mathcal{T}_{C_1}, \ldots, \mathcal{T}_{C_k}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a set of $\mathcal{ALC}$ inclusion axiom, $\mathcal{A}$ is a set of $\mathcal{ALC}$ assertions and $\mathcal{T}_{C_i} = \{(d_h^i, w_h^i)\}$ is a set of weighted typicality inclusions $d_h^i = \mathbf{T}(C_i) \sqsubseteq D_{i,h}$ for concept C_i, where each inclusion d_h^i has a weight w_h^i, a real number. The concepts C_i occurring on the l.h.s. of some typicality inclusion $\mathbf{T}(C_i) \sqsubseteq D$ are called *distinguished concepts*. In the fuzzy case, $\mathcal{T}$ and $\mathcal{A}$ contain fuzzy axioms.

Example 4. *Consider the weighted knowledge base $K = \langle \mathcal{T}, \mathcal{T}_{Bird}, \mathcal{T}_{Penguin}, \mathcal{A} \rangle$, over the set of distinguished concepts $\mathcal{C} = \{Bird, Penguin\}$, with empty ABox and with $\mathcal{T}$ containing the inclusions $Penguin \sqsubseteq Bird$ and $Black \sqcap Grey \sqsubseteq \bot$. The weighted TBox $\mathcal{T}_{Bird}$ contains the following weighted defeasible inclusions:*

(d_1) $\mathbf{T}(Bird) \sqsubseteq Fly$, *+20*

(d_2) $\mathbf{T}(Bird) \sqsubseteq \exists has\_Wings.\top$, *+50*

(d_3) $\mathbf{T}(Bird) \sqsubseteq \exists has\_Feathers.\top$, *+50;*

$\mathcal{T}_{Penguin}$ contains the defeasible inclusions:

(d_4) $\mathbf{T}(Penguin) \sqsubseteq Fly$, *- 70*

(d_5) $\mathbf{T}(Penguin) \sqsubseteq Black$, *+50;*

(d_6) $\mathbf{T}(Penguin) \sqsubseteq Grey$, *+10;*

The meaning is that a bird normally has wings, has feathers and flies, but having wings and feathers (both with weight 50) for a bird is more plausible than flying (weight 20), although flying is regarded as being plausible. For a penguin, flying is not plausible (inclusion (d_4) has a negative weight -70), while being black or being grey are plausible properties of prototypical penguins, in fact, (d_5) and (d_6) have positive weights, resp. 50 and 10, so that being black is more plausible than being grey.

A two-valued semantics for weighted DL knowledge bases has been defined by developing a semantic closure construction in the same spirit as Lehmann's lexicographic closure (1995), but more related to Kern-Isberner's semantics of c-representations (2001; 2014). The approach of c-representations assigns an individual impact to each conditional and generates the world ranks as a sum of impacts of falsified conditionals. Here, conditionals have a positive or negative weight, and negative weights can be interpreted as penalties. We consider a concept-wise construction, as we want to associate different (ranked) preferences to the different concepts. For an element x in the domain Δ, and a concept C_i, the weight $W_i(x)$ of x wrt C_i is defined as the sum of the weights w_h^i of the typicality inclusions $\mathbf{T}(C_i) \sqsubseteq D_{i,h}$ in $\mathcal{T}_{C_i}$ verified by x (and is $-\infty$ when x is not an instance of C_i). From the weights $W_i(x)$ the *preference relation* $\leq_{C_i}$ can be defined by letting: for $x, y \in \Delta$, $x \leq_{C_i} y$ iff $W_i(x) \geq W_i(y)$. The higher the weight of x wrt C_i the higher its typicality relative to C_i. This closure construction defines preferences $<_{C_i}$ and allows for the definition of *concept-wise multipreference interpretations* as in Section 2.

A similar construction has been adopted in the fuzzy case. Rather then summing weights w_h^i of the typicality inclusions $\mathbf{T}(C_i) \sqsubseteq D_{i,h} \in \mathcal{T}_{C_i}$ verified in I, $W_i(x)$ is defined by summing the products $w_h^i \cdot D_{i,h}^I(x)$ for all h, thus considering the degree of membership of x in each $D_{i,h}$ (a value in the interval $[0, 1]$). Furthermore, for fuzzy multipreference interpretations, a condition is needed to enforce the *coherence* of the values $C_i^I(x)$, defining the degree of membership of a domain element x in a concept C_i in a fuzzy interpretation I, with the weights $W_i(x)$, which are computed from the knowledge base (given I). The requirement that, for all $x, y \in \Delta$, $C_i^I(x) \geq C_i^I(y)$ iff $W_i(x) \geq W_i(y)$ leads to the definition of *coherent fuzzy multipreference models* (cf^m-models) of the weighted conditional knowledge base. We refer to (Giordano and Theseider Dupré 2021b) for details.

5.2 Mapping multilayer perceptrons to conditional knowledge bases

Let us now consider how a multilayer perceptron can be mapped to a weighted conditional knowledge base. For each unit k, we consider all the units $j_1, \ldots, j_m$ whose output signals are the input signals of unit k, with synaptic weights $w_{k,j_1}, \ldots, w_{k,j_m}$. Let C_k be the concept name associated to unit k and $C_{j_1}, \ldots, C_{j_m}$ the concept names associated to units $j_1, \ldots, j_m$, respectively. For each unit k the following set $\mathcal{T}_{C_k}$ of typicality inclusions is defined, with their associated weights:

$\mathbf{T}(C_k) \sqsubseteq C_{j_1}$ with w_{k,j_1},
$\ldots,$
$\mathbf{T}(C_k) \sqsubseteq C_{j_m}$ with w_{k,j_m}.

The KB extracted from network $\mathcal{N}$ is defined as the tuple: $K^{\mathcal{N}} = \langle \mathcal{T}_{strict}, \mathcal{T}_{C_1}, \ldots, \mathcal{T}_{C_n}, \mathcal{A} \rangle$, where $\mathcal{T}_{strict} = \mathcal{A} = \emptyset$ and, for each $k \in \mathcal{N}$, $C_k \in \mathcal{C}$, and $K^{\mathcal{N}}$ contains the set $\mathcal{T}_{C_k}$ of weighted typicality inclusions associated to neuron k (as defined above). $K^{\mathcal{N}}$ is a weighted knowledge base over the set of distinguished concepts $\mathcal{C} = \{C_1, \ldots, C_n\}$. Given a network $\mathcal{N}$, it can be proven that the interpretation $\mathcal{M}_{\mathcal{N}}^{f,\Delta}$ (see Section 4.2) is a cf^m-model of the knowledge base $K^{\mathcal{N}}$, provided the activation functions φ of all units are monotonically increasing and have value in $(0, 1]$.
Under some conditions on activation functions (that hold, for instance, for the sigmoid activation function), for any choice of $\mathcal{C} \subseteq N_C$ and for any choice of the domain Δ of input stimuli (provided that they lead to a stationary state of $\mathcal{N}$), the fm-interpretation $\mathcal{M}_{\mathcal{N}}^{f,\Delta}$ is a coherent fuzzy multipreference model of the defeasible knowledge base $K^{\mathcal{N}}$.

This result can be further generalized by weakening the notion of coherence of a fuzzy multipreference interpretation to a notion called *weak consistency* in the technical report (Giordano and Theseider Dupré 2020b) and *faithfulness* in (Giordano 2021), where it has been proven that, also in the fuzzy case, the concept-wise multipreference semantics has interesting properties and satisfies most of the KLM properties of a preferential consequence relation, depending of their reformulation and on the fuzzy combination functions.

6 Conclusions

We have explored the relationships between a concept-wise multipreference semantics and two very different neural network models, Self-Organising Maps and Multilayer Perceptrons, showing that a multi-preferential semantics can be used to provide a logical model of the network behavior after training. Such a model can be used to learn or to validate conditional knowledge from the empirical data used for training and generalization, by model checking of logical properties. A two-valued KLM-style preferential interpretation with multiple preferences has been considered, based on the idea of associating preference relations to categories (in the case of SOMs) or to neurons (for Multilayer Perceptrons), as well as a fuzzy semantics. Due to the diversity of the two models we would expect that a similar approach might be extended to other neural network models and learning approaches.

Much work has been devoted, in recent years, to the combination of neural networks and symbolic reasoning (d'Avila Garcez, Broda, and Gabbay 2001; d'Avila Garcez, Lamb, and Gabbay 2009; d'Avila Garcez et al. 2019), leading to the definition of new computational models, such as Graph Neural Networks (Lamb et al. 2020), Logic Tensor Network (Serafini and d'Avila Garcez 2016), Recursive Reasoning Networks (Hohenecker and Lukasiewicz 2020), neural-symbolic stream fusion (Le-Phuoc, Eiter, and Le-Tuan 2021), and to extensions of logic programming languages with neural predicates (Manhaeve et al. 2018; Yang, Ishay, and Lee 2020). Among the earliest systems combining logical reasoning and neural learning are the KBANN (Towell and Shavlik 1994) and the CLIP (d'Avila Garcez and Zaverucha 1999) systems and Penalty Logic (Pinkas 1995), a non-monotonic reasoning formalism used to establish a correspondence with symmetric connectionist networks. The relationships between normal logic programs and connectionist network have been investigated by Garcez et al. (1999; 2001) and by Hitzler et al. (2004).

The correspondence between neural network models and fuzzy systems has been first investigated by Bart Kosko in his seminal work (Kosko 1992). In his view, "at each instant the n-vector of neuronal outputs defines a fuzzy unit or a fit vector. Each fit value indicates the degree to which the neuron or element belongs to the n-dimentional fuzzy set." Our fuzzy interpretation of a multilayer perceptron regards, instead, each concept (representing a single neuron) as a fuzzy set. This is the usual way of viewing concepts in fuzzy DLs (Straccia 2005; Lukasiewicz and Straccia 2008; Bobillo and Straccia 2016), and we have used fuzzy concepts within a multipreference semantics based on a semantic closure construction in the line of Lehmann's semantics for lexicographic closure (Lehmann 1995) and Kern-Isberner's c-representations (Kern-Isberner 2001; Kern-Isberner and Eichhorn 2014). A combination of fuzzy logic with the preferential semantics of conditional knowledge bases has been first studied by Casini and Straccia (2013), who have also developed a rational closure construction for propositional Gödel logic. The multipreference semantics we have introduced for weighted conditionals appears to be a relative of c-representations, which generate the world ranks as a sum of impacts of falsified conditionals, (Kern-Isberner 2001; Kern-Isberner 2004).

We have further considered a semantics with multiple preferences, in order to make it concept-wise: each distinguished concept C_i has its own set $\mathcal{T}_{C_i}$ of (weighted) typicality inclusions, and an associated preference relation $<_{C_i}$. This allows a preference relation to be associated to each category (e.g., in the preferential interpretation of SOMs) or neuron (in a deep network). Related semantics with multiple preferences have been proposed, starting from Brewka's framework of basic preference descriptions (Brewka 2004), based on different approaches: in system ARS, as a refinement of System Z by Kern-Isberner and Ritterskamp (2010), using techniques for handling preference fusion; in $\mathcal{ALC} + \mathbf{T}$ (an extension of $\mathcal{ALC}$ with typicality) by Gil (2014); in a refinement of rational closure by Gliozzi (2016); by associating multiple preferences to roles by Britz and Varzinczak (2018;

2019); in ranked $\mathcal{EL}$ knowledge bases by Giordano and Theseider Dupré (2020a); in the first-order logic setting by Delgrande and Rantsaudis (2020).

For Multilayer Perceptrons, under a fuzzy semantics, a deep neural network can itself be regarded as a conditional knowledge base, where conditional implications are associated to synaptic connections with their weights. That a conditional logic, belonging to a family of logics which are normally used for hypothetical and counterfactual reasoning, for common sense reasoning, and for reasoning with exceptions, can be used for capturing reasoning in a deep neural network model is rather surprising. It suggests that slow thinking and fast thinking (Kahneman 2011) might be more related than expected.

Opening the black-box and recognizing that multilayer perceptrons can be seen as a set of conditionals, can be exploited as a possible basis for an integrated use of symbolic reasoning and neural networks (at least for this neural network model). While a neural network, once trained, is able and fast in classifying the new stimuli (that is, it is able to do instance checking), all other reasoning services such as satisfiability, entailment and model-checking are missing. These capabilities would be needed for dealing with tasks combining empirical and symbolic knowledge, such as, for instance: to prove whether the network satisfies some (strict or conditional) properties; to learn the weights of a conditional knowledge base from empirical data; to combine defeasible inclusions extracted from a neural network with other defeasible or strict inclusions for inference.

To make these tasks possible, the development of proof methods for such logics is a preliminary step. Undecidability results for fuzzy description logics with general inclusion axioms (Baader and Peñaloza 2011; Cerami and Straccia 2011; Borgwardt and Peñaloza 2012) motivate the investigation of decidable approximations of fuzzy-multipreference entailment. In the two-valued case multipreference entailment is decidable for weighted $\mathcal{EL}^{\perp}$ knowledge bases and a proof method for reasoning with weighted conditional knowledge bases with integer weights has been developed (Giordano and Theseider Dupré 2021a) by exploiting Answer Set Programming (ASP) and *asprin* (Brewka et al. 2015). The approach is based on a fragment of the materialization calculus (Krötzsch 2010), and has been defined by adapting the encoding for ranked $\mathcal{EL}_{\perp}^{+}$ knowledge bases (Giordano and Theseider Dupré 2020a). This is a first step towards the definition of proof methods for multi-valued extensions of our concept-wise preferential semantics based on a notion of faithful interpretations (Giordano 2021). Other possible extensions concern the definition of multiple typicality operators, based on the combination of selected concepts, and a temporal extension to capture the transient behavior of Multilayer Perceptrons.

An interesting issue is whether the mapping of deep neural networks to weighted conditional knowledge bases can be extended to more complex neural network models, such as Graph Neural Networks (Lamb et al. 2020), or whether different logical formalisms and semantics would be needed.

Another issue is whether the fuzzy-preferential interpretation of neural networks can be related with the probabilistic interpretation of neural networks based on statistical AI. This is an interesting issue, as the fuzzy DL interpretations we have considered, where concepts are regarded as fuzzy sets, also suggest a probabilistic account based on Zadeh's probability of fuzzy events (Zadeh 1968). We refer to (Giordano, Gliozzi, and Theseider Dupré 2021) for some results concerning a probabilistic interpretation of SOMs and to (Giordano and Theseider Dupré 2020b) for a preliminary account for MLPs. A methodology for commonsense reasoning based on probabilistic conditional knowledge under the principle of maximum entropy (MaxEnt) has been developed by Kern-Isberner (1998) starting from the propositional case. Wilhelm et al. (2019) have recently shown how to calculate MaxEnt distributions in a first-order setting by using typed model counting and condensed iterative scaling, and have explored the connection to Markov Logic Networks for drawing inferences. A description logic with probabilistic conditionals $\mathcal{ALC}^{\mathcal{ME}}$ has also been proposed (Wilhelm and Kern-Isberner 2019) based on this methodology.

References

Adadi, A., and Berrada, M. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6:52138–52160.

Arrieta, A. B.; Rodríguez, N. D.; Ser, J. D.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; Chatila, R.; and Herrera, F. 2020. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* 58:82–115.

Baader, F., and Peñaloza, R. 2011. Are fuzzy description logics with general concept inclusion axioms decidable? In *IEEE Int. Conf. on Fuzzy Systems, Taipei, Taiwan, 27-30 June, 2011, Proc.*, 1735–1742. IEEE.

Beierle, C.; Falke, T.; Kutsch, S.; and Kern-Isberner, G. 2017. System Z^{FO}: Default reasoning with system z-like ranking functions for unary first-order conditional knowledge bases. *Int. J. Approx. Reason.* 90:120–143.

Bobillo, F., and Straccia, U. 2016. The fuzzy ontology reasoner fuzzydl. *Knowl. Based Syst.* 95:12–34.

Bonatti, P. A.; Faella, M.; Petrova, I.; and Sauro, L. 2015. A new semantics for overriding in description logics. *Artif. Intell.* 222:1–48.

Borgwardt, S., and Peñaloza, R. 2012. Undecidability of fuzzy description logics. In Brewka, G.; Eiter, T.; and McIlraith, S. A., eds., *Principles of Knowledge Representation and Reasoning: Proc. of the 13th Int. Conf., KR 2012, Rome, Italy, June 10-14, 2012*.

Brewka, G.; Delgrande, J. P.; Romero, J.; and Schaub, T. 2015. asprin: Customizing answer set preferences without a headache. In *Proc. AAAI 2015*, 1467–1474.

Brewka, G. 2004. A rank based description language for qualitative preferences. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, Valencia, Spain, August 22-27, 2004*, 303–307.

Britz, K., and Varzinczak, I. J. 2018. Rationality and context in defeasible subsumption. In *Proc. 10th Int. Symp. on Found. of Information and Knowledge Systems, FoIKS 2018, Budapest, May 14-18, 2018*, 114–132.

Britz, A., and Varzinczak, I. 2019. Contextual rational closure for defeasible ALC (extended abstract). In *Proc. 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019*.

Britz, K.; Heidema, J.; and Meyer, T. 2008. Semantic preferential subsumption. In Brewka, G., and Lang, J., eds., *KR 2008*, 476–484. Sidney, Australia: AAAI Press.

Casini, G., and Straccia, U. 2010. Rational Closure for Defeasible Description Logics. In Janhunen, T., and Niemelä, I., eds., *JELIA 2010*, volume 6341 of *LNCS*, 77–90. Helsinki: Springer.

Casini, G., and Straccia, U. 2013. Towards rational closure for fuzzy logic: The case of propositional gödel logic. In *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings*, volume 8312 of *LNCS*, 213–227. Springer.

Casini, G.; Meyer, T.; Varzinczak, I. J.; ; and Moodley, K. 2013. Nonmonotonic Reasoning in Description Logics: Rational Closure for the ABox. In *DL 2013*, volume 1014 of *CEUR Workshop Proceedings*, 600–615.

Cerami, M., and Straccia, U. 2011. On the undecidability of fuzzy description logics with gcis with lukasiewicz t-norm. *CoRR* abs/1107.4212.

d'Avila Garcez, A. S., and Zaverucha, G. 1999. The connectionist inductive learning and logic programming system. *Appl. Intell.* 11(1):59–77.

d'Avila Garcez, A. S.; Gori, M.; Lamb, L. C.; Serafini, L.; Spranger, M.; and Tran, S. N. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP* 6(4):611–632.

d'Avila Garcez, A. S.; Broda, K.; and Gabbay, D. M. 2001. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artif. Intell.* 125(1-2):155–207.

d'Avila Garcez, A. S.; Lamb, L. C.; and Gabbay, D. M. 2009. *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer.

Delgrande, J., and Rantsoudis, C. 2020. A preference-based approach for representing defaults in first-order logic. In *18th Int. Workshop on Non-Monotonic Reasoning, NMR2020, September 12th - 14th, 2020, workshop notes*.

Gardenfors, P., and Rott, H. 1995. Belief revision. *Handbook of Logic in Artificial Intelligence and Logic Programming, volume 4, ed. by D. M. Gabbay, C. J. Hogger, and J. A. Robinson*.

Gardenförs, P. 1988. *Knowledge in Flux*. MIT Press.

Gil, O. F. 2014. On the Non-Monotonic Description Logic ALC+T$_{\text{min}}$. *CoRR* abs/1404.6566.

Giordano, L., and Theseider Dupré, D. 2020a. An ASP approach for reasoning in a concept-aware multipreferential lightweight DL. *Theory and Practice of Logic programming, TPLP* 10(5):751–766.

Giordano, L., and Theseider Dupré, D. 2020b. Weighted defeasible knowledge bases and a multipreference semantics for a deep neural network model. *CoRR* abs/2012.13421.

Giordano, L., and Theseider Dupré, D. 2021a. Weighted conditional $\mathcal{EL}$ knowledge bases with integer weights: an ASP approach. In *Int. Conf. on logic Programming, ICLP 2021*. To appear.

Giordano, L., and Theseider Dupré, D. 2021b. Weighted defeasible knowledge bases and a multipreference semantics for a deep neural network model. In *Proc. 17th European Conf. on Logics in AI, JELIA 2021, May 17-20*, volume 12678 of *LNCS*, 225–242. Springer.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2007. Preferential Description Logics. In *LPAR 2007*, volume 4790 of *LNAI*, 257–272. Yerevan, Armenia: Springer.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Artif. Intell.* 226:1–33.

Giordano, L.; Gliozzi, V.; and Theseider Dupré, D. 2020. On a plausible concept-wise multipreference semantics and its relations with self-organising maps. In Calimeri, F.; Perri, S.; and Zumpano, E., eds., *CILC 2020, Rende, Italy, October 13-15, 2020*, volume 2710 of *CEUR*, 127–140.

Giordano, L.; Gliozzi, V.; and Theseider Dupré, D. 2021. A conditional, a fuzzy and a probabilistic interpretation of self-organising maps. *CoRR* abs/2103.06854.

Giordano, L. 2021. On the KLM properties of a fuzzy DL with Typicality. In *16th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2021*. Springer. To appear.

Gliozzi, V., and Plunkett, K. 2019. Grounding bayesian accounts of numerosity and variability effects in a similarity-based framework: the case of self-organising maps. *Journal of Cognitive Psychology* 31(5–6).

Gliozzi, V. 2016. Reasoning about multiple aspects in rational closure for DLs. In *Proc. AI*IA 2016, Genova, Italy, November 29 - December 1, 2016*, 392–405.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2019. A survey of methods for explaining black box models. *ACM Comput. Surv.* 51(5):93:1–93:42.

Haykin, S. 1999. *Neural Networks - A Comprehensive Foundation*. Pearson.

Hinton, G. 1986. Learning distributed representation of concepts. In *Proceedings 8th Annual Conference of the Cognitive Science Society. Erlbaum, Hillsdale, NJ*.

Hitzler, P.; Hölldobler, S.; and Seda, A. K. 2004. Logic programs and connectionist networks. *J. Appl. Log.* 2(3):245–272.

Hohenecker, P., and Lukasiewicz, T. 2020. Ontology reasoning with deep neural networks. *J. Artif. Intell. Res.* 68:503–540.

Kahneman, D. 2011. *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

Katsuno, H., and Mendelzon, A. O. 1989. A unified view of propositional knowledge base updates. In Sridharan, N. S., ed., *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, 1413–1419. Morgan Kaufmann.

Katsuno, H., and Sato, K. 1991. A unified view of consequence relation, belief revision and conditional logic. In *IJCAI'91*, 406–412.

Kern-Isberner, G., and Eichhorn, C. 2014. Structural inference from conditional knowledge bases. *Stud Logica* 102(4):751–769.

Kern-Isberner, G., and Ritterskamp, M. 2010. Preference fusion for default reasoning beyond system Z. *J. Autom. Reasoning* 45(1):3–19.

Kern-Isberner, G. 1998. Characterizing the principle of minimum cross-entropy within a conditional-logical framework. *Artif. Intell.* 98(1-2):169–208.

Kern-Isberner, G. 2001. *Conditionals in Nonmonotonic Reasoning and Belief Revision - Considering Conditionals as Agents*, volume 2087 of *LNCS*. Springer.

Kern-Isberner, G. 2004. A thorough axiomatization of a principle of conditional preservation in belief revision. *Ann. Math. Artif. Intell.* 40(1-2):127–164.

Kohonen, T.; Schroeder, M.; and Huang, T., eds. 2001. *Self-Organizing Maps, Third Edition*. Springer Series in Information Sciences. Springer.

Kosko, B. 1992. *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Prentice Hall.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44(1-2):167–207.

Krötzsch, M. 2010. Efficient inferencing for OWL EL. In *Proc. JELIA 2010*, 234–246.

Lamb, L. C.; d'Avila Garcez, A. S.; Gori, M.; Prates, M. O. R.; Avelar, P. H. C.; and Vardi, M. Y. 2020. Graph neural networks meet neural-symbolic computing: A survey and perspective. In Bessiere, C., ed., *Proc. IJCAI 2020*, 4877–4884. ijcai.org.

Le-Phuoc, D.; Eiter, T.; and Le-Tuan, A. 2021. A scalable reasoning and learning approach for neural-symbolic stream fusion. In *AAAI 2021, February 2-9*, 4996–5005. AAAI Press.

Lehmann, D., and Magidor, M. 1992. What does a conditional knowledge base entail? *Artificial Intelligence* 55(1):1–60.

Lehmann, D. J. 1995. Another perspective on default reasoning. *Ann. Math. Artif. Intell.* 15(1):61–82.

Lewis, D. 1973. *Counterfactuals*. Basil Blackwell Ltd.

Lukasiewicz, T., and Straccia, U. 2008. Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Semant.* 6(4):291–308.

Lukasiewicz, T., and Straccia, U. 2009. Description logic programs under probabilistic uncertainty and fuzzy vagueness. *Int. J. Approx. Reason.* 50(6):837–853.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and Raedt, L. D. 2018. Deepproblog: Neural probabilistic logic programming. In *NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 3753–3763.

McLeod, P.; Plunkett, K.; and Rolls, E., eds. 1998. *Introduction to Connectionist Modelling of Cognitive Processes*. Oxford university Press.

Nute, D. 1980. Topics in conditional logic. *Reidel, Dordrecht*.

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *TARK'90, Pacific Grove, CA, USA, 1990*, 121–135. Morgan Kaufmann.

Pensel, M., and Turhan, A. 2018. Reasoning in the defeasible description logic $EL_{\perp}$ - computing standard inferences under rational and relevant semantics. *Int. J. Approx. Reasoning* 103:28–70.

Pinkas, G. 1995. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. *Artif. Intell.* 77(2):203–247.

Serafini, L., and d'Avila Garcez, A. S. 2016. Learning and reasoning with logic tensor networks. In *Proc. AI*IA 2016, Genova, Italy, November 29 - December 1, 2016*, volume 10037 of *LNCS*, 334–348. Springer.

Straccia, U. 2005. Towards a fuzzy description logic for the semantic web (preliminary report). In *Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proc.*, volume 3532 of *LNCS*, 167–181. Springer.

Tenenbaum, J. B., and Griffiths, T. L. 2001. Generalization, similarity, and bayesian inference. *Behavioral and Brain Sciences* 24:629–641.

Towell, G. G., and Shavlik, J. W. 1994. Knowledge-based artificial neural networks. *Artif. Intell.* 70(1-2):119–165.

Wilhelm, M., and Kern-Isberner, G. 2019. Maximum entropy calculations for the probabilistic description logic $\mathcal{ALC}^{ME}$. In *Description Logic, Theory Combination, and All That, LNAI 11560, pp. 588–609.*

Wilhelm, M.; Kern-Isberner, G.; Finthammer, M.; and Beierle, C. 2019. Integrating typed model counting into first-order maximum entropy computations and the connection to markov logic networks. In *Proc. 32-nd Int. Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019*, 494–499. AAAI Press.

Yang, Z.; Ishay, A.; and Lee, J. 2020. Neurasp: Embracing neural networks into answer set programming. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1755–1762. ijcai.org.

Zadeh, L. 1968. Probability measures of fuzzy events. *J.Math.Anal.Appl* 23:421–427.

Postulates for Transformations Among Epistemic States Represented by Ranking Functions or Total Preorders

Jonas Haldimann[1] , **Christoph Beierle**[1] , **Gabriele Kern-Isberner**[2]

[1]FernUniversität in Hagen, 58084 Hagen, Germany
[2]TU Dortmund, 44221 Dortmund, Germany
{jonas.haldimann, christoph.beierle}@fernuni-hagen.de, gabriele.kern-isberner@cs.tu-dortmund.de

Abstract

There are different kinds of models for representing epistemic states. Two popular approaches to this are ranking functions (OCFs) and total preorders (TPOs) on possible worlds. Both approaches allow for modelling conditional beliefs. To better understand the relationship among the different models, we consider mappings between models concerning the preservation of desirable properties like syntax splitting, or the compatability with operations like marginalization and conditionalization. We introduce a set of postulates for such transitions and evaluate them with respect to mappings within and across the two frameworks. Doing this, we establish both dependencies as well as incompatibilities among the postulates. Our results will be useful in particular for transferring methods and tools developed for OCF-based semantics to the TPO framework as well as the other way around.

1 Introduction

In the field of knowledge representation, there is a long tradition to employ conditionals as fundamental objects. A conditional formalizes a defeasible rule "If A then usually B" for logical formulas A, B and is often denoted as $(B|A)$. A set of conditionals is called a conditional belief base. As conditional logic is more expressive than propositional logic, it requires a richer semantics as well. There are different approaches to the semantics for conditional logic, e.g., (Lewis 1973; Adams 1975; Kraus, Lehmann, and Magidor 1990; Pearl 1990; Dubois and Prade 1994; Benferhat, Dubois, and Prade 1999; Kern-Isberner 2004). These approaches often use either some form of ranking functions (Spohn 1988) or total preorders on interpretations as models for conditionals and conditional knowledge bases.

In this paper, we focus on these two kinds of models for conditionals, ranking functions (or ordinal conditional functions, OCFs) and total preorders on worlds (TPOs). Both models have their own advantages. TPOs are fundamental for nonmonotonic logics (AGM revision, System P) whereas OCFs are convenient implementations of TPOs that, however, crucially provide an arithmetics that is lacking in TPOs. This arithmetics allows in particular for a more sophisticated conditional reasoning, approximating nicely what is possible in probabilistics. Studying mappings between TPOs and OCFs make it possible to transfer properties and techniques from the richer framework of OCFs to TPOs, on the one hand, and to focus on purely qualitative aspects of reasoning and revision with OCFs, on the other hand. More specifically, TPOs are used in characterisation theorems for AGM revisions (Katsuno and Mendelzon 1992) as well as system P inference (Adams 1975; Kraus, Lehmann, and Magidor 1990). OCFs allow to model the strength of conditional beliefs by assigning numbers to logical interpretations (Spohn 1988; Goldszmidt and Pearl 1996). Furthermore, some belief revision operators with interesting properties have been defined for OCFs, e.g., (Kern-Isberner 2004). To better understand the connection between OCFs and TPOs, we investigate transformations among these frameworks, i.e., functions that map OCFs to TPOs or TPOs to OCFs. Furthermore, we generalize by also including transformations from OCFs to OCFs and TPOs to TPOs.

We formalize functions on these models within and across the two different frameworks as *epistemic state mappings* and propose postulates that govern epistemic state mappings. The postulates require the epistemic state mappings to preserve certain properties of the models like the entailed inference relation and syntax splittings. Syntax splitting is a concept describing that beliefs about different parts of the signature are uncorrelated (Parikh 1999; Peppas et al. 2015). Other postulates ensure compatibility with the operations marginalization and conditionalization. These operations are relevant e.g. for some forms of forgetting (Delgrande 2017; Eiter and Kern-Isberner 2019; Beierle et al. 2019), syntax splitting, and some aspects of belief revision (Kern-Isberner, Beierle, and Brewka 2020; Sezgin and Kern-Isberner 2020). We investigate relationships among our postulates in general as well as for each framework in particular. Our results elaborate dependencies among the postulates, and they also unveil situations where certain combinations of postulates cannot be satisfied simultanously.

In summary, the main contributions of this paper are:

- Introduction of epistemic state mappings for TPOs and OCFs
- Coverage of marginalization and conditionalization also for the iterated case via the introduction of restricted TPOs and restricted OCFs
- Formalization of desireable properties of epistemic state

mappings in terms of general postulates

- Establishment of relationships among the postulates and of realizability results for the postulates and for subsets thereof.

The paper is structured as follows. After giving some background on conditional logic, ranking functions and total preorders in Section 2, we introduce the operations marginalization and conditionalization and the property syntax splitting in Section 3. We proceed to introduce the concept of epistemic state mappings and postulates for such mappings in Section 4. Then we analyse the relationship among the postulates for epistemic state mappings from total preorders to total preorders in Section 5 and among the postulates for epistemic state mappings from ranking functions to ranking functions in Section 6. In Section 7, we consider epistemic state mappings from ranking functions to total preorders, and in Section 8 we consider epistemic state mappings from total preorders to ranking functions. In Section 9, we conclude and point out future work.

2 Background: Conditional Logic, Ranking Functions, and Total Preorders

A *(propositional) signature* is a finite set Σ of identifiers. For a signature Σ, we denote the propositional language over Σ by $\mathcal{L}_\Sigma$. Usually, we denote elements of the signatures with lowercase letters $a, b, c, \ldots$ and formulas with uppercase letters $A, B, C, \ldots$. We may denote a conjunction $A \wedge B$ by AB and a negation $\neg A$ by $\overline{A}$ for brevity of notation. The set of interpretations over a signature Σ is denoted as Ω_Σ. Interpretations are also called *worlds* and Ω_Σ is called the *universe*. An interpretation $\omega \in \Omega_\Sigma$ is a *model* of a formula $A \in \mathcal{L}_\Sigma$ if A holds in ω. This is denoted as $\omega \models A$. The set of models of a formula (over a signature Σ) is denoted as $\mathit{Mod}_\Sigma(A) = \{\omega \in \Omega_\Sigma \mid \omega \models A\}$. A formula A *entails* a formula B if $\mathit{Mod}_\Sigma(A) \subseteq \mathit{Mod}_\Sigma(B)$.

A *conditional* $(B|A)$ connects two formulas A, B and represents the rule "If A then usually B". For a conditional $(B|A)$ the formula A is called the *antecedent* and the formula B the *consequent* of the conditonal. The conditional language over a signature Σ is denoted as $(\mathcal{L}|\mathcal{L})_\Sigma = \{(B|A) \mid A, B \in \mathcal{L}_\Sigma\}$. $(\mathcal{L}|\mathcal{L})_\Sigma$ is a flat conditional language as it does not allow nesting conditionals. A finite set of conditionals is called a *conditional belief base*.

We use a three-valued semantics of conditionals in this paper (de Finetti 1937). For a world ω a conditional $(B|A)$ is either *verified* by ω if $\omega \models AB$, *falsified* by ω if $\omega \models A\overline{B}$, or *not applicable* to ω if $\omega \models \overline{A}$. Conditionals are usually considered in the context of epistemic states. An *epistemic state* is a structure that represents all beliefs that are relevant for an agent's reasoning.

There exist different kinds of models for epistemic states that can handle conditionals. Two approaches to this are ranking functions (ordinal conditional functions, OCFs) and total preorders (TPOs) on possible worlds.

A *ranking function* (Spohn 1988), also called *ordinal conditional function* (OCF), is a function $\kappa : \Omega_\Sigma \to \mathbb{N}_0 \cup \{\infty\}$ such that $\kappa^{-1}(0) \neq \emptyset$. The intuition of a ranking function is that the rank of a world is lower if the world is more plausible. Therefore, ranking functions can be seen as some kind of "implausibility measure". Ranking functions are extended to formulas by $\kappa(A) = \min_{\omega \in \mathit{Mod}(A)} \kappa(\omega)$ with $\min_\emptyset(\ldots) = \infty$. A ranking function κ models a conditional $(B|A)$, denoted as $\kappa \models (B|A)$ if $\kappa(AB) < \kappa(A\overline{B})$, i.e., if the verification of the conditional is strictly more plausible than its falsification. A ranking function κ models a conditional belief set $\mathcal{R}$, denoted as $\kappa \models \mathcal{R}$ if $\kappa \models r$ for every $r \in \mathcal{R}$. The uniform ranking function κ_{uni} with $\kappa_{\mathrm{uni}}(\omega) = 0$ for every $\omega \in \mathit{Mod}_\Sigma(A)$ represents the state of complete ignorance.

A *total preorder* (TPO) is a total, reflexive, and transitive binary relation. The meaning of a total preorder $\preceq$ on Ω_Σ as model for an epistemic state is that ω_1 is at least as plausible as ω_2 if $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in \Omega_\Sigma$. Total preorders on worlds are extended to formulas by $A \preceq B$ if $\min(\mathit{Mod}_\Sigma(A), \preceq) \preceq \min(\mathit{Mod}_\Sigma(B), \preceq)$. A total preorder $\preceq$ models a conditional $(B|A)$, denoted as $\preceq\ \models (B|A)$ if $AB \prec A\overline{B}$, i.e., if the verification of the conditional is strictly more plausible than its falsification. A total preorder $\preceq$ models a conditional belief set $\mathcal{R}$, denoted as $\preceq\ \models \mathcal{R}$ if $\preceq\ \models r$ for every $r \in \mathcal{R}$.

3 Marginalization, Conditionalization, Syntax Splitting

We want to consider transformations among models of epistemic states represented by ranking functions or total preorders. To establish a notion for the domain of such transformations, we define the sets $\mathcal{M}_{TPO}(\Sigma)$ and $\mathcal{M}_{OCF}(\Sigma)$ containing all models over a certain signature.

Definition 1. *Let Σ be a signature.*

$$\mathcal{M}_{TPO}(\Sigma) = \{\preceq\ \subseteq \Omega_\Sigma \times \Omega_\Sigma \mid \preceq \textit{ total preorder over } \Omega_\Sigma\}$$
$$\mathcal{M}_{OCF}(\Sigma) = \{\kappa : \Omega_\Sigma \mapsto \mathbb{N}_0 \cup \{\infty\} \mid \kappa \textit{ ranking function}\}$$

3.1 Marginalization and Conditionalization on TPOs and OCFs

Two operations on epistemic states that we will use in this paper are marginalization and conditionalization. Marginalization restricts the epistemic state to a sub-signature of the original signature.

Definition 2 (marginalization of ranking functions (Spohn 1988; Beierle and Kern-Isberner 2012))**.** *The* marginalization *of ranking functions from a signature Σ to a sub-signature $\Sigma' \subseteq \Sigma$ is a function $\mathcal{M}_{OCF}(\Sigma) \to \mathcal{M}_{OCF}(\Sigma')$, $\kappa \mapsto \kappa_{|\Sigma'}$ such that $\kappa_{|\Sigma'}(\omega) = \kappa(\omega)$ for $\omega \in \Omega_{\Sigma'}$.*

Definition 3 (marginalization of total preorders (Beierle and Kern-Isberner 2012; Kern-Isberner and Brewka 2017))**.** *The* marginalization *of total preorders from a signature Σ to a sub-signature $\Sigma' \subseteq \Sigma$ is a function $\mathcal{M}_{TPO}(\Sigma) \to \mathcal{M}_{TPO}(\Sigma')$, $\preceq\ \mapsto\ \preceq_{|\Sigma'}$ such that $\omega_1 \preceq_{|\Sigma'} \omega_2$ iff $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in \Omega_{\Sigma'}$.*

Note that a world ω over a sub-signature $\Sigma' \subseteq \Sigma$ is considered as a formula when evaluated in the context of Σ.

The marginalizations of OCFs and TPOs presented above are special cases of general forgetful functors $Mod(\varrho)$ from Σ-models to Σ'-models given in (Beierle and Kern-Isberner 2012) where $\Sigma' \subseteq \Sigma$ and ϱ is the inclusion from Σ' to Σ. Informally, a forgetful functor forgets everything about the interpretation of the symbols in $\Sigma \setminus \Sigma'$ when mapping a Σ-model to a Σ'-model.

Conditionalization on the other hand restricts the set of worlds that are considered in an epistemic state. After the conditionalization with a formula A the resulting state only considers the elements of $Mod_\Sigma(A)$ as possible worlds. To capture the outcome of a conditionalization, we extend the notion of ranking functions and total preorders.

Definition 4 (restricted ranking function)**.** *A* restricted ranking function *over a set $M \subseteq \Omega_\Sigma$ is a function $\kappa : M \to \mathbb{N}_0 \cup \{\infty\}$ such that $\kappa^{-1}(0) \neq \emptyset$. Restricted ranking functions are extended to formulas by $\kappa(A) = \min_{\omega \in Mod(A) \cap M} \kappa(\omega)$ with $\min_\emptyset(\ldots) = \infty$.*

A total preorder $\preceq$ on a set $M \subseteq \Omega_\Sigma$ as model for an epistemic state is also called a *restricted total preorder*. Its intuition is the same as that of usual total preorders: ω_1 is at least as plausible as ω_2 if $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in M$. Restricted total preorders on worlds are extended to formulas by $A \preceq B$ if $\min(Mod_\Sigma(A) \cap M, \preceq) \preceq \min(Mod_\Sigma(B) \cap M, \preceq)$.

Definition 5. *Let Σ be a signature and $A \in \mathcal{L}_\Sigma$. Let $M_A = Mod_\Sigma(A)$.*

$$\mathcal{M}_{TPO}(\Sigma, A) = \{\preceq \subseteq M_A \times M_A \mid \preceq \textit{ total preorder over } M_A\}$$

$$\mathcal{M}_{OCF}(\Sigma, A) = \{\kappa : M_A \mapsto \mathbb{N}_0 \cup \{\infty\} \mid \kappa \textit{ restricted ranking function}\}$$

The restricted OCFs and TPOs properly include the original notation: $\mathcal{M}_I(\Sigma) = \mathcal{M}_I(\Sigma, \top)$ for $I \in \{TPO, OCF\}$. For $\Psi \in \mathcal{M}_I(\Sigma, A)$, we call $sig(\Psi) = \Sigma$ the signature of Ψ and $dom(\Psi) = Mod_\Sigma(A)$ the domain of Ψ. Now we can define conditionalization using restricted OCFs/TPOs.

Definition 6 (conditionalization of ranking functions (Spohn 1988; Sezgin and Kern-Isberner 2020))**.** *The* conditionalization *of ranking functions over a signature Σ to the models of a formula $A \in \mathcal{L}_\Sigma$ is a function $\mathcal{M}_{OCF}(\Sigma) \to \mathcal{M}_{OCF}(\Sigma, A)$, $\kappa \mapsto \kappa|A$ such that $\kappa|A(\omega) = \kappa(\omega) - \kappa(A)$ for $\omega \in Mod_\Sigma(A)$.*

A notion of conditionalization for TPOs with respect to a formula A where the models of $\overline{A}$ are shifted to the uppermost layer has been introduced in (Kern-Isberner, Beierle, and Brewka 2020). Here, we will use the following concept of TPO conditionalization where the models of $\overline{A}$ are removed entirely from the TPO by conditionalization.

Definition 7 (conditionalization of total preorders)**.** *The* conditionalization *of total preorders over a signature Σ to the models of a formula $A \in \mathcal{L}_\Sigma$ is a function $\mathcal{M}_{TPO}(\Sigma) \to \mathcal{M}_{TPO}(\Sigma, A)$, $\preceq \; \mapsto \; \preceq|A$ such that $\omega_1 \; (\preceq|A) \; \omega_2$ iff $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in Mod_\Sigma(A)$.*

Note that Definitions 6 and 7 for conditionalization integrate nicely with our notions of restricted TPOs and restricted OCFs, because models of $\overline{A}$ occur neither in the elements of $\mathcal{M}_{OCF}(\Sigma, A)$ nor $\mathcal{M}_{TPO}(\Sigma, A)$.

3.2 Marginalization and Conditionalization on Restricted TPOs and OCFs

While originally, both marginalization and conditionalization were defined on OCFs/TPOs with the full set of Σ-models, we will also consider the iterative case. Therefore, we extend the definitions of these operations to cover already conditionalized states (which are restricted TPOs/OCFs).

Definition 8 (marginalization of restricted OCFs)**.** *The* marginalization *of ranking functions over $Mod_\Sigma(A)$ from signature Σ to a sub-signature $\Sigma' \subseteq \Sigma$ is a function $\mathcal{M}_{OCF}(\Sigma, A) \to \mathcal{M}_{OCF}(\Sigma', A)$, $\kappa \mapsto \kappa_{|\Sigma'}$ such that $\kappa_{|\Sigma}(\omega) = \kappa(\omega)$ for $\omega \in Mod_{\Sigma'}(A)$.*

Definition 9 (marginalization of restricted TPOs)**.** *The* marginalization *of total preorders over $Mod_\Sigma(A)$ from signature Σ to a sub-signature $\Sigma' \subseteq \Sigma$ is a function $\mathcal{M}_{TPO}(\Sigma, A) \to \mathcal{M}_{TPO}(\Sigma', A)$, $\preceq \; \mapsto \; \preceq_{|\Sigma'}$ such that $\omega_1 \preceq_{|\Sigma'} \omega_2$ iff $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in Mod_{\Sigma'}(A)$.*

Definition 10 (conditionalization of restricted OCFs)**.** *The* conditionalization *of ranking functions over $Mod_\Sigma(B)$ to the models of a formula $A \in \mathcal{L}_\Sigma$ is a function $\mathcal{M}_{OCF}(\Sigma, B) \to \mathcal{M}_{OCF}(\Sigma, A \wedge B)$, $\kappa \mapsto \kappa|A$ such that $\kappa|A(\omega) = \kappa(\omega) - \kappa(A)$ for $\omega \in Mod_\Sigma(A \wedge B)$.*

Definition 11 (conditionalization of restricted TPOs)**.** *The* conditionalization *of total preorders over $Mod_\Sigma(B)$ to the models of a formula $A \in \mathcal{L}_\Sigma$ is a function $\mathcal{M}_{TPO}(\Sigma, B) \to \mathcal{M}_{TPO}(\Sigma, A \wedge B)$, $\preceq \; \mapsto \; \preceq|A$ such that $\omega_1 \preceq|A \; \omega_2$ iff $\omega_1 \preceq \omega_2$ for $\omega_1, \omega_2 \in Mod_\Sigma(A \wedge B)$.*

Because for $\mathcal{M}_I(\Sigma, \top)$, the marginalization/conditionalization of the restricted OCFs/TPOs coincides with the marginalization/conditionalization of OCFs/TPOs, Definitions 8, 9, 10, and 11 of marginalization and conditionalization properly cover and extend the Definitions 2, 3, 6, and 7.

3.3 Syntax Splitting

An interesting feature of a ranking function or a total preorder is if they allow for syntax splittings. Syntax splitting was first introduced as property of belief sets in (Parikh 1999). The basic idea is that a belief set contains independent information over different parts of the signature. The partition of the signature in these parts is called a syntax splitting for the considered belief set. Syntax splittings are useful properties of epistemic states, as they indicate that different parts of the state can be processed independently of each other.

The notion of syntax splitting was extended to other representations of epistemic states such as ranking functions in (Kern-Isberner and Brewka 2017). For a partitioning $\Sigma = \Sigma_1 \,\dot\cup\, \cdots \,\dot\cup\, \Sigma_n$ of a signature Σ and a world $\omega \in \Omega_\Sigma$, the world $\omega^j \in \Omega_{\Sigma_j}$ denotes the variable assignment of the variables in Σ_j as in ω in the following definitions.

Definition 12 (syntax splitting for total preorders (Kern-Isberner and Brewka 2017))**.** *Let $\preceq$ be a total preorder over a signature Σ. Let $\Sigma = \Sigma_1 \,\dot\cup\, \cdots \,\dot\cup\, \Sigma_n$ be a partitioning and $\omega^{\neq i} := \bigwedge_{\substack{j=1,\ldots,n \\ i \neq j}} \omega^j$ for $\omega \in \Omega$ and $i = 1, \ldots, n$. The*

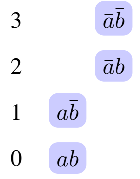


Figure 1: Example for an OCF κ with the syntax splitting $\{a\} \dot{\cup} \{b\}$
.

partitioning $\Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a syntax splitting *for $\preceq$ if, for $i = 1, \ldots, n$,*

$$\omega_1^{\neq i} = \omega_2^{\neq i} \quad \textit{implies} \quad \left(\omega_1 \preceq \omega_2 \textit{ iff } \omega_1^i \preceq_{|\Sigma_i} \omega_2^i\right).$$

Definition 13 (syntax splitting for ranking functions (Kern-Isberner and Brewka 2017))**.** *Let κ be a ranking function over Σ.*

A partitioning $\Sigma = \Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a syntax splitting *for κ if there are ranking functions $\kappa_i : \Sigma_i \mapsto \mathbb{N}_0 \cup \{\infty\}$ for $i = 1, \ldots, n$ such that $\kappa(\omega) = \kappa_1(\omega^1) + \cdots + \kappa_n(\omega^n)$. This is denoted as $\kappa = \kappa_1 \oplus \cdots \oplus \kappa_n$.*

The notion of syntax splitting can be extended to restricted ranking functions and total preorders.

Definition 14 (syntax splitting for restricted TPOs)**.** *Let $\preceq$ be a total preorder in $\mathcal{M}_{TPO}(\Sigma, A)$. Let $\Sigma = \Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ be a partitioning, ω^j be the variable assignment of the variables in Σ_j as in ω, and $\omega^{\neq i} := \bigwedge_{\substack{j=1,\ldots,n \\ i \neq j}} \omega^j$ for $\omega \in \Omega$ and $i = 1, \ldots, n$. The partitioning $\Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a* syntax splitting *for $\preceq$ if*

- *there are formulas $A_1, \ldots, A_n$ such that $A \equiv A_1 \wedge \cdots \wedge A_n$ and $A_i \in \Sigma_i$ for $i = 1, \ldots, n$*
- *and, for $i = 1, \ldots, n$ and $\omega_1, \omega_2 \in dom(\preceq)$,*

$$\omega_1^{\neq i} = \omega_2^{\neq i} \quad \textit{implies} \quad \left(\omega_1 \preceq \omega_2 \textit{ iff } \omega_1^i \preceq_{|\Sigma_i} \omega_2^i\right).$$

Definition 15 (syntax splitting for restricted OCFs)**.** *Let κ be a ranking function in $\mathcal{M}_{OCF}(\Sigma, A)$. Let ω^j be the variable assignment of the variables in Σ_j as in ω. A partitioning $\Sigma = \Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a* syntax splitting *for κ if*

- *there are formulas $A_1, \ldots, A_n$ such that $A \equiv A_1 \wedge \cdots \wedge A_n$ and $A_i \in \Sigma_i$ for $i = 1, \ldots, n$*
- *and there are ranking functions $\kappa_i \in \mathcal{M}_{OCF}(\Sigma_i, A_i)$ for $i = 1, \ldots, n$ such that $\kappa(\omega) = \kappa_1(\omega^1) + \cdots + \kappa_n(\omega^n)$ for $\omega \in dom(\kappa)$.*

This is denoted as $\kappa = \kappa_1 \oplus \cdots \oplus \kappa_n$.

Again, the definitions of syntax splitting for restricted total preorders or ranking functions are compatible with the definition of syntax splitting for TPOs/OCFs.

Example 1. *The ranking function κ over $\Sigma = \{a, b\}$ displayed in Figure 1 and the total preorder induced by κ both have the syntax splitting $\{a\} \dot{\cup} \{b\}$.*

Analoguosly to Parikh's (P), postulates for revision and contraction of total preorders and ranking functions, that are based on the notions of syntax splitting from Definitions 12 and 13, have been introduced and investigated in (Kern-Isberner and Brewka 2017), (Haldimann, Kern-Isberner, and Beierle 2020), and (Haldimann, Beierle, and Kern-Isberner 2021).

4 Postulates for Mappings on Epistemic States

To formalize the transformations among models of epistemic states we introduce epistemic state mappings.

Definition 16. *Let $I_1, I_2 \in \{TPO, OCF\}$. An* epistemic state mapping *from I_1 to I_2, denoted as $\xi : I_1 \rightsquigarrow I_2$, is a function family $\xi = (\xi_{\Sigma,A})$ for signatures Σ and formulas $A \in \mathcal{L}_\Sigma$ with $\xi_{\Sigma,A} : \mathcal{M}_{I_1}(\Sigma, A) \to \mathcal{M}_{I_2}(\Sigma, A)$ such that $A \equiv B$ implies $\xi_{\Sigma,A} = \xi_{\Sigma,B}$.*

Example 2. *The family of functions $\xi^{reverse}$ that reverses every TPO defined by $\xi^{reverse}_{\Sigma,A}(\preceq) = \preceq'$ with $\omega_1 \preceq' \omega_2$ iff $\omega_2 \preceq \omega_1$ for a signature Σ, $A \in \mathcal{L}_\Sigma$, $\preceq \in \mathcal{M}_{TPO}(\Sigma, A)$, and $\omega_1, \omega_2 \in Mod_\Sigma(A)$ is an epistemic state mapping from TPOs to TPOs.*

The family of functions τ that maps every OCF to the TPO induced by it, defined by $\tau_{\Sigma,A}(\kappa) = \leq_\kappa$ for a signature Σ, $A \in \mathcal{L}_\Sigma$, and $\kappa \in \mathcal{M}_{OCF}(\Sigma, A)$ is an epistemic state mapping from TPOs to OCFs.

Every epistemic state mapping represents a way to transform epistemic states of kind I_1 to epistemic states of kind I_2 for different domains. Desirable properties of epistemic state mappings $(\xi_{\Sigma,A})$ can be stated in form of postulates.

Some of these postulates use the fact that both ranking functions and total preorder induce a total preorder on their domain. For a total preorder $\preceq$, the induced order $\leq_\preceq$ is the order $\preceq$ itself. For a ranking function κ, the induced ordering $\leq_\kappa$ is given by $\omega_1 \leq_\kappa \omega_2$ iff $\kappa(\omega_1) \leqslant \kappa(\omega_2)$ for $\omega_1, \omega_2 \in dom(\kappa)$.

Postulates. *Let $I_1, I_2 \in \{TPO, OCF\}$ and let $(\xi_{\Sigma,A})$ be an epistemic state mapping from I_1 to I_2. Let Σ be a signature and $A \in \mathcal{L}_\Sigma$, and $\Psi \in \mathcal{M}_{I_1}(\Sigma, A)$.*

Let $(C|D) \in (\mathcal{L} \mid \mathcal{L})_\Sigma$.

(IE) $\Psi \models (C|D)$ *iff* $\xi_{\Sigma,A}(\Psi) \models (C|D)$.

(wIE$^\Rightarrow$) $\Psi \models (C|D)$ *implies* $\xi_{\Sigma,A}(\Psi) \models (C|D)$.

(wIE$^\Leftarrow$) $\xi_{\Sigma,A}(\Psi) \models (C|D)$ *implies* $\Psi \models (C|D)$.

Let ω_1, ω_2 in $dom(\Psi)$.

(Ord) $\omega_1 <_\Psi \omega_2$ *iff* $\omega_1 <_{\xi_{\Sigma,A}(\Psi)} \omega_2$.

(wOrd$^\Rightarrow$) $\omega_1 <_\Psi \omega_2$ *implies* $\omega_1 <_{\xi_{\Sigma,A}(\Psi)} .\omega_2$

(wOrd$^\Leftarrow$) $\omega_1 <_{\xi_{\Sigma,A}(\Psi)} \omega_2$ *implies* $\omega_1 <_\Psi \omega_2$.

(SynSplit) *If $sig(\Psi) = \Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a syntax splitting for Ψ, then $\Sigma_1 \dot{\cup} \cdots \dot{\cup} \Sigma_n$ is a syntax splitting for $\xi_{\Sigma,A}(\Psi)$.*

(SynSplit$^\mathbf{b}$) *If $\Sigma = \Sigma_1 \dot{\cup} \Sigma_2$ is a syntax splitting for Ψ, then $\Sigma_1 \dot{\cup} \Sigma_2$ is a syntax splitting for $\xi_{\Sigma,A}(\Psi)$.*

Let $\Sigma' \subseteq \Sigma$ with $\Sigma' \neq \emptyset$ and $A' \in \mathcal{L}_{\Sigma'}$ such that $Mod_{\Sigma'}(A') = \{\omega' \mid \omega \in Mod_\Sigma(A)\}$ where ω' is the assignment of the variables in Σ' as in ω.

$$\begin{tikzcd}
\mathcal{M}_I(\Sigma, A) \arrow[r, "\cdot_{|\Sigma'}"] \arrow[d, "\xi"'] & \mathcal{M}_I(\Sigma', A) \arrow[d, "\xi"] \\
\mathcal{M}_I(\Sigma, A) \arrow[r, "\cdot_{|\Sigma'}"'] & \mathcal{M}_I(\Sigma', A)
\end{tikzcd}$$

(a) Illustration of (Marg).

$$\begin{tikzcd}
\mathcal{M}_I(\Sigma, A) \arrow[r, "\cdot|F"] \arrow[d, "\xi"'] & \mathcal{M}_I(\Sigma, A \wedge F) \arrow[d, "\xi"] \\
\mathcal{M}_I(\Sigma, A) \arrow[r, "\cdot|F"'] & \mathcal{M}_I(\Sigma, A \wedge F)
\end{tikzcd}$$

(b) Illustration of (Cond).

Figure 2: A commutative diagram illustrating the postulates (Cond) and (Marg).

(Marg) $\xi_{\Sigma',A'}(\Psi_{|\Sigma'}) = \xi_{\Sigma,A}(\Psi)_{|\Sigma'}$

Let $F \in \mathcal{L}_\Sigma$ with $Mod_\Sigma(F) \cap dom(\Psi) \neq \emptyset$.

(Cond) $\xi_{\Sigma,A\wedge F}(\Psi|F) = \xi_{\Sigma,A}(\Psi)|F$

The postulate (IE) requires *inferential equivalence* and states that the epistemic state mapping may not change the set of conditionals accepted by an epistemic state. The epistemic state and its mapping induce the same inference relation with respect to conditionals. This is a quite strong postulate, the postulates (wIE$^{\Rightarrow}$) and (wIE$^{\Leftarrow}$) are weaker versions of (IE). Postulate (wIE$^{\Rightarrow}$) states that an epistemic state mapping may not remove conditionals from the set of inferred conditionals. Postulate (wIE$^{\Leftarrow}$) states that after an epistemic state mapping, we may not accept additional conditionals.

The postulate (Ord) expresses the postulate (IE) in terms of the induced total preorders of the epistemic states. Analogously (wOrd$^{\Rightarrow}$) and (wOrd$^{\Leftarrow}$) represent (wIE$^{\Rightarrow}$) and (wIE$^{\Leftarrow}$), respectively.

(SynSplit) states that an epistemic state mapping should preserve syntax splittings of the epistemic state. (SynSplit$^{\text{b}}$) is a special case of (SynSplit) for syntax splittings in two sub-signatures.

The postulate (Marg) ensures the compatibility of an epistemic state mapping with marginalization. It states that changing the order in which marginalization and the epistemic state mapping are applied does not matter. This postulate is illustrated in Figure 2a. Similarly, the postulate (Cond) ensures the compatibility of an epistemic state mapping with conditionalization. (Cond) is illustrated in Figure 2b.

It is easy to see that (IE) is equivalent to the conjunction of (wIE$^{\Rightarrow}$) and (wIE$^{\Leftarrow}$) and that (Ord) is equivalent to the conjunction of (wOrd$^{\Rightarrow}$) and (wOrd$^{\Leftarrow}$). Other relationships among the postulates, as the following, are less obvious.

Proposition 1. *The following relationships hold between the introduced postulates:*

1. *(IE) is equivalent to (Ord).*
2. *(wIE$^{\Rightarrow}$) is equivalent to (wOrd$^{\Rightarrow}$).*
3. *(wIE$^{\Leftarrow}$) is equivalent to (wOrd$^{\Leftarrow}$).*

Proof. Let $\xi : I_1 \rightsquigarrow I_2$ be a epistemic state mapping with $I_1, I_2 \in \{TPO, OCF\}$.
Ad (2): "$\Leftarrow$" Let $(\xi_{\Sigma,A})$ satisfy (wOrd$^{\Rightarrow}$). Let $\Psi \in \mathcal{M}_{I_1}(\Sigma, A)$ and $\Phi = \xi_{\Sigma,A}(\Psi)$. If $\Psi \models (D|C)$, then $\min(Mod_\Sigma(CD), \lessdot_\Psi) \lessdot_\Psi \min(Mod_\Sigma(C\overline{D}), \lessdot_\Psi)$. In this case, (wOrd$^{\Rightarrow}$) implies $\min(Mod_\Sigma(CD), \lessdot_\Phi) \lessdot_\Phi \min(Mod_\Sigma(C\overline{D}), \lessdot_\Phi)$. This is equivalent to $\Phi \models (C|D)$. Therefore, $(\xi_{\Sigma,A})$ satisfies (wIE$^{\Rightarrow}$).

"$\Rightarrow$" Let $(\xi_{\Sigma,A})$ satisfy (wIE$^{\Rightarrow}$). Let $\Psi \in \mathcal{M}_{I_1}(\Sigma, A)$ and $\Phi = \xi_{\Sigma,A}(\Psi)$. Let $\omega_1, \omega_2 \in \Omega$ with $\omega_1 \lessdot_\Psi \omega_2$. Then, $\Psi \models (\omega_1|\omega_1 \vee \omega_2)$. (wIE$^{\Rightarrow}$) implies that $\Phi \models (\omega_1|\omega_1 \vee \omega_2)$. Therefore, $\omega_1 \lessdot_\Phi \omega_2$. We see that $(\xi_{\Sigma,A})$ satisfies (wOrd$^{\Rightarrow}$).
Ad (3): "$\Leftarrow$" Let $(\xi_{\Sigma,A})$ satisfy (wOrd$^{\Leftarrow}$). Let $\Psi \in \mathcal{M}_{I_1}(\Sigma, A)$ and $\Phi = \xi_{\Sigma,A}(\Psi)$. If $\Phi \models (D|C)$, then $\min(Mod_\Sigma(CD), \lessdot_\Phi) \lessdot_\Phi \min(Mod_\Sigma(C\overline{D}), \lessdot_\Phi)$. In this case, (wOrd$^{\Leftarrow}$) implies $\min(Mod_\Sigma(CD), \lessdot_\Psi) \lessdot_\Psi \min(Mod_\Sigma(C\overline{D}), \lessdot_\Psi)$. This is equivalent to $\Psi \models (D|C)$. Therefore, $(\xi_{\Sigma,A})$ satisfies (wIE$^{\Leftarrow}$).

"$\Rightarrow$" Let $(\xi_{\Sigma,A})$ satisfy (wIE$^{\Leftarrow}$). Let $\Psi \in \mathcal{M}_{I_1}(\Sigma, A)$ and $\Phi = \xi_{\Sigma,A}(\Psi)$. Let $\omega_1, \omega_2 \in \Omega$ with $\omega_1 \lessdot_\Phi \omega_2$. Then, $\Phi \models (\omega_1|\omega_1 \vee \omega_2)$. (wIE$^{\Leftarrow}$) implies that $\Psi \models (\omega_1|\omega_1 \vee \omega_2)$. Therefore, $\omega_1 \lessdot_\Psi \omega_2$. We see that $(\xi_{\Sigma,A})$ satisfies (wOrd$^{\Leftarrow}$).
Ad (1): This follows from (2) and (3) as (IE) is the conjunction of (wIE$^{\Rightarrow}$) and (wIE$^{\Leftarrow}$) and (Ord) is the conjunction of (wOrd$^{\Rightarrow}$) and (wOrd$^{\Leftarrow}$). □

In the next sections, we will investigate the introduced postulates further for specific combinations of I_1 and I_2.

5 Mapping Total Preorders to Total Preorders

Let us first consider epistemic state mappings from total preorders to total preorders. If we want (IE) or the equivalent (Ord) to hold, we do not have much choice.

Proposition 2. *The only epistemic state mapping from TPOs to TPOs that fulfils (Ord) is the identity.*

From Proposition 2 it follows that (IE) or (Ord) imply (SynSplit), (Cond), and (Marg) for epistemic state mappings from TPOs to TPOs as the identity fulfils these postulates.

But what if we require only (wIE$^{\Rightarrow}$) or (wIE$^{\Leftarrow}$)? To better understand these postulates, we can think of a total preorder $\preceq$ as a stack of "layers". We say that two worlds $\omega_1, \omega_2 \in dom(\preceq)$ have the same position in $\preceq$, denoted as $\omega_1 \approx_\preceq \omega_2$, if $\omega_1 \preceq \omega_2$ and $\omega_1 \preceq \omega_2$. The relation $\approx_\preceq$ is an equivalence relation and layers are the equivalence classes of $\approx_\preceq$ on $dom(\preceq)$. I.e., two worlds $\omega_1, \omega_2 \in dom(\preceq)$ are in the same layer if they have the same position in the TPO. The layers are stacked according to the TPO: the lower a layer is, the smaller the worlds in it are with respect to $\preceq$.

A consequence of each (wIE$^{\Rightarrow}$) and (wIE$^{\Leftarrow}$) is that we cannot swap parts of different layers. If $\omega_1 \prec \omega_2$ then it is not possible that $\omega_2 \prec' \omega_1$ where $\preceq' = \xi(\preceq)$ if ξ fulfils either (wIE$^{\Rightarrow}$) or (wIE$^{\Leftarrow}$).

(wOrd$^{\Rightarrow}$) allows the "splitting" of layers. For ω_1, ω_2 with $\omega_1 \approx_{\preceq} \omega_2$ we may have an epistemic state mapping ξ fulfilling (wOrd$^{\Rightarrow}$) with $\omega_1 \prec' \omega_2$ where $\preceq' = \xi(\preceq)$. Thus, (wOrd$^{\Rightarrow}$) allows to extend the set of accepted conditionals as stated in the equivalent (wIE$^{\Rightarrow}$). However, the opposite direction is not allowed: An epistemic state mapping fulfilling (wOrd$^{\Rightarrow}$) may not merge parts of different layers together.

(wOrd$^{\Leftarrow}$) is the opposite. For ω_1, ω_2 with $\omega_1 \prec \omega_2$ we may have an epistemic state mapping ξ fulfilling (wOrd$^{\Leftarrow}$) with $\omega_1 \approx_{\xi(\preceq)} \omega_2$, i.e., merging of layers is allowed. Note that in this case we have $\omega_1 \approx_{\xi(\preceq)} \omega_3 \approx_{\xi(\preceq)} \omega_2$ for any $\omega_1 \preceq \omega_3 \preceq \omega_2$. Thus, (wOrd$^{\Leftarrow}$) allows to reduce the set of accepted conditionals which can be seen as a form of forgetting. (wOrd$^{\Leftarrow}$) does not allow splitting of layers.

6 Mapping Ranking Functions to Ranking Functions

Let us consider the case where we map ranking functions to ranking functions. For such epistemic state mappings all postulates are compatible, in the sense that all postulates can be satisfied simultaneously by some epistemic state mapping.

Proposition 3. *The epistemic state mapping $\xi : OCF \rightsquigarrow OCF, \kappa \mapsto a \cdot \kappa$ for some $a \in \mathbb{N}^+$ fulfils (Ord), (SynSplit), (Cond), and (Marg).*

To understand the meaning of (Ord), (wOrd$^{\Rightarrow}$), and (wOrd$^{\Leftarrow}$), we can use the concept of layers introduced in the previous section for total preorders. For a ranking function κ, each layer contains the worlds in $\kappa^{-1}(k)$ for a $k \in \mathbb{N}_0$. Contrary to total preorders, ranking functions can have empty layers. This empty layers (or the lack thereof) make ranking functions more expressive than total preorders.

The implications of (Ord) for epistemic state mappings from OCFs to OCFs are similar to the implications for epistemic state mappings from TPOs to TPOs in terms of layers. The layers are not swapped, split, or merged by the epistemic state mapping. (Ord) allows for adding or removing empty layers. For example, the epistemic state mapping that removes all empty layers beneath a non-empty layer fulfils (Ord).

In contrast to (Ord), the postulate (wOrd$^{\Rightarrow}$) allows splitting of layers. If two worlds have the same rank in a ranking function κ they may have different ranks in $\xi(\kappa)$. But (wOrd$^{\Rightarrow}$) prevents merging different layers. If two worlds have different ranks in a ranking function κ before the epistemic state mapping, they may not have the same rank in $\xi(\kappa)$.

The postulate (wOrd$^{\Leftarrow}$) allows merging but not splitting of layers. If two worlds ω_1, ω_2 have different ranks in κ they may have the same rank in $\kappa' = \xi(\kappa)$ without violating (wOrd$^{\Leftarrow}$). In this case it holds that $\kappa'(\omega_1) = \kappa'(\omega_2) = \kappa'(\omega_3)$ for any world ω_3 with $\kappa(\omega_1) \leqslant \kappa(\omega_3) \leqslant \kappa(\omega_2)$.

7 Mapping Ranking Functions to Total Preorders

In this section, we want to investigate epistemic state mappings from ranking functions to total preorders on worlds.

Proposition 4 (τ^*)**.** *There is a unique epistemic state mapping $\tau^* : OCF \rightsquigarrow TPO$ fulfilling (IE).*

Proof. Let κ be any ranking function. (Ord) states, that $\preceq = \xi(\kappa)$ induces the same ranking function as κ. As the total preorder induced by a total preorder is the total preorder itself, the only epistemic state mapping from ranking functions to total preorders fulfilling (IE) is

$$\tau^* : OCF \rightsquigarrow TPO, \quad \kappa \mapsto \leq_{\kappa}.$$

□

In the following, we will investigate the properties of the epistemic state mapping τ^*. τ^* is injective: For a given total preorder $\preceq$ it is easy to construct a ranking function κ such that $\preceq = \tau^*(\kappa)$. But τ^* is not surjective as there are more ranking functions than total preorders for any given (non-empty) signature.

The transformation τ^* preserves syntax splittings of the ranking function.

Proposition 5. *τ^* fulfils (SynSplit).*

Proof. Let $\kappa = \kappa_1 \oplus \cdots \oplus \kappa_n$ be a ranking function over Σ with a syntax splitting $\Sigma = \Sigma_1 \dot\cup \cdots \dot\cup \Sigma_n$. Let $\preceq = \tau^*(\kappa)$. Let $i \in \{1, \ldots, n\}$ and $\omega_1, \omega_2 \in \Omega_\Sigma$ with $\omega_1^{\neq i} = \omega_2^{\neq i}$ and $\omega_1 \preceq \omega_2$. Because τ^* fulfils (O2T$^{\text{order}}$), we have $\kappa(\omega_1) \leqslant \kappa(\omega_2)$. The syntax splitting on κ and $\omega_1^{\neq i} = \omega_2^{\neq i}$ implies $\kappa_i(\omega_1^i) \leqslant \kappa_i(\omega_2^i)$. This and the syntax splitting on κ implies $\omega_1^i \preceq_{|\Sigma_i} \omega_2^i$.

Thus, $\Sigma_1 \dot\cup \cdots \dot\cup \Sigma_n$ is a syntax splitting for $\preceq$. □

As a direct implication of this, τ^* fulfils (SynSplit$^{\text{b}}$). However, the transformation may introduce *new* syntax splittings as the following example shows.

Example 3. *Let $\Sigma = \{a, b\}$ and κ_1, κ_2 be ranking functions over Σ such that*

$$\kappa_1(ab) = 0 \quad \kappa_1(a\overline{b}) = 1 \quad \kappa_1(\overline{a}b) = 1 \quad \kappa_1(\overline{a}\overline{b}) = 2$$
$$\kappa_2(ab) = 0 \quad \kappa_2(a\overline{b}) = 1 \quad \kappa_2(\overline{a}b) = 1 \quad \kappa_2(\overline{a}\overline{b}) = 3$$

κ_1 has the syntax splitting $\{a\} \dot\cup \{b\}$, κ_2 has not. Both ranking functions are mapped to the total preorder $ab \prec a\overline{b}, \overline{a}b \prec \overline{a}\overline{b}$ by τ^* which has the syntax splitting $\{a\} \dot\cup \{b\}$.

Furthermore, the example shows that there cannot be a notion of syntax splitting for total preorders such that for every ranking function κ the total preorder $\tau^*(\kappa)$ has a syntax splitting *if and only if* κ has a syntax splitting.

The function τ^* behaves nicely with respect to marginalization and conditionalization.

Proposition 6. *τ^* fulfils (Marg).*

Proof. Let $\kappa \in \mathcal{M}_{OCF}(\Sigma, A)$ be an OCF and $\Sigma_1 \subseteq \Sigma$. Let $\preceq_1 = \tau^*(\kappa_{|\Sigma_1})$ and $\preceq_2 = \tau^*(\kappa)$. Let $\omega_a, \omega_b \in \Omega_{\Sigma_1}$.

$$\begin{aligned}
& & \omega_a \ &\preceq_1 \ \omega_b \\
\Leftrightarrow & & \kappa_{|\Sigma_1}(\omega_a) \ &\leqslant \ \kappa_{|\Sigma_1}(\omega_b) \\
\Leftrightarrow & & \min(\{\kappa(\omega) \mid \omega \in \mathit{dom}(\kappa),\ \omega^1 = \omega_a\}, \leqslant) & \\
& \leqslant & \min(\{\kappa(\omega) \mid \omega \in \mathit{dom}(\kappa),\ \omega^1 = \omega_b\}, \leqslant) & \\
\Leftrightarrow & & \min(\{\omega \mid \omega \in \mathit{Mod}_\Sigma(A),\ \omega^1 = \omega_a\}, \preceq_2) & \\
& \preceq_2 & \min(\{\omega \mid \omega \in \mathit{Mod}_\Sigma(A),\ \omega^1 = \omega_b\}, \preceq_2) & \\
\Leftrightarrow & & \omega_a \ &\preceq_{2|\Sigma_1} \ \omega_b
\end{aligned}$$

□

Proposition 7. *τ^* fulfils (Cond).*

Proof. Let $\kappa \in \mathcal{M}_{OCF}(\Sigma, A)$ be a ranking function and $F \in \mathcal{L}_\Sigma$ such that $\mathit{Mod}_\Sigma(A) \cap \mathit{Mod}_\Sigma(F) \neq \emptyset$. Let $\preceq_1 = \tau^*(\kappa|F)$ and $\preceq_2 = \tau^*(\kappa)$. Let $\omega_1, \omega_2 \in \mathit{Mod}(A \wedge F)$.

$$\begin{aligned}
& & \omega_1 \ &\preceq_1 \ \omega_2 \\
\Leftrightarrow & & \kappa_{|F}(\omega_1) \ &\leqslant \ \kappa_{|F}(\omega_2) \\
\Leftrightarrow & & \kappa(\omega_1) \ &\leqslant \ \kappa(\omega_2) \\
\Leftrightarrow & & \omega_1 \ &\preceq_2 \ \omega_2 \\
\Leftrightarrow & & \omega_1 \ &\preceq_2|F \ \omega_2
\end{aligned}$$

□

8 Mapping Total Preorders to Ranking Functions

Now, we want to consider epistemic state mappings that map a total preorder to a ranking function.

Since the functions in τ^* are not bijective, we cannot simply reverse them. On the contrary, there is more than one epistemic state mapping $\rho : \mathit{TPO} \rightsquigarrow \mathit{OCF}$ that fulfils (Ord). That is not surprising as a ranking function contains more information than a total preorder over the same domain. The additional information is the absolute distance between worlds. The functions in ρ need to fill in this missing information.

Example 4. *Let $\rho : \mathit{TPO} \rightsquigarrow \mathit{OCF}$ be an epistemic state mapping defined as follows. For $\preceq \in \mathcal{M}_{TPO}(\Sigma, A)$ let $L_0^{\preceq} = \min(\mathit{dom}(\preceq), \preceq)$ and $L_k^{\preceq} = \min(\mathit{dom}(\preceq) \setminus (L_0^{\preceq} \cup \dots \cup L_{k-1}^{\preceq}), \preceq)$. Every set $L_k^{\preceq}$ corresponds to the k-th layer of $\preceq$. The sets $L_i^{\preceq}$ and $L_j^{\preceq}$ are disjunct for $i \neq j$. We define $\xi(\preceq) = \kappa$ with $\kappa(\omega) = k$ such that $\omega \in L_k^{\preceq}$ for every $\omega \in \mathit{dom}(\preceq)$.*

For example, the TPO $ab \prec \overline{a}b, a\overline{b} \prec \overline{a}\overline{b}$ over $\Sigma = \{a, b\}$ is mapped to $\kappa : \{ab \mapsto 0, \overline{a}b \mapsto 1, a\overline{b} \mapsto 1, \overline{a}\overline{b} \mapsto 2\}$ by ρ.

The epistemic state mapping ρ fulfils (Ord).

To limit the possible outcomes of the transformation, we consider additional postulates such as (SynSplit). Unfortunately, there is no epistemic state mapping ρ that fulfils both (Ord) and (SynSplit).

Proposition 8. *There is no epistemic state mapping $\rho : \mathit{TPO} \rightsquigarrow \mathit{OCF}$ that fulfils (Ord) and (SynSplit).*

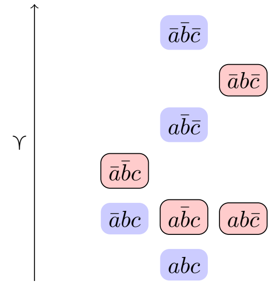


Figure 3: Total preorder $\preceq$ on $\Sigma = \{a, b, c\}$ with syntax splitting $\{a\} \,\dot{\cup}\, \{b\} \,\dot{\cup}\, \{c\}$. There is no ranking function with that syntax splitting that induces $\preceq$.

Proof. Let $\Sigma = \{a, b, c\}$ be a signature and $\preceq$ be the total preorder over Σ displayed in Figure 3. This TPO has the syntax splitting $\{a\}\dot{\cup}\{b\}\dot{\cup}\{c\}$. Considering the highlighted (red and circled) worlds in Figure 3, we can see that there is no ranking function κ such that both (Ord) holds and κ has the syntax splitting $\{a\} \,\dot{\cup}\, \{b\} \,\dot{\cup}\, \{c\}$. The worlds $a\overline{b}c$ and $ab\overline{c}$ must have the same rank due to (Ord) and the world $\overline{a}\overline{b}c$ must have a lower rank than $\overline{a}b\overline{c}$. A ranking function with these properties cannot have the considered syntax splitting: A ranking function κ with the syntax splitting $\{a\}\dot{\cup}\{b\}\dot{\cup}\{c\}$ would also have the syntax splitting $\{a\} \,\dot{\cup}\, \{b, c\}$ and therefore, we would have $\kappa(a\overline{b}c) - \kappa(ab\overline{c}) = \kappa(\overline{a}\overline{b}c) - \kappa(\overline{a}b\overline{c})$. □

This incompatibility persists if we consider the weaker (SynSplit$^\text{b}$) instead of (SynSplit) and (wIE$^\Rightarrow$) instead of (IE).

Proposition 9. *There is no epistemic state mapping $\rho : \mathit{TPO} \rightsquigarrow \mathit{OCF}$ that fulfils both (wIE$^\Rightarrow$) and (SynSplit$^\text{b}$).*

Proof. Let $\Sigma = \{a, b, c, d\}$ be a signature and $\preceq$ be the total preorder over Σ displayed in Figure 4. This TPO has the syntax splitting $\{a, b\} \,\dot{\cup}\, \{c, d\}$. Assume there is a ranking function κ with syntax splitting $\{a, b\}\dot{\cup}\{b, c\}$ such that $\omega_1 \prec \omega_2$ implies $\kappa(\omega_1) < \kappa(\omega_2)$. Then there are ranking functions $\kappa_1 : \Omega_{\{a,b\}} \to \mathbb{N}_0$ and $\kappa_2 : \Omega_{\{c,d\}} \to \mathbb{N}_0$ such that $\kappa = \kappa_1 \oplus \kappa_2$. Let

$$\begin{array}{llll}
\kappa_1(ab) = 0 & \kappa_1(a\overline{b}) = i & \kappa_1(\overline{a}b) = j & \kappa_1(\overline{a}\overline{b}) = k \\
\kappa_2(cd) = 0 & \kappa_2(c\overline{d}) = l & \kappa_2(\overline{c}d) = m & \kappa_2(\overline{c}\overline{d}) = n.
\end{array}$$

As $\omega_1 \prec \omega_2$ implies $\kappa(\omega_1) < \kappa(\omega_2)$ for every $\omega_1, \omega_2 \in \Omega_\Sigma$ we have that

$$\begin{aligned}
m + j = \kappa_1(\overline{a}b) + \kappa_2(\overline{c}d) = \kappa(\overline{a}b\overline{c}d) &< \kappa(a\overline{b}\overline{c}\overline{d}) \\
&= \kappa_1(a\overline{b}) + \kappa_2(\overline{c}\overline{d}) = i + n.
\end{aligned}$$

Analogously, we get $j > n$ from $\kappa(\overline{a}bcd) > \kappa(ab\overline{c}\overline{d})$ and $m > i$ from $\kappa(ab\overline{c}d) > \kappa(a\overline{b}cd)$. The combination of these inequations is a contradiction. The assumed ranking function κ cannot exist. □

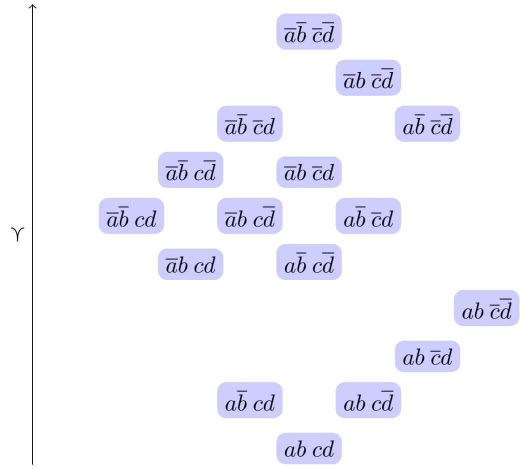

⋎
$\overline{a}\overline{b}\,\overline{c}\overline{d}$
$\overline{a}b\,\overline{c}\overline{d}$
$\overline{a}\overline{b}\,\overline{c}d$
$a\overline{b}\,\overline{c}\overline{d}$
$\overline{a}\overline{b}\,c\overline{d}$
$\overline{a}b\,\overline{c}d$
$\overline{a}\overline{b}\,cd$
$\overline{a}b\,c\overline{d}$
$a\overline{b}\,\overline{c}d$
$\overline{a}b\,cd$
$a\overline{b}\,c\overline{d}$
$ab\,\overline{c}\overline{d}$
$ab\,\overline{c}d$
$a\overline{b}\,cd$
$ab\,c\overline{d}$
$ab\,cd$

Figure 4: Total preorder $\preceq$ on $\Sigma = \{a, b, c, d\}$ with syntax splitting $\{a, b\} \dot\cup \{c, d\}$. There is no ranking function with that syntax splitting that induces a superset of $\preceq$.

However, the combination of (wIE$^{\Leftarrow}$) and (SynSplit) is consistent, as we will see later (in Proposition 14).

Any epistemic state mapping ρ from total preorders to ranking functions satisfying (Ord) is compatible with τ^* (see Proposition 4) with respect to marginalization and conditionalization.

Proposition 10. *Let $\rho : TPO \rightsquigarrow OCF$ be an epistemic state mapping that fulfils (Ord). For every total preorder $\preceq \in \mathcal{M}_{TPO}(\Sigma, A)$ and $\Sigma' \subseteq \Sigma$ it holds that*

$$\tau^*(\rho(\preceq)_{|\Sigma'}) = \preceq_{|\Sigma'}.$$

Proof. Let ρ satisfy (Ord). Let $\preceq$ be a TPO over Σ and $\kappa = \rho(\preceq)$. Let $\Sigma' \subseteq \Sigma$ and $\preceq' = \tau^*(\kappa_{|\Sigma'})$. Let $\omega_1, \omega_2 \in dom(\preceq')$.

$$\begin{aligned}
& \omega_1 \preceq' \omega_2 \\
\Leftrightarrow \quad & \kappa_{|\Sigma'}(\omega_1) \leqslant \kappa_{|\Sigma'}(\omega_2) \\
\Leftrightarrow \quad & \min(\{\kappa(\omega) \mid \omega \in \Omega_\Sigma, \omega_1 \models \omega\}, \leqslant) \\
& \leqslant \min(\{\kappa(\omega) \mid \omega \in \Omega_\Sigma, \omega_2 \models \omega\}, \leqslant) \\
\Leftrightarrow \quad & \min(\{\omega \mid \omega \in \Omega_\Sigma, \omega_1 \models \omega\}, \preceq) \\
& \preceq \min(\{\omega \mid \omega \in \Omega_\Sigma, \omega_2 \models \omega\}, \preceq) \\
\Leftrightarrow \quad & \omega_1 \preceq_{|\Sigma'} \omega_2
\end{aligned}$$

□

Proposition 11. *Let $\rho : TPO \rightsquigarrow OCF$ be an epistemic state mapping fulfilling (Ord). For every total preorder $\preceq \in \mathcal{M}_{TPO}(\Sigma, A)$ and $F \in \mathcal{L}_\Sigma$ it holds that*

$$\tau^*(\rho(\preceq)|F) = \preceq|F.$$

Proof. Let ρ satisfy (Ord). Let $\preceq$ be a TPO over $Mod_\Sigma(A)$ and $\kappa = \rho(\preceq)$. Let $F \in \mathcal{L}_\Sigma$ and $\preceq' = \tau^*(\kappa|F)$. Let $\omega_1, \omega_2 \in Mod_\Sigma(A \wedge F)$.

$$\begin{aligned}
& \omega_1 \preceq' \omega_2 \\
\Leftrightarrow \quad & \kappa|F(\omega_1) \leqslant \kappa|F(\omega_2) \\
\Leftrightarrow \quad & \kappa(\omega_1) \leqslant \omega_2 \\
\Leftrightarrow \quad & \omega_1 \preceq \omega_2 \\
\Leftrightarrow \quad & \omega_1 \preceq|F\, \omega_2
\end{aligned}$$

□

$$\begin{tikzcd}
\mathcal{M}_{OCF}(\Sigma, A) \arrow[r, "\cdot_{|\Sigma'}"] & \mathcal{M}_{OCF}(\Sigma', A) \arrow[d, "\tau^*"] \\
\mathcal{M}_{TPO}(\Sigma, A) \arrow[u, "\rho"] \arrow[r, "\cdot_{|\Sigma'}"'] & \mathcal{M}_{TPO}(\Sigma', A)
\end{tikzcd}$$

(a) A commutating diagram illustrating Proposition 10.

$$\begin{tikzcd}
\mathcal{M}_{OCF}(\Sigma, A) \arrow[r, "\cdot|F"] & \mathcal{M}_{OCF}(\Sigma, A \wedge F) \arrow[d, "\tau^*"] \\
\mathcal{M}_{TPO}(\Sigma, A) \arrow[u, "\rho"] \arrow[r, "\cdot|F"'] & \mathcal{M}_{TPO}(\Sigma, A \wedge F)
\end{tikzcd}$$

(b) A commutating diagram illustrating Proposition 11.

Figure 5: Illustration of Propositions 10 and 11. Precondition of both propositions is that $\rho : TPO \rightsquigarrow OCF$ fulfils (Ord).

It would be useful, if a transformation from a total preorder to a ranking function preserved marginalization and conditionalization in the way τ^* does for the other direction.

But Postulate (Cond) is unfulfillable in combination with (Ord). (Cond) is even incompatible with the weaker Postulate (wIE$^{\Rightarrow}$).

Proposition 12. *There is no epistemic state mapping $\rho : TPO \rightsquigarrow OCF$ that fulfils (Cond) and (wIE$^{\Rightarrow}$).*

Proof. Let $\Sigma = \{a, b\}$ be a signature and $\preceq_1, \preceq_2$ be the total preorders over Σ displayed in Figure 6. We have $\preceq_1|a = \preceq_2|a$ and $\preceq_1|b = \preceq_2|b$. Let $\kappa_1 = \rho(\preceq_1)$ and $\kappa_2 = \rho(\preceq_2)$. If (wIE$^{\Rightarrow}$) and (Cond) were true it would imply $\kappa_1(\overline{a}b) = \kappa_1|b(\overline{a}b) = \kappa_2|b(\overline{a}b) = \kappa_2(\overline{a}b)$ and $\kappa_1(a\overline{b}) = \kappa_1|a(a\overline{b}) = \kappa_2|a(a\overline{b}) = \kappa_2(a\overline{b})$. This contradicts (wIE$^{\Rightarrow}$) as (wIE$^{\Rightarrow}$) requires $\kappa_1(\overline{a}b) > \kappa_1(a\overline{b})$ and $\kappa_2(\overline{a}b) < \kappa_2(a\overline{b})$. □

Also, postulate (Marg) is unfulfillable in combination with (Ord) or (wIE$^{\Rightarrow}$) in general.

Proposition 13. *There is no epistemic state mapping $\rho : TPO \rightsquigarrow OCF$ that fulfils (wIE$^{\Rightarrow}$) and (Marg).*

Proof. Let $\Sigma = \{a, b\}$ be a signature and $\preceq_1, \preceq_2$ be the total preorders over Σ displayed in Figure 7. Let $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$. We have $\preceq_{1|\Sigma_1} = \preceq_{2|\Sigma_1}$ and $\preceq_{1|\Sigma_2} = \preceq_{2|\Sigma_2}$. Let $\kappa_1 = \rho(\preceq_1)$ and $\kappa_2 = \rho(\preceq_2)$. If (wIE$^{\Rightarrow}$) and (Marg) were true it would imply $\kappa_1(\overline{a}b) = \kappa_{1|\Sigma_1}(\overline{a}b) = \kappa_{2|\Sigma_1}(\overline{a}b) =$

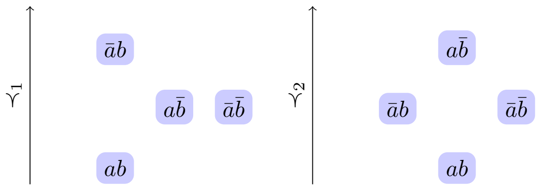


Figure 6: Total preorders $\preceq_1$ and $\preceq_2$ on $\Sigma = \{a, b\}$ which show that (Cond) is incompatible with (wIE$^{\Rightarrow}$) for epistemic state mappings from TPOs to OCFs.

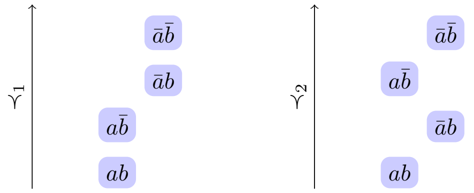


Figure 7: Total preorders $\preceq_1$ and $\preceq_2$ on $\Sigma = \{a, b\}$ which show that (Marg) is incompatible with (wIE$^{\Rightarrow}$) for epistemic state mappings from TPOs to OCFs.

$\kappa_2(\overline{a}b)$ and $\kappa_1(a\overline{b}) = \kappa_{1|\Sigma_2}(a\overline{b}) = \kappa_{2|\Sigma_2}(a\overline{b}) = \kappa_2(a\overline{b})$. This contradicts (wIE$^{\Rightarrow}$) as (wIE$^{\Rightarrow}$) requires $\kappa_1(\overline{a}b) > \kappa_1(a\overline{b})$ and $\kappa_2(\overline{a}b) < \kappa_2(a\overline{b})$. □

The Propositions 9, 12, and 13 all showed that (wIE$^{\Rightarrow}$) in combination with some of the other postulates cannot be fulfilled. (wIE$^{\Leftarrow}$) on the other hand can be fulfilled in combination with these other postulates.

Proposition 14. *The combination of (wIE$^{\Leftarrow}$), (Cond), and (Marg) is consistent.*

Proof. The epistemic state mapping that maps every TPO to the uniform ranking function κ_{uni} over the respective domain fulfils all three postulates. □

However, the following triviality result shows that there is only one epistemic state mapping fulfilling the combination of (wIE$^{\Leftarrow}$) and (Cond).

Proposition 15. *The only epistemic state mapping ρ : TPO $\rightsquigarrow$ OCF that fulfils (wIE$^{\Leftarrow}$) and (Cond) maps every TPO to the trivial uniform ranking function κ_{uni}.*

Proof. Let ρ be an epistemic state mapping fulfilling (wIE$^{\Leftarrow}$) and (Marg). Let Σ be a signature and $\omega_1, \omega_2 \in \Omega_\Sigma$ with $\omega_1 \neq \omega_2$. Choose a third world $\omega_3 \in \Omega_\Sigma$ with $\omega_3 \notin \{\omega_1, \omega_2\}$ and consider the TPOs

$$\omega_3 \prec_1 \omega_2 \prec_1 \omega_1 \prec_1 \omega_4, \ldots, \omega_n$$
$$\omega_3 \prec_2 \omega_1 \prec_2 \omega_2 \prec_2 \omega_4, \ldots, \omega_n$$

with $\{\omega_4, \ldots, \omega_n\} = \Omega_\Sigma \setminus \{\omega_1, \omega_2, \omega_3\}$. Let $\kappa_1 = \rho(\preceq_1)$ and $\kappa_2 = \rho(\preceq_2)$. The postulate (wIE$^{\Rightarrow}$) requires that

$$\kappa_1(\omega_2) \leqslant \kappa_1(\omega_1) \quad \text{and} \quad \kappa_2(\omega_1) \leqslant \kappa_2(\omega_2). \qquad (*)$$

Let $A = \omega_3 \vee \omega_1$ and $B = \omega_3 \vee \omega_2$. Conditionalization yields $\preceq'_A = \preceq_1|A = \preceq_2|A$ and $\preceq'_B = \preceq_1|B = \preceq_2|B$. Postulate (Cond) requires $\kappa_1|A = \rho(\preceq'_A) = \kappa_2|A$ and $\kappa_1|B = \rho(\preceq'_B) = \kappa_2|B$. This implies $\kappa_1(\omega_1) = \kappa_2(\omega_1)$ and $\kappa_1(\omega_2) = \kappa_2(\omega_2)$. With $(*)$ it follows that $\kappa_1(\omega_2) \leqslant \kappa_1(\omega_1) = \kappa_2(\omega_1) \leqslant \kappa_2(\omega_2) = \kappa_2(\omega_2)$. Therefore we can replace the $\leqslant$ in this chain of (in-)equations by $=$. Let $C = \omega_1 \vee \omega_2$. We can see that both $\preceq_1|C = \{\omega_1 \prec \omega_2\}$ and $\preceq_2|C = \{\omega_2 \prec \omega_1\}$ are mapped to the uniform ranking function κ_{uni} due to (Cond).

Since we can choose any two worlds as ω_1, ω_2 in this argumentation, (Cond) requires that any TPO is mapped to the uniform ranking function κ_{uni}. □

Note that this results only apply to epistemic state mappings defined for all TPOs. Mappings that are defined over a certain subset of TPOs might still fulfil combinations of postulates.

9 Conclusions and Future Work

In this paper, we introduced the notion of epistemic state mappings, i.e., mappings within and across the frameworks of OCFs and TPOs. We proposed postulates for epistemic state mappings that ensure the preservation of certain properties of the epistemic state across the mapping. The properties considered in the paper include the set of entailed conditionals and syntax splitting. Other postulates ensure compatibility with the operations marginalization and conditionalization, respectively. Furthermore, we investigated the relationships among the proposed postulates in general and for each combination of the considered framework. Some postulates are entailed by other postulates, e.g., (SynSplit) entails (SynSplit$^{\mathrm{b}}$), (IE) is equivalent to (Ord). We also showed that there are constellations and combinations of the postulates which cannot be satisfied simultaneously, e.g., there is no epistemic state mapping from TPOs to OCFs that fulfils both (wIE$^{\Rightarrow}$) and (SynSplit$^{\mathrm{b}}$). The only epistemic state mapping from TPOs to OCFs that fulfils both (wIE$^{\Rightarrow}$) and (SynSplit$^{\mathrm{b}}$) is the trivial mapping of every TPO to κ_{uni} representing the state of complete ignorance.

Our current work includes extending the investigation of epistemic state mappings and their properties for establishing further relationships between OCFs and TPOs and thus to transfer more results between the two frameworks. Furthermore, we will consider epistemic state mappings among particular subclasses of TPOs and OCFs. We expect to find interesting and relevant subclasses such that epistemic state mappings over these subclasses fulfil combinations of postulates that are not fulfilled by epistemic state mappings over the full sets of TPOs and OCFs.

References

Adams, E. 1975. *The Logic of Conditionals*. Dordrecht: D. Reidel.

Beierle, C., and Kern-Isberner, G. 2012. Semantical investigations into nonmonotonic and probabilistic logics. *Annals of Mathematics and Artificial Intelligence* 65(2-3):123–158.

Beierle, C.; Kern-Isberner, G.; Sauerwald, K.; Bock, T.; and Ragni, M. 2019. Towards a general framework for kinds of forgetting in common-sense belief management. *KI – Künstliche Intelligenz* 33(1):57–68.

Benferhat, S.; Dubois, D.; and Prade, H. 1999. Possibilistic and standard probabilistic semantics of conditional knowledge bases. *J. of Logic and Computation* 9(6):873–895.

de Finetti, B. 1937. La prévision, ses lois logiques et ses sources subjectives. *Ann. Inst. H. Poincaré* 7(1):1–68. Engl. transl. *Theory of Probability*, J. Wiley & Sons, 1974.

Delgrande, J. P. 2017. A knowledge level account of forgetting. *J. Artif. Intell. Res.* 60:1165–1213.

Dubois, D., and Prade, H. 1994. Conditional objects as nonmonotonic consequence relationships. *IEEE Transactions on Systems, Man, and Cybernetics* 24(12):1724–1740.

Eiter, T., and Kern-Isberner, G. 2019. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI – Künstliche Intelligenz* 33(1):9–33.

Goldszmidt, M., and Pearl, J. 1996. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence* 84:57–112.

Haldimann, J.; Beierle, C.; and Kern-Isberner, G. 2021. Syntax splitting for iterated contractions, ignorations, and revisions on ranking functions using selection strategies. In Faber, W.; Friedrich, G.; Gebser, M.; and Morak, M., eds., *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings*, volume 12678 of *Lecture Notes in Computer Science*, 85–100. Springer.

Haldimann, J. P.; Kern-Isberner, G.; and Beierle, C. 2020. Syntax splitting for iterated contractions. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, 465–475.

Katsuno, H., and Mendelzon, A. O. 1992. Propositional knowledge base revision and minimal change. *Artif. Intell.* 52(3):263–294.

Kern-Isberner, G., and Brewka, G. 2017. Strong syntax splitting for iterated belief revision. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1131–1137.

Kern-Isberner, G.; Beierle, C.; and Brewka, G. 2020. Syntax splitting = relevance + independence: New postulates for nonmonotonic reasoning from conditional belief bases. In *KR-2020*, 560–571.

Kern-Isberner, G. 2004. A thorough axiomatization of a principle of conditional preservation in belief revision. *Annals of Mathematics and Artificial Intelligence* 40(1-2):127–164.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44:167–207.

Lewis, D. 1973. *Counterfactuals*. Cambridge, Mass.: Harvard University Press.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, Language, and Computation* 2:266–278.

Pearl, J. 1990. Bayesian and belief-functions formalisms for evidential reasoning: a conceptual analysis. In Ras, Z., and Zemankova, M., eds., *Intelligent Systems – State of the art and future directions*. Chichester: Ellis Horwood. 73–117.

Peppas, P.; Williams, M.-A.; Chopra, S.; and Foo, N. Y. 2015. Relevance in belief revision. *Artificial Intelligence* 229((1-2)):126–138.

Sezgin, M., and Kern-Isberner, G. 2020. Generalized ranking kinematics for iterated belief revision. In *Proceedings of the Thirty-Third International FLAIRS Conference, FLAIRS-33*. AAAI Press.

Spohn, W. 1988. Ordinal conditional functions: a dynamic theory of epistemic states. In Harper, W., and Skyrms, B., eds., *Causation in Decision, Belief Change, and Statistics, II*. Kluwer Academic Publishers. 105–134.

Modeling Human Reasoning About Conditionals

Marcos Cramer[1] , **Steffen Hölldobler**[1,2] , **Marco Ragni**[3*]
[1]Technische Universität Dresden, Dresden, Germany
[2]North Caucasus Federal University, Stavropol, Russian Federation
[3]Danish Institute of Advanced Studies, South Denmark University
marcos.cramer@tu-dresden.de, sh43@posteo.de, ragni@sdu.dk

Abstract

Numerous results in psychology demonstrate that inferences humans draw from conditional sentences (i.e., sentences of the form "if *antecedent* then *consequent*") differ systematically from classical two-valued logical inferences. Today, still no formal approach yet exists which captures the specifics of semantic differences between types of conditional sentences (obligation vs. factual) or types of antecedents (necessary vs. non-necessary). We claim that the three-valued, non-monotonic weak completion semantics can model human conditional reasoning adequately, especially with this distinction. We test the predictions of the weak completion semantics in a psychological experiment and demonstrate its cognitive adequacy. We situate the results within formal and cognitive theories and argue that we need logics that are descriptive for the human inference process.

1 Introduction

To demonstrate some specifics of human reasoning, we consider four examples: What follows in each of the following reasoning problems?

1. *If it rains, then the roofs must be wet* and *it rains* (AA).
2. *If Paul rides a motorbike, then Paul must wear a helmet* and *Paul does not ride a motorbike* (DA).
3. *If the library is open, then Elisa is studying late in the library* and *Elisa is studying late in the library* (AC).
4. *If Nancy rides her motorbike, then Nancy goes to the mountains* and *Nancy does not go to the mountains* (DC).

In each example, a conditional sentence is given together with a positive or negative atomic sentence, which is the affirmation of the antecedent (AA), the denial of the antecedent (DA), the affirmation of the consequent (AC), or the denial of the consequent (DC). The examples are adapted from the literature (Dietz Saldanha, Hölldobler, and Lourêdo Rocha 2017; Byrne 2005; Byrne 1989).

We claim that most humans answer *the roofs are wet*, *Paul does not wear a helmet*, *the library is open*, and *Nancy does not ride her motorbike*, respectively, if they have not been exposed to logic before. For the Examples 1 and 4 the answers can be obtained by applying modus ponens and modus tollens, respectively; two valid inference rules in classical two-valued logic. However, for Examples 2 and 3 the answers are invalid in classical two-valued logic.

Such a logic does not seem to be of great help when modeling human conditional reasoning as long as conditional sentences are represented by implications. Moreover, as Byrne has shown in (Byrne 1989) for each of the four types of inference, humans may suppress previously drawn conclusions when additional knowledge becomes available; this holds for valid as well as invalid inferences with respect to classical two-valued logic. Hence, this calls for a theory based on non-monotonic logic. The well established *mental model theory* (Johnson-Laird and Byrne 1991; Khemlani, Byrne, and Johnson-Laird 2018) claims that conditional sentences trigger the representation of sets of possibilities. The respective possibilities can be modulated by a reasoner's knowledge, or pragmatics, or semantics leading to different representations (Johnson-Laird and Byrne 2002). Barrouillet et al. demonstrated in (Barrouillet, Grosset, and Lecas 2000) that there is an implicit order on these possibilities in conditional reasoning. A default representation (not considering these modulations above) correctly predicts the answers in the cases AA and DC, but in the cases DA and AC it predicts that humans will answer *nothing follows*. It is well-known that humans sometimes consider conditional sentences as bi-conditionals (see e.g. (Johnson-Laird and Byrne 1991)), but it is surprising that this seems to hold for all four examples if our earlier claim is correct.

Returning to human conditional reasoning, the main question tackled in this paper is *how can human conditional reasoning be adequately modeled*? Following Bibel (Bibel 1991) we believe that *there is an adequate general proof method that can automatically discover any proof done by humans provided the problem (including all required knowledge) is stated in appropriately formalized terms* where adequateness, roughly speaking, is understood as the property of a theorem proving method that *for any given knowledge base, the method solves simpler problems faster than more difficult ones*.

In this paper we will show that the *weak completion semantics (WCS)*, a three-valued, and non-monotonic cognitive theory, can adequately model human conditional reasoning. In particular, it can adequately model the four examples discussed above. Moreover, it can also explain the differences humans seem to make in the cases AC and DC when

[*] Authors are given in alphabetical order.

dealing with conditional sentences classified as obligation or factual and antecedents classified as necessary or non-necessary. In the case of AC, humans answer with *nothing follows* significantly more often when given a non-necessary antecedent. In the case of DC, on the other hand, they answer with *nothing follows* much more often when given a factual conditional.

In order to validate the claims made above as well as the predictions made by the WCS we designed and performed an experiment involving 56 logically naive participants from Central Europe and Great Britain. The results confirm the claims made above as well as (most of) the predictions made by the WCS. But the results also point towards open research questions.

The paper is organized as follows. After presenting the WCS in Section 2, we introduce a classification of conditional sentences in Section 3. Taking this classification into account, we extend the WCS. As shown in Section 4, this will lead to a number of predictions made by the WCS. These predictions including the claims made at the beginning of this paper are tested in an experiment specified in Section 5. The experiment will be evaluated in Section 6. A discussion and an outlook to future work concludes the paper in Section 7.

2 The Weak Completion Semantics

We assume the reader to be familiar with logic and logic programming as presented in e.g. (Fitting 1996) and (Lloyd 1984). Let $\top$, $\bot$, and $\cup$ be truth constants denoting *true*, *false*, and *unknown*, respectively. A *(logic) program* is a finite set of clauses of the form $B \leftarrow \mathit{body}$, where B is an atom and *body* is either $\top$, or $\bot$, or a finite, non-empty set of literals. Clauses of the form $B \leftarrow \top$, $B \leftarrow \bot$, and $B \leftarrow L_1, \ldots, L_n$ are called *facts*, *assumptions*, and *rules*, respectively, where L_i, $1 \leq i \leq n$, are literals. We restrict our attention to propositional programs although the WCS extends to first-order programs as well (Hölldobler 2015).

Throughout this paper, $\mathcal{P}$ will denote a program. An atom B is *defined* in $\mathcal{P}$ iff $\mathcal{P}$ contains a clause of the form $B \leftarrow \mathit{body}$. As an example consider the program

$$\mathcal{P}_c = \{C \leftarrow A \wedge \neg ab,\ ab \leftarrow \bot\},$$

where A, C, and ab are atoms. C and ab are defined, whereas A is undefined. ab is an abnormality predicate which is assumed to be false. In the WCS, this program represents the conditional sentence *if A then C*.

Consider the following transformation: (1) For all defined atoms B occurring in $\mathcal{P}$, replace all clauses of the form $B \leftarrow \mathit{body}_1$, $B \leftarrow \mathit{body}_2$, ... by $B \leftarrow \mathit{body}_1 \vee \mathit{body}_2 \vee \ldots$. (2) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$. The resulting set of equivalences is called the *weak completion* of $\mathcal{P}$. It differs from the completion defined in (Clark 1978) in that undefined atoms are not mapped to false, but to unknown instead.

As shown in (Hölldobler and Kencana Ramli 2009a), each weakly completed program admits a least model under the three-valued Łukasiewicz logic (Łukasiewicz 1920) (see Table 1). This model will be denoted by $\mathcal{M}_\mathcal{P}$. It can be computed as the least fixed point of a semantic operator introduced in (Stenning and van Lambalgen 2008). Let $\mathcal{P}$ be a program and I a three-valued interpretation represented by the pair $\langle I^\top, I^\bot \rangle$, where $I^\top$ and $I^\bot$ are the sets of atoms mapped to true and false by I, respectively, and atoms which are not listed are mapped to unknown by I. We define $\Phi_\mathcal{P}\, I = \langle J^\top, J^\bot \rangle$,[1] where

$$\begin{aligned} J^\top &= \{B \mid \text{there is } B \leftarrow \mathit{body} \in \mathcal{P} \text{ and } I\,\mathit{body} = \top\}, \\ J^\bot &= \{B \mid \text{there is } B \leftarrow \mathit{body} \in \mathcal{P} \text{ and} \\ &\qquad \text{for all } B \leftarrow \mathit{body} \in \mathcal{P} \text{ we find } I\,\mathit{body} = \bot\}. \end{aligned}$$

Following (Kakas, Kowalski, and Toni 1992) we consider an *abductive framework* $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{IC}, \models_{wcs} \rangle$, where $\mathcal{P}$ is a logic program, $\mathcal{A}_\mathcal{P} = \{B \leftarrow \top \mid B \text{ is undefined in } \mathcal{P}\} \cup \{B \leftarrow \bot \mid B \text{ is undefined in } \mathcal{P}\}$ is the *set of abducibles*, $\mathcal{IC}$ is a finite set of *integrity constraints*,[2] and $\mathcal{M}_\mathcal{P} \models_{wcs} F$ iff $\mathcal{M}_\mathcal{P}$ maps the formula F to true. Let $\mathcal{O}$ be an *observation*, i.e., a finite set of literals. $\mathcal{O}$ is *explainable* in the abductive framework $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{IC}, \models_{wcs} \rangle$ iff there exists a non-empty $\mathcal{X} \subseteq \mathcal{A}_\mathcal{P}$ called an *explanation* such that $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}} \models_{wcs} L$ for all $L \in \mathcal{O}$ and $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}}$ satisfies $\mathcal{IC}$. Formula F *follows credulously* from $\mathcal{P}$ and $\mathcal{O}$ iff there exists an explanation $\mathcal{X}$ for $\mathcal{O}$ such that $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}} \models_{wcs} F$. F *follows skeptically* from $\mathcal{P}$ and $\mathcal{O}$ iff $\mathcal{O}$ can be explained and for all explanations $\mathcal{X}$ for $\mathcal{O}$ we find $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}} \models_{wcs} F$. One should observe that if an observation $\mathcal{O}$ cannot be explained, then *nothing follows* credulously as well as skeptically. In case of skeptical consequences this is an application of the so-called *Gricean implicature* (Grice 1975): humans normally do not quantify over things which do not exist.

Given premises, general knowledge, and observations, *reasoning in the WCS* is hence modeled in five steps:

1. Reasoning towards a program $\mathcal{P}$ following (Stenning and van Lambalgen 2008).
2. Weakly completing the program.
3. Computing the least model $\mathcal{M}_\mathcal{P}$ of the weak completion of $\mathcal{P}$ under the three-valued Łukasiewicz logic.
4. Reasoning with respect to $\mathcal{M}_\mathcal{P}$.
5. If observations cannot be explained, then applying skeptical abduction.

In Section 4 we will explain how these five steps work in the case of the conditional reasoning tasks considered in this paper. More examples can be found, for example, in (Dietz, Hölldobler, and Ragni 2012) or (Oliviera da Costa et al. 2017).

3 A Classification of Conditional Sentences

Obligation versus Factual Conditionals A conditional sentence whose consequent appears to be obligatory given the antecedent is called an *obligation conditional*. As pointed out by Byrne (Byrne 2005), for each obligation conditional there are two initial possibilities people think about. The first possibility is the conjunction of the antecedent and the consequent; it is permitted. The second possibility is the

[1] Whenever we apply a unary operator like $\Phi_\mathcal{P}$ to an argument like I, then we omit parenthesis and write $\Phi_\mathcal{P}\, I$ instead. Likewise, we write $I\,\mathit{body}$ instead of $I(\mathit{body})$.

[2] In all examples discussed in this paper $\mathcal{IC} = \emptyset$.

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

Table 1: The truth tables for the Łukasiewicz logic. One should observe that U $\leftarrow$ U $=$ U $\leftrightarrow$ U $= \top$ as shown in the grey cells.

conjunction of the antecedent and the negation of the consequent; it is forbidden. Reconsidering Example 1, the permitted possibility is *it rains* and *the roofs are wet*, whereas the forbidden possibility is *it rains* and *the roofs are not wet*. Obligations are deontic obligations, i.e. legal, moral, or societal obligations of a person to perform certain actions, or naive physical obligations that cannot be avoided under normal circumstances. The fact that the consequence is obligatory may be explicitly marked with a word like *must*, but this is unnecessary. The exemplary conditional sentences 1 and 2 presented in the Introduction appear to be obligations.

If the consequent of a conditional is not obligatory, then it is called a *factual conditional*. In particulr, there is no forbidden possibility in such a case. This appears to hold for Examples 3 and 4 given in the introduction.

Necessary versus Non-Necessary Antecedents The antecedent A of a conditional sentence *if* A *then* C is said to be *necessary* if and only if its consequent C cannot be true unless A is true. More precisely, A may be true while C is not, but C cannot be true while A is not. For example, the *library being open* is a necessary antecedent for *studying late in the library*, but visitors of a library can have varying reasons like *reading textbooks* or *having an essay to write* for *studying late in the library*. In the examples presented in the introduction, it appears that the antecedents of Examples 1 and 3 are necessary, whereas the antecedents of Examples 2 and 4 appear to be non-necessary.

Pragmatics Humans may classify conditional sentences as obligation or factual and antecedents as necessary or non-necessary. This is an informal and pragmatic classification. It depends on the background knowledge and experience of a person as well as on the context. For example, the conditional sentence *if it is cloudy, then it is raining* discussed in (Khemlani, Byrne, and Johnson-Laird 2018) may be classified as an obligation conditional with necessary antecedent by people living in Java, whereas it may be classified as a factual conditional by people living in Central Europe.

WCS The classification of conditional sentences can be taken into account by extending the definition of the set of abducibles:

$$\mathcal{A}^e_{\mathcal{P}} = \mathcal{A}_{\mathcal{P}} \cup \mathcal{A}^{nn}_{\mathcal{P}} \cup \mathcal{A}^f_{\mathcal{P}},$$

where $\mathcal{A}_{\mathcal{P}}$ is as defined above,

$$\begin{array}{rcll} \mathcal{A}^{nn}_{\mathcal{P}} & = & \{C \leftarrow \top \mid & C \text{ is the head of a rule occurring in } \mathcal{P} \\ & & & \text{representing a conditional with} \\ & & & \text{non-necessary antecedent}\}, \\ \mathcal{A}^f_{\mathcal{P}} & = & \{ab \leftarrow \top \mid & ab \text{ occurs in the body of a rule in } \mathcal{P} \\ & & & \text{representing a factual conditional}\}. \end{array}$$

$C \leftarrow A \wedge \neg ab$	A non-necessary	A necessary
Factual	$ab \leftarrow \top,\ C \leftarrow \top$	$ab \leftarrow \top$
Obligation	$C \leftarrow \top$	

Table 2: The additional facts in the set of abducibles for a rule of the form $C \leftarrow A \wedge \neg ab$ representing a conditional *if* A *then* C.

The set $\mathcal{A}^{nn}_{\mathcal{P}}$ contains facts for the consequents of conditional sentences with non-necessary antecedent. If an antecedent of a conditional sentence is non-necessary then there may be other unknown reasons for establishing the consequent of the conditional sentence. The set $\mathcal{A}^f_{\mathcal{P}}$ contains facts for the abnormalities occurring in the bodies of the representation of factual conditionals. The antecedent of a factual conditional may be true, yet the consequent of the conditional sentence may still not hold. Adding a fact for the abnormality predicate occurring in the body will force this abnormality to become true and its negation to become false. Hence, the body of the clause containing the abnormality predicate will be false.[3] Table 2 illustrates the new facts in the set of abducibles.

4 Predictions of WCS for Human Responses

If a conditional premise *if* A *then* C is given as the first premise, then according to (Stenning and van Lambalgen 2008) this shall be represented as a *license for inference* by the program $\mathcal{P}_c$ presented in Section 2. It is called a licence as in human reasoning it is usually not the case that all antecedents which are necessary to enforce a conclusion are mentioned. Weakly completing the program we obtain

$$\{C \leftrightarrow A \wedge \neg ab,\ ab \leftrightarrow \bot\}.$$

Computing its least model we obtain $\langle \emptyset, \{ab\} \rangle$. In this model A and C are mapped to unknown, whereas ab is mapped to false. Please note that this model is the least fixed point of the $\Phi_{\mathcal{P}_c}$ operator which can be computed by iterating the operator starting with the empty interpretation $\langle \emptyset, \emptyset \rangle$.

In the following subsections we assume that a conditional sentence *if* A *then* C is given as the first premise and consider the four different cases which occur if a second premise is added.

[3] This technique is used in (Dietz, Hölldobler, and Ragni 2012) to represent an enabling relation and model the suppression effect. In particular, *a library not being open* prevents a person from *studying in it*.

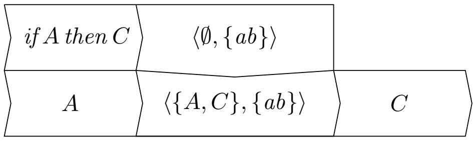

$if\,A\,then\,C$
$\langle\emptyset,\{ab\}\rangle$
A
$\langle\{A,C\},\{ab\}\rangle$
C

Figure 1: AA reasoning. The left column shows the premises, the middle column the constructed least models, and the right column the generated responses.

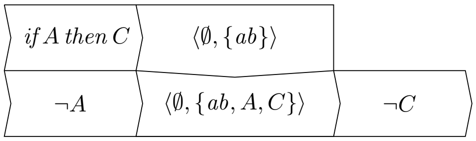

$if\,A\,then\,C$
$\langle\emptyset,\{ab\}\rangle$
$\neg A$
$\langle\emptyset,\{ab,A,C\}\rangle$
$\neg C$

Figure 2: DA reasoning.

4.1 Affirmation of the Antecedent

If the antecedent A of the conditional sentence $if\,A\,then\,C$ is affirmed as a second premise, then this is represented by the program

$$\mathcal{P}_{aa} = \mathcal{P}_c \cup \{A \leftarrow \top\}.$$

Weakly completing the program and computing its least model we obtain $\langle\{A,C\},\{ab\}\rangle$. Reasoning with respect to this model we conclude C (see Figure 1). This is independent of the classification of conditional sentences as obligation or factual or that of antecedent as necessary or non-necessary. Example 1 presented in the introduction belongs to this category with A and C denoting *it rains* and *the roofs must be wet*, respectively.

Predictions of WCS for Human Responses on AA In AA inferences with the premises $if\,A\,then\,C$ and A, most humans will answer C, and this is independent of the classification of the conditional sentence and the antecedent.

4.2 Denial of the Antecedent

If the antecedent A of the conditional sentence $if\,A\,then\,C$ is denied as a second premise, then this is represented by the program

$$\mathcal{P}_{da} = \mathcal{P}_c \cup \{A \leftarrow \bot\}.$$

Weakly completing the program and computing its least model we obtain $\langle\emptyset,\{ab,A,C\}\rangle$. Reasoning with respect to this model we conclude $\neg C$ (see Figure 2). This is independent of the classification of the conditional sentence as well as the antecedent. Example 2 presented in the introduction belongs to this category with A and C denoting *Paul rides a motorbike* and *Paul is wearing a helmet*, respectively.

Prediction of WCS for Human Responses on DA In DA inferences with the premises $if\,A\,then\,C$ and $\neg A$, most humans will answer $\neg C$, and this is independent of the classification of the conditional sentence and the antecedent.

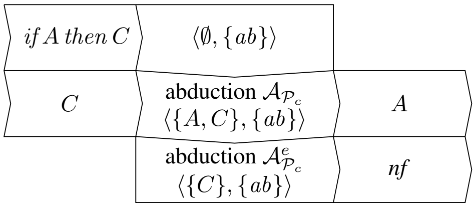

$if\,A\,then\,C$
$\langle\emptyset,\{ab\}\rangle$
C
abduction $\mathcal{A}_{\mathcal{P}_c}$
$\langle\{A,C\},\{ab\}\rangle$
A
abduction $\mathcal{A}^e_{\mathcal{P}_c}$
$\langle\{C\},\{ab\}\rangle$
nf

Figure 3: AC reasoning. The answer *nothing follows (nf)* is given if the antecedent of the conditional sentence is non-necessary, the reasoner considers $\mathcal{A}^e_{\mathcal{P}_c}$ and is reasoning skeptically.

4.3 Affirmation of the Consequent

If the consequent C of the conditional sentence $if\,A\,then\,C$ is affirmed as a second premise, then this is considered to be an observation to be explained because C is already defined in the program $\mathcal{P}_c$. But A is undefined. Hence, we obtain $\mathcal{A}_{\mathcal{P}_c} = \{A \leftarrow \top, A \leftarrow \bot\}$. $\{A \leftarrow \top\}$ is the only minimal explanation for $\{C\}$. Let

$$\mathcal{P}_{ac} = \mathcal{P}_c \cup \{A \leftarrow \top\}.$$

Weakly completing the program and computing its least model we obtain $\langle\{A,C\},\{ab\}\rangle$. Reasoning with respect to this least model we conclude A.

However, if the classification of antecedents is taken into account and if the conditional sentence has a non-necessary antecedent, then the set of abducibles will be extended by the fact $C \leftarrow \top$. In this case, there is a second minimal explanation for $\{C\}$, viz. $\{C \leftarrow \top\}$. Let

$$\mathcal{P}'_{ac} = \mathcal{P}_c \cup \{C \leftarrow \top\}.$$

Weakly completing the program and computing its least model we obtain $\langle\{C\},\{ab\}\rangle$. Taking both explanations into account and reasoning skeptically, we conclude *nothing follows (nf)* (see Figure 3).[4] The case of a conditional sentence with necessary antecedent is exemplified by Example 3 from the introduction, with A and C denoting *the library is open* and *Elisa is studying late in the library*, respectively. Here we conclude that *the library is open*.

Let us now consider an everyday conditional sentence with non-necessary antecedent. What follows from

5. *if Paul rides a motorbike, then Paul must wear a helmet* and *Paul wears a helmet*?

We expect that a significant number of humans will answer *nothing follows*.

Prediction of WCS for Human Responses on AC In AC inferences with the premises $if\,A\,then\,C$ and C, most humans will answer A. If A is a non-necessary antecedent, then the number of *nf* answers will increase. Moreover, the

[4] Formally, C and $\neg ab$ follow skeptically, but this is nothing new as C is the observation and ab is assumed to be false in the weak completion of $\mathcal{P}_c$. We would like to draw conclusions which preserve semantic information, are parsimonious, and state something new (Johnson-Laird and Byrne 1991).

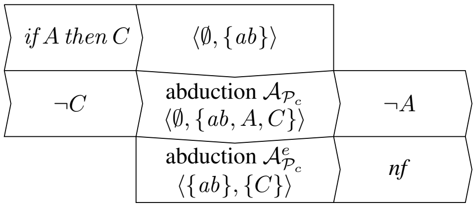

if A then C
$\langle \emptyset, \{ab\} \rangle$
$\neg C$
abduction $\mathcal{A}_{\mathcal{P}_c}$
$\langle \emptyset, \{ab, A, C\} \rangle$
$\neg A$
abduction $\mathcal{A}^e_{\mathcal{P}_c}$
$\langle \{ab\}, \{C\} \rangle$
nf

Figure 4: DC reasoning. The answer *nf* is given if the conditional sentence is a factual one, the reasoner considers $\mathcal{A}^e_{\mathcal{P}_c}$ and is reasoning skeptically.

time to generate an *nf* answer will be longer than the time to generate the answer A.

4.4 Denial of the Consequent

If the consequent C of the conditional sentence *if* A *then* C is denied as a second premise, then this is again considered to be an observation because C is already defined in $\mathcal{P}_c$. But A is undefined. Hence, we obtain $\mathcal{A}_{\mathcal{P}_c} = \{A \leftarrow \top,\ A \leftarrow \bot\}$. $\{A \leftarrow \bot\}$ is the only minimal explanation for $\{\neg C\}$. Let

$$\mathcal{P}_{dc} = \mathcal{P}_c \cup \{A \leftarrow \bot\}.$$

Weakly completing the program and computing its least model we obtain $\langle \emptyset, \{ab, A, C\} \rangle$. Reasoning with respect to this least model we conclude $\neg A$.

However, if the classification of conditional sentences is taken into account and if the sentence is a factual one, then the set of abducibles will be extended by the fact $ab \leftarrow \top$. In this case, there is a second minimal explanation for $\{\neg C\}$, viz. $\{ab \leftarrow \top\}$. Let

$$\mathcal{P}'_{dc} = \mathcal{P}_c \cup \{ab \leftarrow \top\}.$$

Weakly completing the program and computing its least model we obtain $\langle \{ab\}, \{C\} \rangle$. Taking both explanations into account and reasoning skeptically, we conclude *nf* (see Figure 4). Example 4 presented in the introduction belongs to this category with A and C denoting *Nancy rides her motorbike* and *Nancy goes to the mountains*, respectively. As this was classified as a factual conditional we expect that a significant number of humans will answer *nothing follows*.

Let us now consider an everyday obligation conditional. What follows from

6. *if Paul rides a motorbike, then Paul must wear a helmet* and *Paul does not wear a helmet*?

We expect that most participants will conclude that *Paul does not ride a motorbike*.

Prediction of WCS for Human Responses on DC In DC inferences with the premises *if* A *then* C and $\neg C$, most humans will answer $\neg A$. If the conditional sentence is a factual one, then the number of *nf* answers will increase. Moreover, the time to generate an *nf* answer will be longer than the time to generate the answer $\neg A$.

5 Putting it to the Test

The goal of our investigation is to test the predictions made in the previous section in an everyday context, i.e., in a context familiar to the participants.

5.1 Participants, materials and methods

We tested 56 logically naive participants on an online website (Prolific, prolific.co). We restricted the participants to Central Europe and Great Britain to have a similar background knowledge about weather etc. We assume that the participants had not received any education in logic beyond high school training. We took the usual precautions for such a procedure; for example, the website checked that participants were proficient speakers of English. The participants were first presented with a story followed by a first assertion ("a conditional premise"), and a second assertion ("a (possibly negated) atomic premise"), and then for each problem they had to answer the question "What follows?". Both parts were presented simultaneously. The participants responded by clicking one of the answer options. They could take as much time as they needed. Participants acted as their own controls.

The participants carried out 48 problems consisting of the 12 conditionals listed in the Appendix and solved all four inference types (AA, DA, AC, DC). They could select one of three responses: *nothing follows*, the atomic sentence that had not been presented in the second premise, and the negation of this atomic sentence. We chose the content based on (i) previously tested conditional sentences in the literature and (ii) on everyday context. The classification of the conditional sentences was done by the authors.

As an example consider the following story: *Peter has a lawn in front of his house. He is keen to make sure that the grass on lawn does not dry out, so whenever it has been dry for multiple days, he turns on the sprinkler to water the lawn.* Then, the conditional sentence *if it rains, then the lawn is wet* and the negated atomic sentence *the lawn is not wet* are given. In this case, the three answers from which participants could select were *it rains*, *it does not rain*, and *nothing follows*.

6 Evaluation

6.1 Affirmation of the Antecedent

The total number of selected responses as well as the median response time (in milliseconds) for C (*Mdn* C) and *nf* (*Mdn nf*) responses can be found in Table 3 for AA inferences that is for a given conditional *if* A *then* C and fact A.

The everyday context elicited a high response rate of AA inferences of about 95% (640 out of 672) for C-answers. The number of participants answering $\neg C$ or *nf* as well as the classification of conditional sentences appears to be irrelevant. The WCS models human AA inferences adequately.

6.2 Denial of the Antecedent

The total number of selected responses as well as the median response time (in milliseconds) for $\neg C$ (*Mdn* $\neg C$) and

Class	C	$\neg C$	*nf*	Sum	$Mdn\,C$	$Mdn\,nf$
(1)	55	1	0	56	3343	na
(2)	55	1	0	56	3487	na
(3)	53	3	0	56	3516	na
ON	163	5	0	168	3408	na
(4)	53	1	2	56	3403	3472
(5)	53	2	1	56	3903	3572
(6)	54	1	1	56	3088	6959
ONN	160	4	4	168	3543	4183
(7)	49	1	6	56	3885	7051
(8)	54	1	1	56	3559	7349
(9)	54	1	1	56	3710	3826
FN	157	3	8	168	3615	6926
(10)	51	2	3	56	3929	6647
(11)	54	1	1	56	3777	5073
(12)	55	1	0	56	2977	na
FNN	160	4	4	168	3644	5860
Obligation	323	9	4	336	3516	4183
Factual	317	7	12	336	3640	6575
Necessary	320	8	8	336	3546	6926
Non-nec	320	8	8	336	3588	4934
Total	640	16	16	672	3570	5925

Table 3: The results for AA inferences. The grey line shows the numbers for Example 1. '*na*' is an acronym for *not applicable*. 'ON' refers to obligation conditionals with necessary antecedent, which are the conditional sentences (1) - (3) in the experiment. 'ONN' refers to obligation conditionals with non-necessary antecendent, which are the conditional sentences (4) - (6) in the experiment. 'FN refers to factual conditionals with necessary antecendent, which are the conditional sentences (7) - (9) in the experiment. 'FNN' refers to factual conditionals with non-necessary antecedent, which are the conditional sentences (10) - (12) in the experiment. In the lines labeled 'Obligation' and 'Factual' the results for obligation and factual conditionals are shown, respectively. In the lines labeled 'Necessary' and 'Non-nec' the results for conditionals with necessary and non-necessary antecedents are shown, respectively. The line labeled 'Total' shows the results for all experiments.

Class	C	$\neg C$	*nf*	Sum	$Mdn\,\neg C$	$Mdn\,nf$
(1)	0	45	11	56	2863	4901
(2)	2	54	0	56	3367	na
(3)	2	51	3	56	3647	10477
ON	4	150	14	168	3356	5115
(4)	1	40	15	56	3722	7189
(5)	3	28	25	56	5735	7814
(6)	4	36	16	56	3602	6240
ONN	8	104	56	168	4064	7471
(7)	2	51	3	56	3928	7273
(8)	1	47	8	56	3296	5728
(9)	1	52	3	56	3549	8735
FN	4	150	14	168	3605	6582
(10)	1	39	16	56	3725	6874
(11)	0	41	15	56	3374	5887
(12)	1	41	14	56	3205	7002
FNN	2	121	45	168	3374	6221
Obligation	12	254	70	336	3583	6613
Factual	6	271	59	336	3518	6221
Necessary	8	300	28	336	3474	5808
Non-nec	10	225	101	336	3646	6700
Total	18	525	129	672	3558	6450

Table 4: The results for DA inferences. The grey line shows the numbers for Example 2. If the antecedent is non-necessary, then *nf* is answered significantly more often (grey cells).

nf ($Mdn\,nf$) responses can be found in Table 4 for DA inferences that is for a given conditional sentence *if* A *then* C and atomic sentence $\neg A$.

The everyday context elicited a high response rate of DA inferences of about 78% (525 out of 672) for $\neg C$-answers, but the case of *nf*-answers varied from 8% (14 out of 168) up to 33% (56 out of 168). The number of participants answering C is irrelevant.

The answer *nf* was more often given in case of conditional sentences with non-necessary antecedents than in the case of conditional sentences with necessary antecedents (30% vs. 8%, Wilcoxon signed rank, $W = 0$, $p < .001$). The WCS predicts the answer $\neg C$ given by the majority of the participants, but it cannot model the difference of the *nf*-answers. We speculate that in case of an *nf*-answer the clauses representing conditional sentences with non-necessary antecedents should not be weakly completed. This would require a modification to the semantic definitions of WCS, whose theoretical and algorithmic properties have not yet been investigated.

Class	A	$\neg A$	*nf*	Sum	$Mdn\ A$	$Mdn\ nf$
(1)	37	1	18	56	3952	7995
(2)	48	1	7	56	4003	4170
(3)	43	1	12	56	3458	9001
ON	128	3	37	168	3797	8175
(4)	42	1	13	56	3659	8828
(5)	32	1	23	56	4704	6044
(6)	29	1	26	56	3593	4396
ONN	103	3	62	168	3968	5939
(7)	51	1	4	56	3767	4397
(8)	42	1	13	56	3798	4565
(9)	45	1	10	56	3492	4598
FN	138	3	27	168	3699	4565
(10)	34	2	20	56	5224	6289
(11)	29	2	25	56	3218	6205
(12)	33	1	22	56	3483	4992
FNN	96	5	67	168	3885	6116
Obligation	231	6	99	336	3888	6044
Factual	234	8	94	336	3769	5650
Necessary	266	6	64	336	3735	5450
Non-nec	199	8	129	336	3906	6039
Total	465	14	193	672	3826	5802

Table 5: The results for AC inferences. The grey lines show the results for Examples 3 (line marked (7)) and 5 (line marked (4)). If the antecedent is non-necessary, then *nf* is answered significantly more often (grey cells).

Class	A	$\neg A$	*nf*	Sum	$Mdn\ \neg A$	$Mdn\ nf$
(1)	1	45	10	56	3449	4758
(2)	0	50	6	56	4058	7922
(3)	2	46	8	56	3796	4517
ON	3	141	24	168	3767	5732
(4)	3	46	7	56	3872	4154
(5)	1	54	1	56	4946	8020
(6)	0	36	20	56	4062	5235
ONN	4	136	28	168	4293	5803
(7)	1	37	18	56	5974	4744
(8)	3	42	11	56	4367	5013
(9)	0	47	9	56	4208	3966
FN	4	126	38	168	4849	4574
(10)	2	35	19	56	4879	4167
(11)	0	39	17	56	4411	5647
(12)	0	34	22	56	3726	3813
FNN	2	108	58	168	4338	4542
Obligation	7	277	52	336	4053	4790
Factual	6	234	96	336	4459	4345
Necessary	7	267	62	336	4096	4758
Non-nec	6	244	86	336	4325	4555
Total	13	511	148	672	4311	5162

Table 6: The results for DC inferences. The grey lines show the results for Examples 4 (line marked (10)) and 6 (line marked (4)). In case of factual conditionals, *nf* is answered significantly more often (grey cells).

6.3 Affirmation of the Consequent

The total number of selected responses as well as the median response time (in milliseconds) for A ($Mdn\ A$) and *nf* ($Mdn\ nf$) responses can be found in Table 5 for AC inferences that is for a given conditional sentence $if\ A\ then\ C$ and atomic sentence C.

The everyday context elicited a high response rate of AC inferences of about 69% (465 out of 672) for A-answers, but the case of *nf*-answers varied from 16% (27 out of 168) up to 40% (67 out of 168). The number of participants answering $\neg A$ is irrelevant.

As predicted by the WCS, the answer *nf* was more often given in case of conditional sentences with non-necessary antecedents than in the case of sentences with necessary antecedents (38% vs. 19%, Wilcoxon signed rank, $W = 82$, $p < .001$).

6.4 Denial of the Consequent

The total number of selected responses as well as the median response time (in milliseconds) for $\neg A$ ($Mdn\ \neg A$) and *nf* ($Mdn\ nf$) responses can be found in Table 6 for DC inferences that is for a given conditional sentence $if\ A\ then\ C$ and negative atomic sentence $\neg C$.

The everyday context elicited a high response rate of DC inferences of about 76% (511 out of 672) for $\neg A$-answers, but the case of *nf*-answers varied from 14% (24 out of 168) up to 35% (58 out of 168). The number of participants answering A is irrelevant.

As predicted by the WCS, the answer *nf* was more often given in case of a factual conditional than in case of an obligation conditional (35% vs. 14%, Wilcoxon signed rank, $W = 133$, $p < .001$). So the predicted increase in the selection of *nf* can be confirmed.

6.5 Interpreting the Results

For each conditional sentence used in the experiments and for each type of inference, the WCS correctly predicted the answer given by a majority of the participants. This can be explained in classical, two-valued logic if one assumes that each conditional sentence used in the experiments was erroneously considered to be a bi-conditional by the majority. This is quite surprising given that six of the twelve antecedents of the conditional sentences used in the experiment were classified as non-necessary (see Appendix).

Moreover, the WCS correctly predicted the rising number of *nf*-answers in AC inferences if the antecedent was non-necessary and in DC inferences if the conditional sentence was a factual one.

Given an AA inference task, reasoners just conclude the consequent of the conditional sentence. This corresponds to modus ponens. Reasoners are familiar with this kind of inference and deviate seldomly.

Given a DA inference task, most reasoners conclude the negation of the consequent of the conditional sentence as predicted. One should note that the median response time of the answer C in AA inferences and the median response time of the answer $\neg C$ in DA inferences are almost identical (3570 vs. 3558). This can also be explained by the WCS in that the steps taken to construct the least models in AA and DA inference tasks are very similar. In each case, the semantic operator $\Phi_{\mathcal{P}}$ needs to be applied twice to reach a fixed point. In the connectionist network implementing the semantic operator (Hölldobler and Kencana Ramli 2009b) the stable states corresponding to the least fixed points are computed in six steps in both cases.

Given an AC or DC inference task, reasoners may search for a minimal explanation of $\{C\}$ or $\{\neg C\}$ using the set $\mathcal{A}_{\mathcal{P}}$ of abducibles. Such a minimal explanation always exists and gives rise to a model that maps the antecedent A of the given conditional sentence *if* A *then* C to either true or false, respectively. This model may be called the *preferred* model in the sense of (Ragni and Knauff 2013). Once the preferred model has been constructed, a reasoner may upon further thought search for models using the extended set $\mathcal{A}^e_{\mathcal{P}} \supseteq \mathcal{A}_{\mathcal{P}}$ of abducibles and find a second minimal explanation, giving rise to a second model. In this second model A is unknown. Reasoning skeptically, the reasoner will answer *nf*. This not only explains the difference between necessary and non-necessary antecedents or obligation and factual conditionals but also why a significantly larger number of participants answered *nf* in the case of non-necessary antecedents and factual conditionals, respectively. In order to fully support his interpretation of the results, further experiments recording the time of deliberation are required.

WCS correctly predicts that the answer *nf* appears significantly less frequently for AC inferences with a necessary antecedent as well as for DC inferences with an obligation conditional. However, even though the answer *nf* does appear significantly less frequently in these cases, the number of *nf* answers for these inference tasks is not as insignificantly small as in the case of AA inferences. This is not predicted by WCS, but may have multiple various reasons: Some reasoners might not consider C or $\neg C$ as an observation that needs to be explained. Rather, if they might just add $C \leftarrow \top$ or $C \leftarrow \bot$ to the program, in which case no model assigning A to true or false can be constructed. Some reasoners might consider C or $\neg C$ as an observation that needs to be explained but not necessarily by A or $\neg A$, respectively; moreover, the classification of the given conditional sentence may depend on the cultural background of the reasoner. Or the reasoners may make a mistake in constructing the preferred model, which is – as mentioned before – the least fixed point of the semantic operator introduced in (Stenning and van Lambalgen 2008).

One should observe that it took participants less time to answer C and $\neg C$ in AA and DA inferences compared to the time to answer A and $\neg A$ in AC and DC inferences. This is a well-known phenomenon, as AC and DC inferences are considered to be more difficult than AA and DA ones (see e.g. (Barrouillet, Grosset, and Lecas 2000)). This can also be explained by the WCS: In AA and DA inferences it suffices to compute the least fixed point of the semantic operator, whereas in AC and DC inferences abduction needs to be added considering the consequent or its negation as an observation to be explained. Apart from that, at the moment we can only speculate why the median response time of $\neg A$-answers in DC inferences is larger than the median response time of A-answers in AC inferences: This may depend on the sequences in which possible explanations for the observations are considered; in particular, the possible explanation $\{A \leftarrow \top\}$ may have been considered before the possible explanation $\{A \leftarrow \bot\}$.

The second hypothesis is, however, a core question: is answering *nf* an indication that a participant did not know the answer, or is it at the end of a deliberation process that might follow the predicted process in the WCS? While this answer cannot be given in general, the median response times for *nf* are higher than for the respective responses A, $\neg A$, C, or $\neg C$. This often indicates more thinking and less guessing, because participants do not quickly and easily respond *nf* to avoid thinking. These findings indicate towards the processes predicted for each inference type in the WCS, but for further support more studies are necessary.

7 Summary and Outlook

As shown in this paper, the WCS adequately models human conditional reasoning in that it generates the answers given by a majority of the participants. It is based on several principles: (1) Conditional sentences are represented as licenses for inference in a (logic) program. (2) Abnormality predicates are used to represent unknown additional conditions; they are initially assumed to be false. (3) The definitions given in a program are weakly completed. (4) Programs are interpreted under the three-valued Łukasiewicz logic. (5) A positive or negative atomic sentence given as a premise is considered to be an observation which needs to be explained if the program already contains a definition for the atomic sentence. (6) Skeptical abduction is applied. (7) The Gricean implicature is applied.

These principles are well justified. In most human reasoning scenarios not all necessary antecedents of a conditional sentence will or can be given. The abnormality predicate takes care of this. If a detail becomes important later on, then this can be added. Reconsider Example 4, the *absence of gas will prevent Nancy from riding the motorbike to the mountains*. *Sufficient amount of gas* is an enabling relation for riding a motorbike. This can be modeled by adding the rule $ab \leftarrow \neg gas$ to the program representing Example 4. The new rule will override $ab \leftarrow \bot$ when the weak completion is computed as $ab \leftrightarrow \bot \vee \neg gas$ is semantically equivalent to $ab \leftrightarrow \neg gas$. In general, positive information will override a negative one. During the weak completion process, the only-if halves of definitions[5] are added, which is based on ideas underlying *conditional perfection* in linguistics (Van der Auwera 1997).

It has been shown that two-valued logics cannot model human reasoning (Ragni et al. 2016). Furthermore, under the three-valued Łukasiewics logic, programs and their weak completions have least models. This does not hold for the three-valued Kleene logic (Kleene 1952). There, $\mathsf{U} \leftarrow \mathsf{U} = \mathsf{U}$ and, consequently, programs like $\{a \leftarrow b\}$ have two minimal models $\langle\{a, b\}, \emptyset\rangle$ and $\langle\emptyset, \{a, b\}\rangle$, but no least one. Reasoning credulously does not adequately model human reasoning as shown in this paper, because credulous reasoning does not account for the growing number of *nf*-responses in AC and DC inferences if conditionals are classified. The Gricean implicature has been applied at various occasions: an atom can only be false if there is evidence for this falsehood; otherwise it is unknown. This can be seen in the definition of weak completion as well as in the definition of the Φ-operator. Likewise, skeptical consequences are only defined if the observations are explainable.

The WCS constructs models, which are considered to be mental models in the sense of (Craik 1945) and (Johnson-Laird 1983). Reasoning is performed with respect to the constructed mental models. The WCS is non-monotonic, multi-valued, and the background knowledge need not be consistent which might be closer to humans than to formal databases. Furthermore, the semantic operator, which is used to construct the models can be represented as a feed-forward network (Hölldobler and Kencana Ramli 2009b; Dietz Saldanha et al. 2018a), can be learned (d'Avila Garcez and Zaverucha 1999; Besold et al. 2017), and can be applied to model the average human reasoner. Thus, it suggests a solution for the five fundamental problems for logical models of human reasoning discussed in (Oaksford and Chater 2020). Moreover, the WCS is computational, meaning answers to queries are computed. It is also comprehensive in that different human reasoning tasks can be modeled without changing the theory. For example, in (Dietz, Hölldobler, and Ragni 2012) it is shown that the suppression task (Byrne 1989) is modeled adequately by the WCS. In (Oliviera da Costa et al. 2017) it is shown that the WCS models human syllogistic reasoning better than the twelve cognitive theories investigated in (Khemlani and Johnson-Laird 2012) and in (Dietz Saldanha et al. 2018b) it was shown how ethical decision problems can be modeled by the WCS. The WCS as shown here, is a formally founded approach that can explain human reasoning and is a bridging system at the intersection between human and formal reasoning.

However, much remains to be done and we have already raised various open questions in the paper. How can the increased number of *nf*-answers in DA inferences be explained? How can we distinguish between the third truth value 'unknown' and 'I don't have a clue' in the answers of participants? How is WCS related to MMT?

Acknowledgements

We like to thank Meghna Bhadra, Islam Hamada, and Claudia Schon for many suggestions to improve the readability of the paper and Sangeet S. Khemlani for many fruitful discussions on cognitive and formal theories.

Appendix: Conditionals of the Experiment

Obligation Conditionals with Necessary Antecedent (ON) (1) *If it rains, then the roofs must be wet.* (2) *If water in the cooking pot is heated over $99°C$, then the water starts boiling.* (3) *If the wind is strong enough, then the sand is blowing over the dunes.*

Obligation Conditionals with Non-Necessary Antecedent (ONN) (4) *If Paul rides a motorbike, then Paul must wear a helmet.* (5) *If Maria is drinking alcoholic beverages in a pub, then Maria must be over 19 years of age.* (6) *If it rains, then the lawn must be wet.*

Factual Conditionals with Necessary Antecedent (FN) (7) *If the library is open, then Sabrina is studying late in the library.* (8) *If the plants get water, then they will grow.* (9) *If my car's start button is pushed, then the engine will start running.*

Factual Conditionals with Non-Necessary Antecedent (FNN) (10) *If Nancy rides her motorbike, then Nancy goes to the mountains.* (11) *If Lisa plays on the beach, then Lisa will get sunburned.* (12) *If Ron scores a goal, then Ron is happy.*

References

Barrouillet, P.; Grosset, N.; and Lecas, J.-F. 2000. Conditional reasoning by mental models: Chronometric and developmental evidence. *Cognition* 75:237–266.

Besold, T. R.; d'Avila Garcez, A. S.; Bader, S.; Bowman, H.; Domingos, P. M.; Hitzler, P.; Kühnberger, K.-U.; Lamb, L. C.; Lowd, D.; Lima, P. M. V.; de Penning, L.; Pinkas, G.; Poon, H.; and Zaverucha, G. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR* abs/1711.03902.

Bibel, W. 1991. Perspectives on automated deduction. In Boyer, R. S., ed., *Automated Reasoning: Essays in Honor of Woody Bledsoe*. Dordrecht: Kluwer Academic. 77–104.

Byrne, R. M. J. 1989. Suppressing valid inferences with conditionals. *Cognition* 31(1):61–83.

Byrne, R. M. J. 2005. *The Rational Imagination: How People Create Alternatives to Reality*. Cambridge, MA, USA: MIT Press.

Clark, K. L. 1978. Negation as failure. In Gallaire, H., and Minker, J., eds., *Logic and Databases*. New York: Plenum. 293–322.

Craik, K. J. W. 1945. *The Nature of Explanation*. Cambridge: Cambridge University Press.

d'Avila Garcez, A. S., and Zaverucha, G. 1999. The connectionist inductive learning and logic programming system. *Applied Intelligence* 11(1):69–77.

[5] The weak completion of the definition $A \leftarrow C \wedge \neg ab$ is $A \leftrightarrow C \wedge \neg ab$, whereas the bi-conditional corresponding to the conditional *if A then C* is *A iff C*.

Dietz, E.-A.; Hölldobler, S.; and Ragni, M. 2012. A computational logic approach to the suppression task. In Miyake, N.; Peebles, D.; and Cooper, R. P., eds., *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, 1500–1505. Cognitive Science Society.

Dietz Saldanha, E.-A.; Hölldobler, S.; Kencana Ramli, C. D. P.; and Palacios Medinacelli, L. 2018a. A core method for the weak completion semantics with skeptical abduction. *Journal of Artificial Intelligence Research* 63:51–86.

Dietz Saldanha, E.-A.; Hölldobler, S.; Schwarz, S.; and Stefanus, L. Y. 2018b. The weak completion semantics and equality. In G. Barthe, G. Sutcliffe, M. V., ed., *LPAR-22. Proceedings of the 22nd International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, volume 57, 326–242. EPiC series in Computing.

Dietz Saldanha, E.-A.; Hölldobler, S.; and Lourêdo Rocha, I. 2017. Obligation versus factual conditionals under the weak completion semantics. In Hölldobler, S.; Malikov, A.; and Wernhard, C., eds., *Proceedings of the Second Young Scientists' International Workshop on Trends in Information Processing*, volume 1837, 55–64. CEUR-WS.org. http://ceur-ws.org/Vol-1837/.

Fitting, M. 1996. *First–Order Logic and Automated Theorem Proving*. Berlin: Springer-Verlag, 2nd edition.

Grice, H. P. 1975. Logic and conversation. In Cole, P., and Morgan, J. L., eds., *Syntax and Semantics*, volume 3. Academic Press, New York. 41–58.

Hölldobler, S., and Kencana Ramli, C. D. P. 2009a. Logic programs under three-valued Łukasiewicz's semantics. In Hill, P. M., and Warren, D. S., eds., *Logic Programming*, volume 5649 of *Lecture Notes in Computer Science*, 464–478. Springer-Verlag Berlin Heidelberg.

Hölldobler, S., and Kencana Ramli, C. D. P. 2009b. Logics and networks for human reasoning. In Alippi, C.; Polycarpou, M. M.; Panayiotou, C. G.; and Ellinasetal, G., eds., *Artificial Neural Networks – ICANN*, volume 5769 of *Lecture Notes in Computer Science*, 85–94. Springer-Verlag Berlin Heidelberg.

Hölldobler, S. 2015. Weak completion semantics and its applications in human reasoning. In Furbach, U., and Schon, C., eds., *Bridging 2015 – Bridging the Gap between Human and Automated Reasoning*, volume 1412 of *CEUR Workshop Proceedings*, 2–16. CEUR-WS.org. http://ceur-ws.org/Vol-1412/.

Johnson-Laird, P. N., and Byrne, R. M. J. 1991. *Deduction*. Hove and London (UK): Lawrence Erlbaum Associates.

Johnson-Laird, P. N., and Byrne, R. M. J. 2002. Conditionals: A theory of meaning, pragmatics, and inference. *Psychological Review* 109:646–678.

Johnson-Laird, P. N. 1983. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge: Cambridge University Press.

Kakas, A. C.; Kowalski, R. A.; and Toni, F. 1992. Abductive Logic Programming. *Journal of Logic and Computation* 2(6):719–770.

Khemlani, S., and Johnson-Laird, P. N. 2012. Theories of the syllogism: A meta-analysis. *Psychological Bulletin* 138(3):427–457.

Khemlani, S. S.; Byrne, R. M. J.; and Johnson-Laird, P. N. 2018. Facts and possibilities: A model-based theory of sentenial reaoning. *Cognitive Science* 1–38.

Kleene, S. C. 1952. *Introduction to Metamathematics*. North-Holland.

Lloyd, J. W. 1984. *Foundations of Logic Programming*. Springer-Verlag.

Łukasiewicz, J. 1920. O logice trójwartościowej. *Ruch Filozoficzny* 5:169–171.

Oaksford, M., and Chater, N. 2020. New paradigms in the psychology of reasoning. *Annual Review of Psychology* 71:12.1–12.26.

Oliviera da Costa, A.; Dietz Saldanha, E.-A.; Hölldobler, S.; and Ragni, M. 2017. A computational logic approach to human syllogistic reasoning. In Gunzelmann, G.; Howes, A.; Tenbrink, T.; and Davelaar, E. J., eds., *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, 883–888. Austin, TX: Cognitive Science Society.

Ragni, M., and Knauff, M. 2013. A theory and a computational model of spatial reasoning. *Psychological Review* 120:561–588.

Ragni, M.; Dietz, E.-A.; Kola, I.; and Hölldobler, S. 2016. Two-valued logic is not sufficient to model human reasoning, but three-valued logic is: A formal analysis. In Furbach, U., and Schon, C., eds., *Bridging 2016 – Bridging the Gap between Human and Automated Reasoning*, volume 1651 of *CEUR Workshop Proceedings*, 61–73. CEUR-WS.org. http://ceur-ws.org/Vol-1651/.

Stenning, K., and van Lambalgen, M. 2008. *Human Reasoning and Cognitive Science*. MIT Press.

Van der Auwera, J. 1997. Pragmatics in the last quarter century: The case of conditional perfection. *Journal of Pragmatics* 27(3):261–274.

Forgetting Formulas and Signature Elements in Epistemic States

Alexander Becker[1] , **Gabriele Kern-Isberner**[1] , **Kai Sauerwald**[2] , **Christoph Beierle**[2]

[1]TU Dortmund University, Germany
[2]FernUniversität in Hagen, Germany

{alexander2.becker, gabriele.kern-isberner}@tu-dortmund.de,
{christoph.beierle, kai.sauerwald}@fernuni-hagen.de

Abstract

Delgrande's knowledge level account of forgetting provides a general approach to forgetting syntax elements from sets of formulas with links to many other forgetting operations, in particular, to Boole's variable elimination. On the other hand, marginalisation of epistemic states is a specific approach to actively reduce signatures in more complex semantic frameworks, also aiming at forgetting atoms that is very well known from probability theory. In this paper, we bring these two perspectives of forgetting together by showing that marginalisation can be considered as an extension of Delgrande's approach to the level of epistemic states. More precisely, we generalize Delgrande's axioms of forgetting to forgetting in epistemic states, and show that marginalisation is the most specific and informative forgetting operator that satisfies these axioms. Moreover, we elaborate suitable phrasings of Delgrande's concept of forgetting for formulas by transferring the basic ideas of the axioms to forgetting formulas from epistemic states. However, here we show that this results in trivial approaches to forgetting formulas. This finding supports the claim that forgetting syntax elements is essentially different from belief contraction, as e.g. axiomatized in the AGM belief change framework.

1 Introduction

In the past decade, the popularity and presence of artificial intelligence (AI) grew rapidly and thereby reached almost every part of our daily lives. From product and media recommendations, voice assistants, and smart homes over industrial optimizations, medical research, and traffic, to even criminal prosecution. And most probably, the importance of AI will grow even further in the near future, due to the ever-increasing amount of data that accumulates day by day and the huge potential it carries. However, so far only little attention was given to the concept of forgetting, even though it plays an essential role in many areas of our daily lives as well. In 2018 the General Data Protection Regulation (GDPR) became applicable, which gives every citizen of the European Union the right to be forgotten (GDPR - Article 17). This raises the question what it actually means to forget something, and whether it is sufficient to only delete some data in order to forget certain information. This is clearly not the case, since AI systems fitted on this data might still be able to infer the information we like to forget. Thus, forgetting is far more complex than just deleting data. From a cognitive point of view, forgetting is an inextricable part of any learning process that helps handling information overload, sort out irrelevant information, and resolve contradictions. Moreover, it is also of importance when it comes to knowledge management in organisational contexts (Kluge et al. 2019), socio-digital systems (Ellwart et al. 2019), and domains with highly dynamic information such as supply chain and network management. These few examples illustrate the importance of forgetting in AI systems to guarantee individual privacy and informational self-determination, but also efficient reasoning by blinding out irrelevant information.

In the domain of logic and knowledge representation, several logic-specific forgetting definitions exist, e.g. Boole's variable elimination (Boole 1854), fact forgetting in first-order logic (Lin and Reiter 1994) and forgetting in modal logic (Baral and Zhang 2005). However, none of these specific approaches argued about the general notions of forgetting, but rather provided a way to compute its result. In (Delgrande 2017), Delgrande presented a general forgetting approach with the goal to unify many of the hitherto existing logic-specific approaches. Moreover, he stated a set of properties he refers to as *right* and *desirable* when it comes to the notions of forgetting. In contrast to Delgrande's approach, Beierle et al. (2019) presented a general framework for cognitively different kinds of forgetting, which also consider the common-sense understanding of forgetting, and their realisation by means of ordinal conditional functions. In the following, we take this broad, common-sense motivated view of forgetting in contrast to the viewpoint put forward by Delgrande, who explicitly states that e.g. the belief change of contraction should not be considered as forgetting.

In this work, we show that Delgrande's forgetting approach is included in and even generalized by the cognitively different kinds of forgetting presented in (Beierle et al. 2019), concretely by means of the marginalisation. Moreover, we show that the forgetting properties Delgrande refers to as *right* and *desirable* are not suitable to axiomatise the general properties of all kinds of forgetting, but only of those that aim to forget signature elements instead of formulas. Thus, the here presented results form another step towards a general framework for different kinds of forgetting, and provide a deeper understanding of their properties and inherent differences.

Finally, we want to give an overview of how this work is structured. In Section 2, we give all the preliminaries needed in the later sections including model theoretical basics and ordinal conditional functions. Then we will present both of the above-mentioned general forgetting approaches in Section 3 and show that the marginalisation extends Delgrande's forgetting to epistemic states, since both approaches always result in the same posterior beliefs. In Section 4, we will then generalize and extend the properties stated by Delgrande to epistemic states, and show that the marginalisation satisfies all of them. Moreover, we show that the marginalisation is the most specific approach satisfying these properties. Finally, we extend the same properties to forgetting formulas in epistemic states and show that they are not suitable for axiomatizing general properties of forgetting, since they imply trivial approaches of forgetting formulas. In Section 6, we present our conclusions as well as some outlooks for future works.

2 Formal Basics

In the following, we introduce the formal basics as needed in this work. With $\mathcal{L}_\Sigma$ we state a propositional language over the finite signature Σ with formulas $\varphi, \psi \in \mathcal{L}_\Sigma$. The corresponding interpretations are denoted as Ω_Σ. The interpretations $\omega \in \Omega_\Sigma$ that satisfy a formula $\varphi \in \mathcal{L}_\Sigma$, i.e. $\omega \models \varphi$, are called models of φ and are denoted as $\llbracket \varphi \rrbracket_\Sigma$. If the signature of a model set is unambiguously given by the context, we also write $\llbracket \varphi \rrbracket$ instead. The explicit declaration of the corresponding signature is of particular importance when arguing about different (sub-)signatures. Moreover, each model $\omega \in \Omega_\Sigma$ can also be considered as a conjunction of literals corresponding to the truth values ω assigns to each signature element $\rho \in \Sigma$. Thus, we can also write $\omega \models \omega'$, where $\omega, \omega' \in \Omega_\Sigma$, but ω' is considered to be the conjunction of literals corresponding to the interpretation. Note that we will make use of this notation several times in this paper. When we specifically want to argue about some signature elements in an interpretation $\omega \in \Omega_\Sigma$, we denote those signature elements $\rho \in \Sigma$ as $\dot{\rho}$ for which the concrete truth assignment is not needed, e.g. $p\dot{b}\dot{f} \in \Omega_\Sigma$ with $\Sigma = \{p, b, f\}$. For two formulas $\varphi, \psi \in \mathcal{L}_\Sigma$, we say that φ infers ψ, denoted as $\varphi \models_\Sigma \psi$, if and only if $\llbracket \varphi \rrbracket \subseteq \llbracket \psi \rrbracket$. In case that both model sets are equal, φ and ψ are equivalent, i.e. $\varphi \equiv \psi$, iff $\varphi \models \psi$ and $\psi \models \varphi$. Furthermore, the deductively closed set of all formulas that can be inferred from a formula $\varphi \in \mathcal{L}_\Sigma$ is given by $Cn_\Sigma(\varphi) = \{\psi \in \mathcal{L}_\Sigma \mid \varphi \models_\Sigma \psi\}$. Again, the signature in the index of the Cn operator as well as $\models$ can be omitted when its clearly given by the context. Notice that a formula $\varphi \in \mathcal{L}_\Sigma$ is always equivalent to its deductive closure, since their models are equal. The deductive closure $Cn_\Sigma(\varphi)$ of a formula $\varphi \in \mathcal{L}_\Sigma$ can also be expressed by means of the theory $Th(\llbracket \varphi \rrbracket) = \{\psi \in \mathcal{L}_\Sigma \mid \llbracket \varphi \rrbracket \models \psi\}$ of its models $\llbracket \varphi \rrbracket$. All of the above-mentioned formal basics also hold for sets of formulas $\Gamma \subseteq \mathcal{L}_\Sigma$.

In order to argue about inferences and models in different (sub-)signatures, further basic terms are needed. For two interpretations $\omega, \omega' \in \Omega_\Sigma$, we say that ω and ω' are elementary equivalent with the exception of the signature elements P, denoted as $\omega \equiv_P \omega'$, if and only if they agree on the truth values they assign to all signature elements in $\Sigma \setminus P$ (Delgrande 2017). Furthermore, we define the reduction and expansion of models in Def. 1, which allow us to argue about models in sub- or super-signatures as well.

Definition 1. (Delgrande 2017) *Let* $\Sigma' \subseteq \Sigma$ *be signatures and* $\varphi \in \mathcal{L}_\Sigma$, $\varphi' \in \mathcal{L}_{\Sigma'}$ *formulas. The* reduction to Σ' *of models* $\llbracket \varphi \rrbracket_\Sigma$ *is defined as*

$$(\llbracket \varphi \rrbracket_\Sigma)_{|\Sigma'} = \{\omega' \in \Omega_{\Sigma'} \mid \textit{there is } \omega \in \llbracket \varphi \rrbracket_\Sigma \textit{ s.t. } \omega \models_\Sigma \omega'\}.$$

The expansion to Σ *of models* $\llbracket \varphi' \rrbracket_{\Sigma'}$ *is defined as*

$$(\llbracket \varphi' \rrbracket_{\Sigma'})_{\uparrow\Sigma} = \bigcup_{\omega' \in \llbracket \varphi' \rrbracket_{\Sigma'}} \omega'_{\uparrow\Sigma},$$

where $\omega'_{\uparrow\Sigma} = \{\omega \in \Omega_\Sigma \mid \omega \models_\Sigma \omega'\}$*. Thereby,* $\omega \models_\Sigma \omega'$ *denotes that* $\omega \in \Omega_\Sigma$ *is more specific than* $\omega' \in \Omega_{\Sigma'}$ *w.r.t.* Σ*, which holds if and only if* $\omega_{|\Sigma'} = \omega'$*.*

Notice that multiple subsequently performed reductions $(\llbracket \varphi \rrbracket_{|\Sigma'})_{|\Sigma''}$ can be reduced to a single reduction $\llbracket \varphi \rrbracket_{|\Sigma''}$, if the signature Σ'' is a subset of Σ'.

In this work, we generally argue about epistemic states in the form of ordinal conditional functions (OCFs) introduced in a more general form by Spohn (1988). An OCF κ is a ranking function that assigns a rank $r \in \mathbb{N}_0$ to each interpretation $\omega \in \Omega_\Sigma$ with $\kappa^{-1}(0) \neq \emptyset$. The rank of an interpretation can be understood as a degree of plausibility, where $\kappa(\omega) = 0$ means that ω is most plausible. The most plausible interpretations according to an OCF κ are also called models of κ, and are therefore denoted by $\llbracket \kappa \rrbracket_\Sigma$. The rank of formula $\kappa(\varphi) = \min\{\kappa(\omega) \mid \omega \in \llbracket \varphi \rrbracket\}$ is given by the minimal rank of its models, where $\kappa(\varphi \vee \psi) = \min\{\kappa(\varphi), \kappa(\psi)\}$. The beliefs of an OCF $Bel_\Sigma(\kappa) = \{\varphi \in \mathcal{L}_\Sigma \mid \llbracket \kappa \rrbracket \models \varphi\}$ is the deductively closed set of formulas $\varphi \in \mathcal{L}_\Sigma$ that are satisfied by the OCF's models $\llbracket \kappa \rrbracket_\Sigma$. Instead of $Bel_\Sigma(\kappa) \models \varphi$, we also write $\kappa \models \varphi$.

3 Delgrande's Forgetting and Marginalisation

In this section, we will first introduce Delgrande's general forgetting approach (Delgrande 2017) as well as some of its most important properties. Afterwards, we consider the OCF marginalisation as a kind of forgetting (Beierle et al. 2019) and show that it generalizes Delgrande's definition to epistemic states.

3.1 Delgrande's General Forgetting Approach

In (Delgrande 2017), Delgrande defines a general forgetting approach with the goal to unify many of the hitherto existing logic-specific forgetting definitions, e.g. forgetting in propositional logic (Boole 1854), first-order logic (Lin and Reiter 1994), or answer set programming (Wong 2009; Zhang and Foo 2006). While most of these logic-specific approaches depend on the syntactical structure of the knowledge, Delgrande defines forgetting on the knowledge level itself, which means that it is independent of any syntactical properties, and only argues about the beliefs that can be inferred. Concretely, this is realized by arguing about the deductive closure $Cn_\Sigma(\Gamma)$ of a set of formulas Γ as seen in Def. 2

Definition 2. (Delgrande 2017) *Let Σ and P be signatures, $\mathcal{L}_\Sigma$ a language with corresponding consequence operator Cn_Σ, and $\mathcal{L}_{\Sigma\setminus P} \subseteq \mathcal{L}_\Sigma$ a sub-language, then* forgetting a signature P *in a set of formulas $\Gamma \subseteq \mathcal{L}_\Sigma$ is defined as*

$$\mathcal{F}(\Gamma, P) = Cn_\Sigma(\Gamma) \cap \mathcal{L}_{\Sigma\setminus P}.$$

By intersecting the prior knowledge $Cn_\Sigma(\Gamma)$ with the sub-language $\mathcal{L}_{\Sigma\setminus P}$ all formulas that mention any signature element $\rho \in P$ will be removed. Therefore, forgetting according to Def. 2 results in those consequences of Γ that are included in the reduced language $\mathcal{L}_{\Sigma\setminus P}$. However, since many of the logic-specific forgetting approaches do not result in a sub-language, Delgrande provides a second definition of forgetting that results in the original language instead (Def. 3). This allows comparing the results of the different forgetting approaches more easily.

Definition 3. (Delgrande 2017) *Let Σ and P be signatures and $\mathcal{L}_\Sigma$ a language with corresponding consequence operator Cn_Σ, then* forgetting a signature P in the original language $\mathcal{L}_\Sigma$ *in a set of formulas $\Gamma \subseteq \mathcal{L}_\Sigma$ is defined as*

$$\mathcal{F}_O(\Gamma, P) = Cn_\Sigma(\mathcal{F}(\Gamma, P)).$$

Thereby, forgetting in the original language $\mathcal{L}_\Sigma$ is defined as the deductive closure of $\mathcal{F}(\Gamma, P)$ with respect to Σ. Due to the syntax independent nature of Delgrande's forgetting definition, it is theoretically applicable to each logic with a well-defined consequence operator. Note that even though the posterior knowledge still consists of formulas mentioning the forgotten signature elements P, we know that they do not provide any information about P, since forgetting in the original signature results in knowledge equivalent the result of forgetting in the reduced language, due to the deductive closure Cn_Σ. This also follows from the model theoretical properties of both forgetting definitions stated in Th. 1.

Theorem 1. (Delgrande 2017) *Let $\Gamma \subseteq \mathcal{L}_\Sigma$ be a set of formulas and P a signature, then the following equations hold:*

1. $[\![\mathcal{F}(\Gamma, P)]\!]_{\Sigma\setminus P} = ([\![\Gamma]\!]_\Sigma)_{|(\Sigma\setminus P)}$
2. $[\![\mathcal{F}(\Gamma, P)]\!]_\Sigma = (([\![\Gamma]\!]_\Sigma)_{|(\Sigma\setminus P)})_{\uparrow\Sigma}$

From Th. 1, we can conclude that the models of forgetting in the original language are equal to those of forgetting in the reduced language with respect to Σ (Cor. 1).

Corollary 1. *Let $\Gamma \subseteq \mathcal{L}_\Sigma$ be a set of formulas and P a signature, then the following holds:*

$$[\![\mathcal{F}_O(\Gamma, P)]\!]_\Sigma = ([\![\mathcal{F}(\Gamma, P)]\!]_{\Sigma\setminus P})_{\uparrow\Sigma} = [\![\mathcal{F}(\Gamma, P)]\!]_\Sigma$$

In Ex. 1 below, we illustrate the relations of both forgetting definitions stated by Delgrande.

$[\![\Gamma]\!]_\Sigma$	$[\![\mathcal{F}(\Gamma, \{p\})]\!]_{\Sigma\setminus\{p\}}$	$[\![\mathcal{F}_O(\Gamma, \{p\})]\!]_\Sigma$
$\overline{p}\overline{b}\overline{f}, pb\overline{f}, \overline{p}bf$	$\overline{b}\overline{f}, b\overline{f}, bf$	$\overline{p}\overline{b}\overline{f}, p\overline{b}\overline{f}, \overline{p}b\overline{f},$ $pb\overline{f}, \overline{p}bf, pbf$

Table 1: Models of Γ, $\mathcal{F}(\Gamma, \{p\})$, and $\mathcal{F}_O(\Gamma, \{p\})$ with respect to the corresponding signatures of the languages, where $\Gamma = \{p \to b, f \to \overline{p}, f \to b, \overline{f} \to (p \vee \overline{b})\} \subseteq \mathcal{L}_\Sigma$ and $\Sigma = \{p, b, f\}$.

Example 1. *In this example, we illustrate both Delgrande's forgetting in the reduced as well as in the original language, and its effects on the model level. For this, we consider the knowledge base $\Gamma = \{p \to b, f \to \overline{p}, f \to b, \overline{f} \to (p \vee \overline{b})\} \subseteq \mathcal{L}_\Sigma$ with $\Sigma = \{p, b, f\}$, where the signature elements can be read as:*

p *– the observed animal is a penguin*,
b *– the observed animal is a bird*,
f *– the observed animal can fly*.

Thus, $\overline{f} \to (p \vee \overline{b})$ for example reads if the observed animal cannot fly, then it is a penguin or not a bird at all. *In the following, we want to forget the subsignature $\{p\} \subseteq \Sigma$. Forgetting $\{p\}$ in the reduced language $\mathcal{L}_{\Sigma\setminus\{p\}}$ results in*

$$\mathcal{F}(\Gamma, \{p\}) = Cn_\Sigma(\Gamma) \cap \mathcal{L}_{\Sigma\setminus\{p\}} = Th_\Sigma([\![\Gamma]\!]_\Sigma) \cap \mathcal{L}_{\Sigma\setminus\{p\}},$$

where $[\![\Gamma]\!]_\Sigma = \{\overline{p}\overline{b}\overline{f}, pb\overline{f}, \overline{p}bf\}$. Concretely, $\mathcal{F}(\Gamma, \{p\})$ consists of all conclusions that can be drawn from Γ and are part of the reduced language $\mathcal{L}_{\Sigma\setminus\{p\}}$, i.e. those conclusions that do not argue about penguins (p). According to Th. 1, we know that the models after forgetting $\{p\}$ from Γ correspond to the prior models $[\![\Gamma]\!]_\Sigma$ reduced to $\Sigma \setminus \{p\}$:

$$\begin{aligned} &[\![\mathcal{F}(\Gamma, \{p\})]\!]_{\Sigma\setminus\{p\}} = ([\![\Gamma]\!]_\Sigma)_{|\Sigma\setminus\{p\}} \\ &= \{\overline{p}\overline{b}\overline{f}, pb\overline{f}, \overline{p}bf\}_{|\Sigma\setminus\{p\}} = \{\overline{b}\overline{f}, b\overline{f}, bf\}. \end{aligned}$$

Thus, the posterior models after forgetting $\{p\}$ are obtain by mapping each interpretation $\dot{p}\dot{b}\dot{f}$ to $\dot{b}\dot{f}$.

If we forget $\{p\}$ in the original language $\mathcal{L}_\Sigma$ instead, we obtain

$$\begin{aligned} &\mathcal{F}_O(\Gamma, \{p\}) = Cn_\Sigma(\mathcal{F}(\Gamma, \{p\})) = Th([\![\mathcal{F}(\Gamma, \{p\})]\!]_\Sigma) \\ &= Th(([\![\mathcal{F}(\Gamma, \{p\})]\!]_{\Sigma\setminus\{p\}})_{\uparrow\Sigma}) = Th((([\![\Gamma]\!]_\Sigma)_{|\Sigma\setminus\{p\}})_{\uparrow\Sigma}). \end{aligned}$$

By means of the deductive closure of $\mathcal{F}(\Gamma, \{p\})$ with respect to Σ, the result of forgetting in the reduced language is extended by those formulas $\varphi \in \mathcal{L}_\Sigma$ in the original language that can be inferred by it. However, due to the relations of the prior models $[\![\Gamma]\!]_\Sigma$ and those after forgetting $\{p\}$ in the reduced and the original language

$$\begin{aligned} &[\![\mathcal{F}_O(\Gamma, \{p\})]\!]_\Sigma = (([\![\Gamma]\!]_\Sigma)_{|\Sigma\setminus\{p\}})_{\uparrow\Sigma} \\ &= (\{\overline{p}\overline{b}\overline{f}, pb\overline{f}, \overline{p}bf\}_{|\Sigma\setminus\{p\}})_{\uparrow\Sigma} = \{\overline{b}\overline{f}, b\overline{f}, bf\}_{\uparrow\Sigma} \\ &= \{\overline{p}\overline{b}\overline{f}, p\overline{b}\overline{f}, pb\overline{f}, \overline{p}b\overline{f}, \overline{p}bf, pbf\}, \end{aligned}$$

we see that $\mathcal{F}_O(\Gamma, \{p\})$ can only contain trivial proposition about penguins (p), since we know that if $p\dot{b}\dot{f} \in [\![\mathcal{F}_O(\Gamma, \{p\})]\!]$, then $\overline{p}\dot{b}\dot{f} \in [\![\mathcal{F}_O(\Gamma, \{p\})]\!]$ must hold as well. This way non-trivial propositions about penguins are prevented, which is why forgetting in the original language can still be considered as forgetting p. We provide an overview of the different models in Tab. 1.

Besides defining a general forgetting approach, Delgrande also states several properties of his definition, which he refers to as *right* and *desirable* (Delgrande 2017). In this work, we refer to these properties as **(DFP-1)**-**(DFP-7)** as stated in Th. 2.

Theorem 2. (Delgrande 2017) *Let $\mathcal{L}_\Sigma$ be a language over signature Σ and Cn_Σ the corresponding consequence operator, then the following relations hold for all sets of formulas $\Gamma, \Gamma' \subseteq \mathcal{L}_\Sigma$ and signatures P, P'.*

(DFP-1) $\Gamma \models \mathcal{F}(\Gamma, P)$

(DFP-2) *If* $\Gamma \models \Gamma'$*, then* $\mathcal{F}(\Gamma, P) \models \mathcal{F}(\Gamma', P)$

(DFP-3) $\mathcal{F}(\Gamma, P) = Cn_{\Sigma \setminus P}(\mathcal{F}(\Gamma, P))$

(DFP-4) *If* $P' \subseteq P$*, then* $\mathcal{F}(\Gamma, P) = \mathcal{F}(\mathcal{F}(\Gamma, P'), P)$

(DFP-5) $\mathcal{F}(\Gamma, P \cup P') = \mathcal{F}(\Gamma, P) \cap \mathcal{F}(\Gamma, P')$

(DFP-6) $\mathcal{F}(\Gamma, P \cup P') = \mathcal{F}(\mathcal{F}(\Gamma, P), P')$

(DFP-7) $\mathcal{F}(\Gamma, P) = \mathcal{F}_O(\Gamma, P) \cap \mathcal{L}_{\Sigma \setminus P}$

(DFP-1) states the monotony of forgetting, which means that it is not possible to obtain new knowledge by means of forgetting. **(DFP-2)** states that any consequence relation $\Gamma \models \Gamma'$ of prior knowledge sets is preserved after forgetting a signature P in both. **(DFP-3)** describes that forgetting always results in a deductively closed knowledge set with respect to the reduced signature. This also corresponds to Delgrande's idea of defining forgetting on the knowledge level – forgetting is applied to a deductively closed set and results in such. In **(DFP-4)**, Delgrande states that forgetting two signatures P' and P consecutively always equals the forgetting of P, if P' is included in P. Thus, forgetting a signature twice has no effect on the prior knowledge. **(DFP-5)** and **(DFP-6)** argue about iterative and simultaneous forgetting. Finally, **(DFP-7)** describes the relation between forgetting in the original and the reduced language by stating that the result of forgetting in the reduced language can always be obtained by intersecting the result of forgetting in the original language with the reduced language. Note that we changed the notation of **(DFP-7)** in order to make it more explicit. For more information on **(DFP-1)**-**(DFP-7)** we refer to (Delgrande 2017).

3.2 Marginalisation

A general framework of forgetting and its instantiation to an approach using OCFs is developed in (Beierle et al. 2019). For the purpose of this paper, we concentrate on the marginalisation, which on a cognitive level corresponds to the notion of focussing and can briefly be summarized as:

1. Focussing on relevant aspects retains our beliefs about them.
2. Focussing on relevant aspects (temporarily) changes our beliefs such that they do not contain any information about irrelevant aspects anymore.

In practice, this notion of forgetting is useful when it comes to efficient and focussed query answering by means of abstracting from irrelevant details, e.g. marginalisation is crucially used in all inference techniques for probabilistic networks. At this point, we consider the relevant aspects to be given and focus on the marginalisation (Def. 4) as a kind of forgetting as such.

Definition 4. *(Beierle et al. 2019) Let κ be an OCF over signature Σ and $\omega' \in \Omega_{\Sigma'}$ an interpretation with $\Sigma' \subseteq \Sigma$. $\kappa_{|\Sigma'}$ is called a* marginalisation *of κ to Σ' with*

$$\kappa_{|\Sigma'}(\omega') = \min\{\kappa(\omega) \mid \omega \in \Omega_\Sigma \textit{ with } \omega \models \omega'\}.$$

By marginalising an OCF to a subsignature Σ', we consider interpretations over Σ' as conjunctions and assign the corresponding rank to them.

The first notion of focussing corresponds to Lem. 1, which states that a formula over the reduced signature is believed after the marginalisation, if and only if it is also believed by the prior OCF. Thus, the beliefs that only argue about the relevant aspects Σ' are retained.

Lemma 1. *Let κ be an OCF over Σ and $\Sigma' \subseteq \Sigma$, then for each $\varphi \in \mathcal{L}_{\Sigma'}$ the following holds:*

$$\kappa_{|\Sigma'} \models \varphi \Leftrightarrow \kappa \models \varphi$$

Similarly to Delgrande's forgetting, marginalisation reduces beliefs to a subsignature. Note that Lem. 1 directly follows from (Beierle et al. 2019), where they already stated that this relations generally holds for conditional beliefs. Furthermore, Lem. 1 allows us to express the posterior beliefs analogously to Delgrande's forgetting definition (Prop. 1).

Proposition 1. *Let κ be an OCF over signature Σ and $\Sigma' \subseteq \Sigma$ a reduced signature.*

$$Bel(\kappa_{|\Sigma'}) = Bel(\kappa) \cap \mathcal{L}_{\Sigma'}$$

Proof of Prop. 1. Due to Lemma 1, we have $Bel(\kappa) \cap \mathcal{L}_{\Sigma'} = Bel(\kappa_{|\Sigma'}) \cap \mathcal{L}_{\Sigma'} = Bel(\kappa_{|\Sigma'})$ because $(Bel(\kappa_{|\Sigma'}) \subseteq \mathcal{L}_{\Sigma'})$. □

Thereby, Prop. 1 also corresponds to the second notion of focussing, due to the intersection with reduced language $\mathcal{L}_{\Sigma'}$. The above-stated relations of the prior and posterior beliefs further imply that the models of the posterior beliefs are equal to the those of the prior when reducing them to Σ' (Prop. 2). This rather technical property allows us to freely switch between the models of the marginalised and the prior OCF, which will be useful in later proofs.

Proposition 2. *Let κ be an OCF over signature Σ and $\Sigma' \subseteq \Sigma$ a subsignature. Then $[\![\kappa_{|\Sigma'}]\!] = [\![\kappa]\!]_{|\Sigma'}$ holds.*

Proof of Prop. 2. By definition,

$$[\![\kappa_{|\Sigma'}]\!] = \{\omega' \in \Omega_{\Sigma'} \mid \kappa_{|\Sigma'}(\omega') = 0\},$$

so applying Def. 4 yields

$$\begin{aligned} &[\![\kappa_{|\Sigma'}]\!] \\ =&\{\omega' \in \Omega_{\Sigma'} \mid \min\{\kappa(\omega) \mid \omega \in \Omega_\Sigma \text{ with } \omega \models \omega'\} = 0\}, \end{aligned}$$

which is the same as

$$\begin{aligned} &\{\omega' \in \Omega_{\Sigma'} \mid \exists \omega \in \Omega_\Sigma \text{ with } \omega \models \omega' \text{ and } \kappa(\omega) = 0\} \\ =&\{\omega' \in \Omega_{\Sigma'} \mid \exists \omega \in \Omega_\Sigma \text{ with } \omega \models \omega' \text{ and } \omega \in [\![\kappa]\!]\} \\ =&\{\omega' \in \Omega_{\Sigma'} \mid \exists \omega \in [\![\kappa]\!] \text{ with } \omega \models \omega'\} = [\![\kappa]\!]_{|\Sigma'}. \end{aligned}$$

□

Similar to Delgrande's idea of forgetting in the original language, we might be interested in arguing about the original signature after focussing, e.g. for reasons of comparability. Thus, we define the concept of lifting an OCF in Def. 5 below.

Definition 5. *Let κ' be an OCF over signature $\Sigma' \subseteq \Sigma$. A* lifting *of κ' to Σ, denoted by $\kappa'_{\uparrow\Sigma}$, is uniquely defined by $\kappa'_{\uparrow\Sigma}(\omega) = \kappa'(\omega_{|\Sigma'})$ for all $\omega \in \Omega_\Sigma$.*

By means of lifting an OCF κ' over signature Σ' to a signature Σ with $\Sigma' \subseteq \Sigma$, we (re-)introduce new signature elements to κ' in a way that $\kappa'_{\uparrow\Sigma}$ acts invariantly towards them. This is guaranteed by the fact that all interpretations $\omega \in \Omega_\Sigma$ that only differ in the truth value they assign to the new signature elements $\Sigma \setminus \Sigma'$ are assigned to the same rank. Analogously to Prop. 2, we show in Prop. 3 that the models of a lifted OCF are equal to the prior models when expanded to the super-signature.

Proposition 3. *Let κ' be an OCF over signature $\Sigma' \subseteq \Sigma$. Then the models of the lifted κ' are the expanded models of κ', i.e., $[\![\kappa'_{\uparrow\Sigma}]\!] = [\![\kappa']\!]_{\uparrow\Sigma}$.*

Proof of Prop. 3. By definition,

$$[\![\kappa']\!]_{\uparrow\Sigma} = \bigcup_{\omega' \in [\![\kappa']\!]} \{\omega \in \Omega_\Sigma \mid \omega \models \omega'\},$$

and hence

$$\begin{aligned}[\![\kappa']\!]_{\uparrow\Sigma} &= \{\omega \in \Omega_\Sigma \mid \exists \omega' \in [\![\kappa']\!]_{\Sigma'} \text{ with } \omega \models \omega'\} \\ &= \{\omega \in \Omega_\Sigma \mid \exists \omega' \in [\![\kappa']\!]_{\Sigma'} \text{ with } \omega_{|\Sigma'} \equiv \omega'\},\end{aligned}$$

due to $\omega \models \omega' \Leftrightarrow \omega_{|\Sigma'} = \omega'$ (Def. 1). Since we know that if there is an interpretation $\omega' \in [\![\kappa']\!]_{\Sigma'}$ that is equivalent to $\omega_{|\Sigma'}$, then $\omega_{|\Sigma'}$ is included in $[\![\kappa']\!]_{\Sigma'}$ as well, and vice-versa, this last set is the same as

$$\begin{aligned}&\{\omega \in \Omega_\Sigma \mid \omega_{|\Sigma'} \in [\![\kappa']\!]_{\Sigma'}\} = \{\omega \in \Omega_\Sigma \mid \kappa'(\omega_{|\Sigma'}) = 0\} \\ =&\{\omega \in \Omega_\Sigma \mid \kappa'_{\uparrow\Sigma}(\omega) = 0\} = [\![\kappa'_{\uparrow\Sigma}]\!],\end{aligned}$$

again by definition. □

Therefore, we also know that the beliefs after lifting are equivalent to the prior with respect to Σ, which can also be denoted as the deductive closure of the prior beliefs with respect to Σ (Prop. 4).

Proposition 4. *Let κ' be an OCF over signature $\Sigma' \subseteq \Sigma$ and $\kappa'_{\uparrow\Sigma}$ be a lifting of κ' to Σ, then the beliefs of $\kappa'_{\uparrow\Sigma}$ are given by $Bel(\kappa'_{\uparrow\Sigma}) = Cn_\Sigma(Bel(\kappa'))$.*

Proof of Prop. 4. In a straightforward way, we obtain from Prop. 3

$$\begin{aligned}&Bel(\kappa'_{\uparrow\Sigma}) = Th([\![\kappa'_{\uparrow\Sigma}]\!]) \\ =& Cn_\Sigma(\bigvee_{\omega \in [\![\kappa'_{\uparrow\Sigma}]\!]} \omega) = Cn_\Sigma(\bigvee_{\omega \in [\![\kappa']\!]_{\uparrow\Sigma}} \omega) \\ =& Cn_\Sigma(\bigvee_{\omega \in \bigcup_{\omega' \in [\![\kappa']\!]} \omega'_{\uparrow\Sigma}} \omega) = Cn_\Sigma(\bigvee_{\omega' \in [\![\kappa']\!]} (\bigvee_{\omega \in \omega'_{\uparrow\Sigma}} \omega)) \\ =& Cn_\Sigma(\bigvee_{\omega' \in [\![\kappa']\!]} \omega') = Cn_\Sigma(Cn_{\Sigma'}(\bigvee_{\omega' \in [\![\kappa']\!]} \omega')) \\ =& Cn_\Sigma(Th([\![\kappa']\!])) = Cn_\Sigma(Bel(\kappa')).\end{aligned}$$

□

Prop. 4 clearly shows that the beliefs of a marginalised OCF relate to those after lifting it to the original signature again in the same way Delgrande's forgetting in the original language relates to forgetting in the reduced language (see Def. 3).

Finally, we can show that the marginalisation generalizes Delgrande's forgetting definition to epistemic states, since both forgetting approaches result in equivalent posterior beliefs when applied to the same prior knowledge (Th. 3).

Theorem 3. *Let $\Gamma \subseteq \mathcal{L}_\Sigma$ be a set of formulas and κ an OCF over signature Σ with $Bel(\kappa) \equiv \Gamma$, then*

$$\mathcal{F}(\Gamma, P) = Bel(\kappa_{|(\Sigma \setminus P)})$$

holds for each signature P.

Proof of Th. 3. Due to Prop. 1, we have $Bel(\kappa_{|(\Sigma\setminus P)}) = Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P}$. Since $Bel(\kappa) \equiv \Gamma$, this is the same as $Cn_\Sigma(\Gamma) \cap \mathcal{L}_{\Sigma\setminus P} = \mathcal{F}(\Gamma, P)$, by definition. □

The equivalence of the prior knowledge for both approaches can be stated as $Bel(\kappa) \equiv \Gamma$, which means that the set of formulas Delgrande's forgetting is applied to must be equivalent to the prior beliefs $Bel(\kappa)$. Furthermore, note that Delgrande's forgetting definition argues about the elements that should be forgotten, while the marginalisation argues about the remaining subsignature.

In Ex. 2 below, we illustrate the marginalisation as well as a subsequently performed lifting of an OCF κ over the signature $\Sigma = \{p, b, f\}$, and show how marginalisation and lifting corresponds to Delgrande's forgetting definitions. For this we refer to the example on Delgrande's forgetting (Ex. 1).

Example 2. *In this example, we illustrate a marginalisation and a consecutively performed lifting of the OCF κ over $\Sigma = \{p, b, f\}$ (see Ex. 1) given in Tab. 2, as well as the relations to Delgrande's forgetting definitions. In the following, we want to forget the subsignature $\{p\} \subseteq \Sigma$.*

First of all, we want to note that the beliefs of κ are equivalent to the knowledge base Γ (Ex. 1), since their corresponding models are the same:

$$\begin{aligned}&Bel_\Sigma(\kappa) = Th([\![\kappa]\!]_\Sigma) = Th(\{\overline{p}\overline{b}\overline{f}, pb\overline{f}, \overline{p}bf\}) \\ =& Th([\![\Gamma]\!]_\Sigma) = Cn_\Sigma(\Gamma) \equiv \Gamma\end{aligned}$$

	κ	$\kappa_{\vert(\Sigma\setminus\{p\})}$	$(\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}$
2	$pbf, \overline{p}b\overline{f}$	-	-
1	$p\overline{b}\,\overline{f}, \overline{p}\overline{b}f, p\overline{b}f$	$\overline{b}f$	$p\overline{b}f, \overline{p}\overline{b}f$
0	$\overline{p}\overline{b}\,\overline{f}, pb\overline{f}, \overline{p}bf$	$\overline{b}\,\overline{f}, b\overline{f}, bf$	$\overline{p}\overline{b}\,\overline{f}, p\overline{b}\,\overline{f}, pb\overline{f},$ $\overline{p}b\overline{f}, \overline{p}bf, pbf$

Table 2: OCFs κ over signature $\Sigma = \{p, b, f\}$, as well as its marginalisation $\kappa_{\vert(\Sigma\setminus\{p\})}$ and the corresponding lifting $(\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}$.

Marginalising κ to $\Sigma \setminus P$ results in $\kappa_{\vert(\Sigma\setminus P)}$ as given in Tab. 2. There it can be seen that the posterior most plausible interpretation correspond to those of κ when omitting p, i.e. each interpretation $\dot{p}\dot{b}\dot{f} \in \llbracket\kappa\rrbracket$ is mapped to $\dot{b}\dot{f} \in \llbracket\kappa_{\vert(\Sigma\setminus\{p\})}\rrbracket$. This exactly corresponds to the way Delgrande's forgetting in the reduced language affects the models of the given knowledge base Γ:

$$\llbracket\kappa_{\vert(\Sigma\setminus\{p\})}\rrbracket_{\Sigma\setminus P} = \llbracket\kappa\rrbracket_{\vert(\Sigma\setminus\{p\})} = \{\overline{p}\overline{b}\,\overline{f}, pb\overline{f}, \overline{p}bf\}_{\vert(\Sigma\setminus\{p\})}$$
$$= \{\overline{b}\,\overline{f}, b\overline{f}, bf\} = \llbracket\mathcal{F}(\Gamma, \{p\})\rrbracket_{\Sigma\setminus\{p\}}$$

In conclusion, we that know the posterior beliefs of the marginalisation and the result of Delgrande's forgetting must be equal:

$$Bel(\kappa_{\vert(\Sigma\setminus\{p\})}) = Th(\llbracket\kappa_{\vert(\Sigma\setminus\{p\})}\rrbracket_{\Sigma\setminus\{p\}})$$
$$= Th(\{\overline{b}\,\overline{f}, b\overline{f}, bf\}) = Th(\llbracket\mathcal{F}(\Gamma, \{p\})\rrbracket_{\Sigma\setminus P}) = \mathcal{F}(\Gamma, \{p\})$$

When we lift the marginalised OCF $\kappa_{\vert(\Sigma\setminus\{p\})}$ back to the original signature Σ, the posterior most plausible interpretations can be obtained by mapping each interpretation $\dot{b}\dot{f} \in \llbracket\kappa_{\vert(\Sigma\setminus\{p\})}\rrbracket_{\Sigma\setminus\{p\}}$ to $\{p\dot{b}\dot{f}, \overline{p}\dot{b}\dot{f}\} \subseteq \llbracket(\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}\rrbracket_{\Sigma}$ (see Tab. 2). Just as for the marginalisation, this exactly corresponds to the way Delgrande's forgetting in the original language affects the prior models of the knowledge base Γ:

$$\llbracket(\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}\rrbracket_{\Sigma} = \{\overline{b}\,\overline{f}, b\overline{f}, bf\}_{\uparrow\Sigma}$$
$$= \{\overline{p}\overline{b}\,\overline{f}, p\overline{b}\,\overline{f}, pb\overline{f}, \overline{p}b\overline{f}, \overline{p}bf, pbf\} = \llbracket\mathcal{F}_O(\Gamma, \{p\})\rrbracket_{\Sigma}$$

Therefore, the result of Delgrande's forgetting in the original language is equal to the beliefs after marginalising and lifting κ:

$$Bel_{\Sigma}((\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}) = Th(\llbracket(\kappa_{\vert(\Sigma\setminus\{p\})})_{\uparrow\Sigma}\rrbracket_{\Sigma})$$
$$= Th(\{\overline{p}\overline{b}\,\overline{f}, p\overline{b}\,\overline{f}, pb\overline{f}, \overline{p}b\overline{f}, \overline{p}bf, pbf\})$$
$$= Th(\llbracket\mathcal{F}_O(\Gamma, \{p\})\rrbracket_{\Sigma}) = \mathcal{F}_O(\Gamma, \{p\})$$

From the equivalence stated in Th. 3, we know that all relations of the logic-specific forgetting approaches and Delgrande's general approach that can be traced back to the equivalence of the results must hold for the marginalisation as well. In the following, we exemplarily state this for Boole's atom forgetting in propositional (Def. 6), of which we know that it can also be described by means of $\mathcal{F}$ (Th. 4).

Definition 6. (Boole 1854) *Let $\varphi \in \mathcal{L}_{\Sigma}$ be a formula and $\rho \in \mathcal{L}_{\Sigma}$ be an atom.* Forgetting *ρ in φ is then defined as*

$$forget(\varphi, \rho) = \varphi[\rho/\top] \vee \varphi[\rho/\bot],$$

where $\varphi[\rho/\top]$ denotes the substitution of ρ by $\top$, and $\varphi[\rho/\bot]$ the substitution by $\bot$.

Theorem 4. (Delgrande 2017) *Let $\mathcal{L}_{\Sigma}$ be the language in propositional logic with signature Σ and let $\rho \in \Sigma$ be an atom.*

$$forget(\varphi, \rho) \equiv \mathcal{F}(\varphi, \{\rho\})$$

From Th. 3 and Th. 4, we can directly conclude that Boole's forgetting definition can also be realized by means of a marginalisation (Cor. 2).

Corollary 2. *Let κ be an OCF over signature Σ and $\varphi \in \mathcal{L}_{\Sigma}$ a formula with $Bel(\kappa) \equiv \varphi$, then*

$$forget(\varphi, \rho) \equiv Bel(\kappa_{\vert\Sigma\setminus\{\rho\}})$$

holds for each atom $\rho \in \Sigma$.

4 Postulates for Forgetting Signatures in Epistemic States

In (Delgrande 2017), Delgrande argues that the properties **(DFP-1)**-**(DFP-7)** (Th. 2) of his forgetting definition are *right* and *desirable* for describing the general notions of forgetting. Since we already proved that his definition can be generalised to epistemic states by means of the marginalisation, we also present an extended and generalised form of **(DFP-1)**-**(DFP-7)**, namely **(DFPes-1)**$_{\Sigma}$-**(DFPes-6)**$_{\Sigma}$, and show that the marginalisation satisfies all of them. For this, let Ψ, Φ be epistemic states, P, P', P_1, P_2 signatures, and $\circ_f^{\Sigma}$ an arbitrary operator that maps an epistemic state together with a signature to a new epistemic state:

(DFPes-1)$_{\Sigma}$ $Bel(\Psi) \models Bel(\Psi \circ_f^{\Sigma} P)$

(DFPes-2)$_{\Sigma}$ If $Bel(\Psi) \models Bel(\Phi)$, then $Bel(\Psi \circ_f^{\Sigma} P) \models Bel(\Phi \circ_f^{\Sigma} P)$

(DFPes-3)$_{\Sigma}$ If $P' \subseteq P$, then $Bel((\Psi \circ_f^{\Sigma} P') \circ_f^{\Sigma} P) \equiv Bel(\Psi \circ_f^{\Sigma} P)$

(DFPes-4)$_{\Sigma}$ $Bel(\Psi \circ_f^{\Sigma} (P_1 \cup P_2)) \equiv Bel(\Psi \circ_f^{\Sigma} P_1) \cap Bel(\Psi \circ_f^{\Sigma} P_2)$

(DFPes-5)$_{\Sigma}$ $Bel(\Psi \circ_f^{\Sigma} (P_1 \cup P_2)) \equiv Bel((\Psi \circ_f^{\Sigma} P_1) \circ_f^{\Sigma} P_2)$

(DFPes-6)$_{\Sigma}$ $Bel(\Psi \circ_f^{\Sigma} P) \equiv Bel((\Psi \circ_f^{\Sigma} P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}$

For a detailed explanation of the above-stated postulates **(DFPes-1)**$_{\Sigma}$-**(DFPes-6)**$_{\Sigma}$, we refer to the explanations of the postulates **(DFP-1)**-**(DFP-7)** as originally stated by Delgrande. However, there are a few points we want to emphasise in particular. First, since the beliefs of an epistemic state are deductively closed by definition, it is not necessary to maintain **(DFP-3)**. Notice that due to omitting **(DFP-3)** the postulates **(DFP-4)**-**(DFP-7)** correspond to **(DFPes-3)**$_{\Sigma}$-**(DFPes-6)**$_{\Sigma}$. Furthermore, we expressed the forgetting in the original signature $\mathcal{F}_O(\Gamma, P)$ in **(DFP-7)** as the beliefs after forgetting P and lifting the posterior epistemic state back to the original signature. The models of $\mathcal{F}_O(\Gamma, P)$ are equal to the models of forgetting P in Γ in the reduced signature lifted back to the original signature, i.e. $\llbracket\mathcal{F}(\Gamma, P)\rrbracket_{\uparrow\Sigma}$ (Cor. 1). When we consider the models of $Bel((\Psi \circ_f^{\Sigma} P)_{\uparrow\Sigma})$, i.e. $\llbracket\Psi \circ_f^{\Sigma} P\rrbracket_{\uparrow\Sigma}$, we see that this also describes the models after forgetting P lifted back to the original signature.

Therefore, **(DFPes-6)**$_\Sigma$ exactly matches the property originally stated by **(DFP-7)**. In the following, we refer to those operators satisfying **(DFPes-1)**$_\Sigma$-**(DFPes-6)**$_\Sigma$ as signature forgetting operators.

Next, we show in Th. 5 that the marginalisation satisfies **(DFPes-1)**$_\Sigma$-**(DFPes-6)**$_\Sigma$, and therefore not only yields results equivalent to those of Delgrande's forgetting definition, but also corresponds to the notions of forgetting stated by Delgrande by means of **(DFP-1)**-**(DFP-7)**.

Note that there exist forgetting approaches that yield results semantically equivalent to those of Delgrande's approach, but do not satisfy **(DFP-1)**-**(DFP-7)**. An example is Boole's atom forgetting (Def. 6), which violates **(DFP-3)**.

Theorem 5. *Let κ be an OCF over signature Σ and P a signature. The marginalisation $\kappa_{|(\Sigma\setminus P)}$ to a subsignature $(\Sigma \setminus P) \subseteq \Sigma$ satisfies **(DFPes-1)**$_\Sigma$-**(DFPes-6)**$_\Sigma$.*

Proof of Th. 5. In the following, we assume the epistemic states Ψ and Φ to be OCFs, since the marginalisation is specifically defined over OCFs, denoted as κ and κ', and further denote the marginalisation $\kappa_{|\Sigma\setminus P}$ as $\kappa \circ_f^{\Sigma,m} P$.

For **(DFPes-1)**$_\Sigma$, we need to show $Bel(\kappa) \models Bel(\kappa \circ_f^{\Sigma,m} P)$, which means $Bel(\kappa) \models Bel(\kappa_{|(\Sigma\setminus P)})$. This holds due to Lem. 1. For **(DFPes-2)**$_\Sigma$, we presuppose $Bel(\kappa) \models Bel(\kappa')$. Then also $Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P} \models Bel(\kappa') \cap \mathcal{L}_{\Sigma\setminus P}$ which is equivalent to $Bel(\kappa_{|(\Sigma\setminus P)}) \models Bel(\kappa'_{|(\Sigma\setminus P)})$ because of Prop. 1, and hence by definition, $Bel(\kappa \circ_f^{\Sigma,m} P) \models Bel(\kappa' \circ_f^{\Sigma,m} P)$.

Regarding **(DFPes-3)**$_\Sigma$, we have the following equalities due to Prop. 2, and because of $P' \subseteq P$:

$$\begin{aligned}
&Bel((\kappa \circ_f^{\Sigma,m} P') \circ_f^{\Sigma,m} P)\\
&= Bel(\kappa_{\Sigma\setminus P'} \circ_f^{\Sigma,m} P) = Bel((\kappa_{|\Sigma\setminus P'})_{|(\Sigma\setminus P')\setminus P})\\
&= Bel((\kappa_{|\Sigma\setminus P'})_{|(\Sigma\setminus(P'\cup P))}) = Th([\![(\kappa_{|\Sigma\setminus P'})_{|(\Sigma\setminus(P'\cup P))}]\!])\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus(P'\cup P)} \mid [\![(\kappa_{|\Sigma\setminus P'})_{|(\Sigma\setminus(P'\cup P))}]\!] \models \varphi\}\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus(P'\cup P)} \mid [\![\kappa_{|\Sigma\setminus P'}]\!]_{|\Sigma\setminus(P'\cup P)} \models \varphi\}\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus(P'\cup P)} \mid ([\![\kappa]\!]_{|\Sigma\setminus P'})_{|\Sigma\setminus(P'\cup P)} \models \varphi\}\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus(P'\cup P)} \mid [\![\kappa]\!]_{|\Sigma\setminus(P'\cup P)} \models \varphi\}\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus P} \mid [\![\kappa]\!]_{|\Sigma\setminus P} \models \varphi\}\\
&= \{\varphi \in \mathcal{L}_{\Sigma\setminus P} \mid [\![\kappa_{|\Sigma\setminus P}]\!] \models \varphi\}\\
&= Th([\![\kappa_{|\Sigma\setminus P}]\!]) = Bel(\kappa_{|\Sigma\setminus P}) = Bel(\kappa \circ_f^{\Sigma,m} P).
\end{aligned}$$

The proof of **(DFPes-4)**$_\Sigma$ is mainly based on Prop. 1, here we compute

$$\begin{aligned}
&Bel(\kappa \circ_f^{\Sigma,m} (P_1 \cup P_2)) = Bel(\kappa_{|\Sigma\setminus(P_1\cup P_2)})\\
&= Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus(P_1\cup P_2)} = Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P_1} \cap \mathcal{L}_{\Sigma\setminus P_2}\\
&= Bel(\kappa) \cap Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P_1} \cap \mathcal{L}_{\Sigma\setminus P_2}\\
&= (Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P_1}) \cap (Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P_2})\\
&= Bel(\kappa_{|\Sigma\setminus P_1}) \cap Bel(\kappa_{|\Sigma\setminus P_2})\\
&= Bel(\kappa \circ_f^{\Sigma,m} P_1) \cap Bel(\kappa \circ_f^{\Sigma,m} P_2).
\end{aligned}$$

Similarly for **(DFPes-5)**$_\Sigma$, we have

$$\begin{aligned}
&Bel(\kappa \circ_f^{\Sigma,m} P_1 \cup P_2) = Bel(\kappa_{|\Sigma\setminus(P_1\cup P_2)})\\
&= Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus(P_1\cup P_2)} = Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus(P_1\cup(P_1\cup P_2))}\\
&= Bel(\kappa) \cap (\mathcal{L}_{\Sigma\setminus P_1} \cap \mathcal{L}_{\Sigma\setminus(P_1\cup P_2)})\\
&= (Bel(\kappa) \cap \mathcal{L}_{\Sigma\setminus P_1}) \cap \mathcal{L}_{\Sigma\setminus(P_1\cup P_2)}\\
&= Bel(\kappa_{|\Sigma\setminus P_1}) \cap \mathcal{L}_{\Sigma\setminus(P_1\cup P_2)} = Bel(\kappa_{|\Sigma\setminus P_1}) \cap \mathcal{L}_{(\Sigma\setminus P_1)\setminus P_2}\\
&= Bel((\kappa_{|\Sigma\setminus P_1})_{|(\Sigma\setminus P_1)\setminus P_2}) = Bel((\kappa \circ_f^{\Sigma,m} P_1)_{|(\Sigma\setminus P_1)\setminus P_2})\\
&= Bel((\kappa \circ_f^{\Sigma,m} P_1) \circ_f^{\Sigma,m} P_2).
\end{aligned}$$

Finally, Prop. 4 is used for proving **(DFPes-6)**$_\Sigma$:

$$\begin{aligned}
&Bel((\kappa \circ_f^{\Sigma,m} P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P} = Bel((\kappa_{|\Sigma\setminus P})_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}\\
&= Cn_\Sigma(Bel(\kappa_{|\Sigma\setminus P})) \cap \mathcal{L}_{\Sigma\setminus P} = Bel(\kappa_{|\Sigma\setminus P}) \cap \mathcal{L}_{\Sigma\setminus P}\\
&= Bel(\kappa_{|\Sigma\setminus P}) = Bel(\kappa \circ_f^{\Sigma,m} P).
\end{aligned}$$

□

From Th. 5 above, we can also conclude that the marginalisation forms the signature forgetting operator that only induces minimal changes to the prior beliefs.

Proposition 5. *Let κ be an OCF over signature Σ, $P \subseteq \Sigma$ a subsignature, and $\circ_f^\Sigma$ an operator satisfying **(DFPes-1)**$_\Sigma$-**(DFPes-6)**$_\Sigma$, where $\kappa \circ_f^\Sigma P$ is an OCF over the reduced signature $\Sigma \setminus P$, then the following relation holds:*

$$Bel(\kappa_{|\Sigma\setminus P}) \models Bel(\kappa \circ_f^\Sigma P)$$

Proof of Prop. 5. Because of **(DFPes-1)**$_\Sigma$ and **(DFPes-6)**$_\Sigma$, we have $Bel(\kappa) \models Bel(\kappa \circ_f^\Sigma P) = Bel((\kappa \circ_f^\Sigma P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}$, hence $\kappa \models \varphi$ for all $\varphi \in Bel((\kappa \circ_f^\Sigma P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}$. But then also, due to Lem. 1, $\kappa_{|\Sigma\setminus P} \models \varphi$ for all $\varphi \in Bel((\kappa \circ_f^\Sigma P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}$, which means $Bel(\kappa_{|\Sigma\setminus P}) \models Bel((\kappa \circ_f^\Sigma P)_{\uparrow\Sigma}) \cap \mathcal{L}_{\Sigma\setminus P}$, and therefore again due to **(DFPes-6)**$_\Sigma$, $Bel(\kappa_{|\Sigma\setminus P}) \models Bel(\kappa \circ_f^\Sigma P)$, which was to be shown. □

Thus, we know that any signature forgetting operator other than the marginalisation must induce further belief changes for some epistemic states and signatures. Such signature forgetting operators could for example depend on some model prioritisation in addition to the epistemic state and the signature itself.

5 Forgetting Signatures vs. Forgetting Formulas – A Triviality Result

In the following, we want to discuss **(DFP-1)**-**(DFP-7)** displaying the right properties to describe the general notions forgetting. In our opinion, these properties might display the *right* properties when assuming forgetting as a reduction of the language, or as forgetting signature elements, respectively. Delgrande also comments on this, and argues that other (belief change) operators that could be considered as some kind of forgetting, e.g. contraction, should simply not be considered as forgetting. We think that this view on the concept of forgetting as such is debatable, since there exist multiple intuitively and cognitively different kinds of forgetting (Beierle et al. 2019) from which Delgrande's approach, which corresponds to the notion of focussing (Th. 3), only forms one. Therefore, it is still to be investigated whether **(DFP-1)**-**(DFP-7)** also states the right properties for other kinds of forgetting.

Following the overview of cognitively different kinds of forgetting presented in (Beierle et al. 2019), it can be seen that the concept of focussing, i.e. the marginalisation, forms the only kind of forgetting that describes forgetting with respect to signatures. Thus, in order to investigate Delgrande's forgetting properties for those kinds of forgetting that argue about formulas instead, we have to generalise and extend **(DFP-1)**-**(DFP-7)** such that they not only argue about arbitrary epistemic states and operators, but also about formulas. We refer to them as **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$. For this, let Ψ, Φ be epistemic states, $\varphi, \psi \in \mathcal{L}$ formulas, and $\circ_f^\mathcal{L}$ an arbitrary belief change operator:

(DFPes-1)$_\mathcal{L}$ $Bel(\Psi) \models Bel(\Psi \circ_f^\mathcal{L} \varphi)$

(DFPes-2)$_\mathcal{L}$ If $Bel(\Psi) \models Bel(\Phi)$, then $Bel(\Psi \circ_f^\mathcal{L} \varphi) \models Bel(\Phi \circ_f^\mathcal{L} \varphi)$

(DFPes-3)$_\mathcal{L}$ If $\varphi \models \psi$, then $Bel(\Psi \circ_f^\mathcal{L} \varphi) \equiv Bel((\Psi \circ_f^\mathcal{L} \psi) \circ_f^\mathcal{L} \varphi)$

(DFPes-4)$_\mathcal{L}$ $Bel(\Psi \circ_f^\mathcal{L} (\varphi \vee \psi)) \equiv Bel(\Psi \circ_f^\mathcal{L} \varphi) \cap Bel(\Psi \circ_f^\mathcal{L} \psi)$

(DFPes-5)$_\mathcal{L}$ $Bel(\Psi \circ_f^\mathcal{L} (\varphi \vee \psi)) \equiv Bel((\Psi \circ_f^\mathcal{L} \varphi) \circ_f^\mathcal{L} \psi)$

(DFPes-6)$_\mathcal{L}$ If $\varphi \not\equiv \top$, then $Bel(\Psi \circ_f^\mathcal{L} \varphi) \not\models \varphi$

While the extension to **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$ works almost analogously to the extension to **(DFPes-1)**$_\Sigma$-**(DFPes-6)**$_\Sigma$, there exist some crucial differences, which we will address in the following. In **(DFP-4)**, Delgrande argues about forgetting signature P, P' for which we assume that P' is fully included in P. In order to extend and generalise this property, we have to examine how this notion can be described with respect to formulas. We found it most accurate to generalise this relation of the information we would like to forget by means of the specificity of formulas, i.e. $\varphi \models \psi$. Thereby, we say that the knowledge described by ψ is fully included in that of φ, if and only if ψ can be inferred from φ. More formally, this can be stated by means of the deductive closures of φ and ψ, i.e. $\varphi \models \psi \Leftrightarrow Cn(\psi) \subseteq Cn(\varphi)$.

In **(DFP-5)** and **(DFP-6)**, Delgrande argues about forgetting two signatures P, P' at once, which is described as forgetting $P \cup P'$. On a more intuitive level this can be viewed as only forgetting a single piece of information that consist of both the information we actual like to forget. When arguing about formulas instead of signatures, this can be expressed by means of a disjunction $\varphi \vee \psi$, where φ and ψ are the two formulas we actually want to forget. Even though it might seem more appropriate to describe this idea by means of a conjunction $\varphi \wedge \psi$, it is not sufficient to forget the conjunction in order to forget both φ and ψ, since it is generally sufficient to forget one of the formulas in order to forget the conjunction as well. Thus, describing the unification of two pieces of information by means of a disjunction guarantees that both formulas can no longer be inferred after forgetting.

Just as for the postulates for forgetting signatures, we omit **(DFP-3)**, since a belief set is already deductively closed by definition. Furthermore, we omit **(DFP-7)** since it argues about the relation of forgetting in the reduced and in the original language, which is not applicable in case of forgetting formulas. Instead, we introduce an additional postulate **(DFPes-6)**$_\mathcal{L}$ that explicitly states the success of the forgetting operator, i.e. after forgetting a non-tautologous formula φ, we are no longer able to infer φ.

When extending **(DFP-1)**-**(DFP-7)** to forgetting formulas, Delgrande's idea that forgetting should be performed on the knowledge level, and therefore should be independent of the syntactic structure of the given knowledge also extends to the knowledge we want to forget. Thus, we show in Th. 6 the syntax independence implied by **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$.

Theorem 6 (Syntax Independence)**.** *Let Ψ be an epistemic state and $\circ_f^\mathcal{L}$ a belief change operator satisfying **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$. Further, let $\varphi, \psi \in \mathcal{L}$ be formulas, then the following holds:*

$$\textit{If } \varphi \equiv \psi, \textit{ then } Bel(\Psi \circ_f^\mathcal{L} \varphi) \equiv Bel(\Psi \circ_f^\mathcal{L} \psi).$$

Proof of Th. 6. From $\varphi \equiv \psi$, we obtain with **(DFPes-3)**$_\mathcal{L}$ and **(DFPes-5)**$_\mathcal{L}$:

$$Bel(\Psi \circ_f^\mathcal{L} \varphi) = Bel((\Psi \circ_f^\mathcal{L} \psi) \circ_f^\mathcal{L} \varphi) = Bel(\Psi \circ_f^\mathcal{L} \psi \vee \varphi)$$

and

$$Bel(\Psi \circ_f^\mathcal{L} \psi) = Bel((\Psi \circ_f^\mathcal{L} \varphi) \circ_f^\mathcal{L} \psi) = Bel(\Psi \circ_f^\mathcal{L} \varphi \vee \psi).$$

Therefore,

$$\begin{aligned} Bel(\Psi \circ_f^\mathcal{L} \varphi) &= Bel(\Psi \circ_f^\mathcal{L} \psi) \cap Bel(\Psi \circ_f^\mathcal{L} \varphi) \\ &= Bel(\Psi \circ_f^\mathcal{L} \psi), \end{aligned}$$

due to **(DFPes-4)**$_\mathcal{L}$. □

Next, we show that there cannot exist any non-trivial belief change operator satisfying **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$. For this, we first show that **(DFPes-3)**$_\mathcal{L}$ together with **(DFPes-5)**$_\mathcal{L}$ imply that the forgetting of any conjunction $\varphi \wedge \psi$ must result in beliefs equivalent to just forgetting φ or ψ (Prop. 6).

Proposition 6. *Let Ψ be an epistemic state and $\circ_f^\mathcal{L}$ a belief change operator satisfying **(DFPes-1)**$_\mathcal{L}$-**(DFPes-6)**$_\mathcal{L}$, then*

$$Bel(\Psi \circ_f^\mathcal{L} \varphi) \equiv Bel(\Psi \circ_f^\mathcal{L} \varphi \wedge \psi) \equiv Bel(\Psi \circ_f^\mathcal{L} \psi)$$

holds for all formulas $\varphi, \psi \in \mathcal{L}$.

Proof of (Prop. 6). Using **(DFPes-3)**$_\mathcal{L}$, **(DFPes-5)**$_\mathcal{L}$, and Th. 6, we compute

$$\begin{aligned} Bel(\Psi \circ_f^\mathcal{L} \varphi \wedge \psi) &= Bel((\Psi \circ_f^\mathcal{L} \varphi) \circ_f^\mathcal{L} \varphi \wedge \psi) \\ = Bel(\Psi \circ_f^\mathcal{L} \varphi \vee (\varphi \wedge \psi)) &= Bel(\Psi \circ_f^\mathcal{L} \varphi). \end{aligned}$$

This holds for ψ analogously. Thus, we can conclude $Bel(\Psi \circ_f^\mathcal{L} \varphi) \equiv Bel(\Psi \circ_f^\mathcal{L} \varphi \wedge \psi) \equiv Bel(\Psi \circ_f^\mathcal{L} \psi)$. □

From Prop. 6 we can especially conclude that forgetting according to **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$ must be independent of the formula we actually like to forget (Cor. 3).

Corollary 3. *Let Ψ be an epistemic state and $\circ_f^{\mathcal{L}}$ a belief change operator satisfying **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$, then*

$$Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv Bel(\Psi \circ_f^{\mathcal{L}} \psi)$$

holds for all formulas $\varphi, \psi \in \mathcal{L}$.

Therefore, we know that a forgetting operator satisfying **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$ must always forget all prior beliefs except for tautologies (Th. 7).

Theorem 7 (Triviality Result)**.** *Let Ψ be an epistemic state. A belief change operator $\circ_f^{\mathcal{L}}$ satisfies **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$, if and only if $Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv \top$ holds for each $\varphi \in \mathcal{L}$.*

Proof of Th. 7. We prove Th. 7 in two steps. First, we show that if a belief change operator satisfies **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$, then it must always result in posterior beliefs $Bel(\Psi \circ_f^{\mathcal{L}} \varphi)$ equivalent to $\top$. Second, we show that each belief change operator $\circ_f^{\mathcal{L}}$ with $Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv \top$ for each $\varphi \in \mathcal{L}$ satisfies **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$. We refer to these two steps as ($\Rightarrow$) and ($\Leftarrow$). Note that we assume all formulas $\varphi, \psi \in \mathcal{L}$ to be non-tautologous.

Case ($\Rightarrow$): From Cor. 3, we know that applying $\circ_f^{\mathcal{L}}$ to an epistemic state Ψ must result in equivalent beliefs for all formulas $\varphi, \psi \in \mathcal{L}$. From **(DFPes-6)**$_{\mathcal{L}}$ we know that after forgetting a formula φ, we are no longer able to infer φ. Since the posterior beliefs are equivalent for all formulas, we can conclude that after applying $\circ_f^{\mathcal{L}}$ to Ψ, we are not able to infer any formula, but tautologies.

$$\begin{array}{ll} Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv Bel(\Psi \circ_f^{\mathcal{L}} \psi), & \\ \quad \text{for all } \varphi, \psi \in \mathcal{L} & \text{(Cor. 3)} \\ \Rightarrow Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \not\models \varphi, \psi, \text{ for all } \varphi, \psi \in \mathcal{L} & \textbf{(DFPes-1)}_{\mathcal{L}} \\ \Leftrightarrow Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv \top, \text{ for all } \varphi \in \mathcal{L} & \end{array}$$

Case ($\Leftarrow$): Let Ψ and Φ be epistemic states and $\varphi, \psi \in \mathcal{L}$ be non-tautologous formulas, and $\circ_f^{\mathcal{L}}$ a belief change operator with $Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv \top$ for all epistemic states Ψ and formulas φ. The fact that $\circ_f^{\mathcal{L}}$ satisfies **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$ directly concludes from the assumption $Bel(\Psi \circ_f^{\mathcal{L}} \varphi) \equiv \top$, for all $\varphi \in \mathcal{L}$.

We showed that both cases ($\Rightarrow$) and ($\Leftarrow$) hold, and therefore proved the triviality result stated in Th. 7. □

6 Conclusion

We discussed two of the existing approaches towards a general forgetting framework. The first approach was that of Delgrande (2017) in which he gives a general forgetting definition that argues about forgetting on the knowledge level, and is capable of representing several of the hitherto existing logic-specific forgetting approaches, such as Boole's atom forgetting in propositional logic (Boole 1854). The second approach was that of Beierle et al. (2019). In contrast to Delgrande's approach, Beierle et al. define several cognitively different kinds of forgetting in a general OCF framework, which is generally more expressive than just arguing about knowledge sets. Thereby, we concretely focussed on the marginalisation or the concept of focussing as one kind of forgetting, respectively, which is of importance when it comes to efficient and focussed query answering. We showed that the marginalisation generalizes Delgrande's forgetting definition to epistemic states by resulting in equivalent posterior beliefs, as well as holding the same properties, which Delgrande referred to as *right* and *desirable*. Furthermore, this implies that the relations Delgrande elaborated between his and the logic-specific approaches also hold for the marginalisation. We exemplarily showed this by means of Boole's atom forgetting in propositional logic. We think that **(DFP-1)**-**(DFP-7)**, or **(DFPes-1)**$_{\Sigma}$-**(DFPes-6)**$_{\Sigma}$ respectively, describe properties that are *right* and *desirable* as long as we consider the forgetting of signature elements. However, we showed that these properties are not suitable for postulating properties for any kind of forgetting formulas, since generalizing these properties to formulas **(DFPes-1)**$_{\mathcal{L}}$-**(DFPes-6)**$_{\mathcal{L}}$ implies the triviality result stated in Th. 7.

In principle, we agree with Delgrande insofar that belief change operators like contraction are essentially different from the notion of forgetting as it is implemented by Delgrande's approach. However, we argue that Delgrande's approach and in general, approaches based on variable elimination, are too narrow to cover cognitive forgetting in its full generality. As our triviality result shows, Delgrande's postulates seem to be unsuitable for describing the forgetting of formulas. Nevertheless, as the works of Beierle et al. (2019) show, very different kinds of forgetting are realizable in a common framework, distinguishable by different properties. So, as part of our future work, we pursue the research question which of Delgrande's postulates (which all seem very rational at first sight) need to be modified or omitted to make the idea of forgetting by variable elimination reconcilable to other forms of forgetting and how Delgrande's forgetting definition itself could be amended to satisfy the adapted postulates.

References

Baral, C., and Zhang, Y. 2005. Knowledge updates: Semantics and complexity issues. *Artificial Intelligence* 164(1-2):209–243.

Beierle, C.; Kern-Isberner, G.; Sauerwald, K.; Bock, T.; and Ragni, M. 2019. Towards a general framework for kinds of forgetting in common-sense belief management. *KI-Künstliche Intelligenz* 33(1):57–68.

Boole, G. 1854. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities.* Dover Publications.

Delgrande, J. P. 2017. A knowledge level account of forgetting. *Journal of Artificial Intelligence Research* 60:1165–1213.

Ellwart, T.; Ulfert, A.-S.; Antoni, C. H.; Becker, J.; Frings, C.; Göbel, K.; Hertel, G.; Kluge, A.; Meeßen, S. M.;

Niessen, C.; et al. 2019. Intentional forgetting in socio-digital work systems: system characteristics and user-related psychological consequences on emotion, cognition, and behavior. *AIS Transactions on Enterprise Systems* 4(1).

Kluge, A.; Schüffler, A. S.; Thim, C.; Haase, J.; and Gronau, N. 2019. Investigating unlearning and forgetting in organizations. *The Learning Organization*.

Lin, F., and Reiter, R. 1994. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, 154–159.

Spohn, W. 1988. Ordinal conditional functions: A dynamic theory of epistemic states. In *Causation in decision, belief change, and statistics*. Springer. 105–134.

Wong, K.-S. 2009. *Forgetting in logic programs*. Ph.D. Dissertation, Ph. D. thesis, The University of New South Wales.

Zhang, Y., and Foo, N. Y. 2006. Solving logic program conflict through strong and weak forgettings. *Artificial Intelligence* 170(8-9):739–778.

On the Learnability of Possibilistic Theories

Cosimo Persia and **Ana Ozaki**[*]
University of Bergen
{cosimo.persia, ana.ozaki}@uib.no

Abstract

We investigate learnability of possibilistic theories from entailments in light of Angluin's exact learning model. We consider cases in which only membership, only equivalence, and both kinds of queries can be posed by the learner. We then show that, for a large class of problems, polynomial time learnability results for classical logic can be transferred to the respective possibilistic extension. In particular, it follows from our results that the possibilistic extension of propositional Horn theories is exactly learnable in polynomial time. As polynomial time learnability in the exact model is transferable to the classical probably approximately correct model extended with membership queries, we establish such results in this model. [†]

1 Introduction

Uncertainty is found in many phases of learning, such as model selection and processing noisy, imperfect, incomplete or limited data. In most cases, knowledge-based systems are constrained to live under conditions of ignorance. There are different approaches to deal with uncertainty (Parsons and Hunter 1998). A well-studied formalism for dealing with it is *possibilistic logic* (Dubois *et al.* 1994; Lang 2000). It admits a graded notion of possibility and makes a clear distinction between the concepts of truth and belief (Dubois and Prade 2001). Uncertainty of formulas in possibilistic logic is not subject to the complement rule as in probability theory (Agarwal and Nayal 2015; Dubois and Prade 1993). Indeed, complementary formulas may be considered fully possible, meaning complete ignorance about their truth value.

Example 1. Consider a doctor who has to diagnose a patient that suffers from extreme fatigue. A doctor can consider blood-related conditions: iron deficiency, iron overload, and vitamin B12 deficiency. Within possibility theory, one can model cases of complete uncertainty. Both iron deficiency and iron overload, which are two mutually exclusive conditions, can be considered fully possible. Consider that vitamin B12 deficiency is considered to be less possible, e.g. associated with the value $1/3$, based on some information provided by the patient. In probability theory, complete ignorance of the first two conditions would make us assign probability $1/3$ to every condition (Laplace criterion). Thus, it would not model the knowledge about vitamin B12 deficiency and the ignorance about iron deficiency and iron overload. ◁

Although possibilistic logic has been extensively studied (Dubois and Prade 2015), there are not many works that investigate learnability of possibilistic theories. We partially cover this gap by studying whether possibilistic theories are learnable in Angluin's exact learning model (Angluin 1988). In this model, a learner interacts with a teacher to exactly identify an abstract target concept. One can see the doctor, in Example 1, as a learner who inquires the patient (playing the role of a teacher) to identify a disease by posing queries.

The most studied communication protocol in this model contains questions of two kinds, called *membership* and *equivalence* queries. Membership queries allow the learner to know whether a certain statement holds (e.g. "Anaemia in family history?"). Equivalence queries allow the learner to check whether a hypothesis (e.g. a diagnose) is correct and, if not, to fix it using a counterexample. In our example, the patient may not be able to provide a counterexample but new symptoms or reactions can reveal that the hypothesis is not correct. To the best of our knowledge, this is the first work where learnability of possibilistic theories is investigated in Angluin's model. We consider cases in which only membership, only equivalence, and both kinds of queries can be posed by the learner. We also study whether known polynomial time exact learning results for classical logic can be transferred to possibilistic settings.

Our main result is that, for a large class of problems, polynomial time learnability (with both types of queries) can be transferred from classical logic to the respective possibilistic extension (Theorem 18). If only membership queries are allowed (and the maximal precision of valuations in the target is fixed) then polynomial time learnability of a classical logic can also be transferred to the possibilistic extension. We leave open the case in which only equivalence queries can be asked. With our main result, we establish, e.g., that the possibilistic extension of propositional Horn (Angluin *et al.* 1992; Frazier and Pitt 1993; Hermo and Ozaki 2020) and fragments of first-order Horn (Arimura 1997; Reddy and Tadepalli 1998; Konev *et al.* 2018) are exactly learnable in polynomial time (with both kinds of queries). We also establish polynomial time learnability results in the probably approximately cor-

[*] Contact Author.

[†] This paper was published at IJCAI20 (https://static.ijcai.org/2020-accepted_papers.html). Available at https://arxiv.org/abs/2005.03157.

rect (PAC) (Valiant 1984) model extended with membership queries.

Related Work. Among the works that combine learning and possibilistic logic, we can find results on learning possibilistic logic theories from default rules within the PAC learning model (Kuzelka *et al.* 2016). Possibilistic logic has been used to reason with default rules (Benferhat *et al.* 1992) to select the most plausible rule and in inductive logic programming to handle exceptions (Serrurier and Prade 2007). In statistical relational learning, possibilistic logic has been used as a formal encoding of statistical regularities found in relational data (Kuzelka *et al.* 2017). Possibilistic formulas can encode Markov logic networks (Kuzelka *et al.* 2015). Formal concept analysis has been applied to generate attribute implications with a degree of certainty (Djouadi *et al.* 2010). We also point out an extension of version space learning that deals with examples associated with possibility degrees (Prade and Serrurier 2008).

In Section 2, we present basic definitions. In Section 3, we investigate whether possibilistic logic theories can be learned and, in Section 4, we show transferability of polynomial time learnability results.

2 Basics

In the following, we provide relevant notions of possibilistic logic and learning theory used in the paper.

2.1 Possibilistic Theories

Let L be a propositional or a first-order (FO) language (restricted to well-formed formulas without free variables) with the semantics of classical FO logic. We say that $\varphi \in L$ is *satisfiable* if there is an interpretation $\mathcal{I}$ such that φ is satisfied in $\mathcal{I}$. Moreover, φ is *falsifiable* if its negation $\neg\varphi$ is satisfiable. An *FO knowledge base* (FO KB) is a finite set of FO formulas. An FO KB is *non-trivial* if it is satisfiable and falsifiable. The *possibilistic extension* of an FO language L is defined as follows. A *possibilistic formula* is a pair (φ, α), where $\varphi \in L$ and α is a real number (with finite precision) in the interval $(0, 1]$, called the *valuation* of φ. A *possibilistic KB* (or a possibilistic theory) is a finite set $\mathcal{K}$ of possibilistic formulas. Given a set Ω of interpretations for L, a *possibility distribution* π is a function from Ω to the interval $[0, 1]$. The *possibility* and *necessity measures*, Π and N, are functions (induced by π) from L to $[0, 1]$, defined respectively as

$$\Pi(\varphi) = \sup\{\pi(\mathcal{I}) \mid \mathcal{I} \in \Omega,\ \mathcal{I} \models \varphi\}$$

$$N(\varphi) = 1 - \Pi(\neg\varphi) = \inf\{1 - \pi(\mathcal{I}) \mid \mathcal{I} \in \Omega,\ \mathcal{I} \models \neg\varphi\}.$$

A possibility distribution π *satisfies* a possibilistic formula (φ, α), written $\pi \models (\varphi, \alpha)$, if $N(\varphi) \geq \alpha$, and it satisfies a possibilistic KB $\mathcal{K} = \{(\varphi_i, \alpha_i) \mid 0 \leq i < n\}$ if it satisfies each $(\varphi_i, \alpha_i) \in \mathcal{K}$. We have that (φ, α) is *entailed* by $\mathcal{K}$, written $\mathcal{K} \models (\varphi, \alpha)$, if all possibility distributions that satisfy $\mathcal{K}$ also satisfy (φ, α). Given $\mathcal{K}$ as above and $\mathcal{I} \in \Omega$, we define the possibility distribution $\pi_{\mathcal{K}}$ as follows: $\pi_{\mathcal{K}}(\mathcal{I}) = 1$, if $\mathcal{I} \models \varphi_i$, for every $(\varphi_i, \alpha_i) \in \mathcal{K}$; otherwise, $\pi_{\mathcal{K}}(\mathcal{I}) = \min\{1 - \alpha_i \mid \mathcal{I} \models \neg\varphi_i, 0 \leq i < n\}$.

The *FO projection* of $\mathcal{K}$ is the set $\mathcal{K}^* = \{\varphi_i \mid (\varphi_i, \alpha_i) \in \mathcal{K}\}$. The *$\alpha$-cut* and the *$\overline{\alpha}$-cut* of $\mathcal{K}$, with $\alpha \in (0, 1]$, are defined respectively as $\mathcal{K}_\alpha = \{(\varphi, \beta) \in \mathcal{K} \mid \beta \geq \alpha\}$ and $\mathcal{K}_{\overline{\alpha}} = \{(\varphi, \beta) \in \mathcal{K} \mid \beta > \alpha\}$. The set of all valuations occurring in $\mathcal{K}$ is $\mathcal{K}^v = \{\alpha \mid (\varphi, \alpha) \in \mathcal{K}\}$. Moreover, $\mathsf{val}(\varphi, \mathcal{K}) = \sup\{\alpha \mid \mathcal{K} \models (\varphi, \alpha)\}$ is the least upper bound of the valuations of formulas entailed by $\mathcal{K}$. Finally, the *inconsistency degree* of $\mathcal{K}$ is defined as $\mathsf{inc}(\mathcal{K}) = \sup\{\alpha \mid \mathcal{K} \models (\bot, \alpha)\}$.

Lemma 2. *(Dubois* et al. *1994) Let $\mathcal{K}$ be a possibilistic KB. For every possibilistic formula (ϕ, α),*

1. *$\mathcal{K} \models (\phi, \alpha)$ iff $\mathcal{K}^*_\alpha \models \phi$;*
2. *$\mathcal{K} \models (\phi, \alpha)$ iff $\alpha \leq \mathsf{val}(\phi, \mathcal{K})$; and*
3. *$\mathcal{K} \models (\phi, \alpha)$ implies $\mathsf{val}(\phi, \mathcal{K}) \in \mathcal{K}^v \cup \{1\}$.*

Proof. Point 1 is a consequence of Propositions 3.5.2, 3.5.5, and 3.5.6, and Point 2 is Property 1 at page 453 in (Dubois *et al.* 1994). We argue about Point 3. By definition of $\pi_{\mathcal{K}}$, for all $\mathcal{I} \in \Omega$, $\pi_{\mathcal{K}}(\mathcal{I})$ is either 1 or $1 - \beta$ for some $\beta \in \mathcal{K}^v$. Let $N_{\mathcal{K}}$ be the necessity measure induced by $\pi_{\mathcal{K}}$. By definition of $N_{\mathcal{K}}$, $N_{\mathcal{K}}(\phi) = \inf\{1 - \pi_{\mathcal{K}}(\mathcal{I}) \mid \mathcal{I} \in \Omega,\ \mathcal{I} \models \neg\varphi\}$. Then, $N_{\mathcal{K}}(\phi) \in \mathcal{K}^v \cup \{0, 1\}$ (recall that $\inf\{\}$ is 1, which is the case for tautologies). By the semantics of possibilistic logic, $N_{\mathcal{K}}(\phi) = \mathsf{val}(\phi, \mathcal{K})$ (Dubois *et al.* 1994, Corollary 3.2.3). As (ϕ, α) is a possibilistic formula, $\alpha > 0$. So, by Point 2, $N_{\mathcal{K}}(\phi) = \mathsf{val}(\phi, \mathcal{K}) \in \mathcal{K}^v \cup \{1\}$. □

We denote by $=_p$ the operator that checks if two numbers are equal up to precision p. For example $0.124 =_2 0.12345$ but $0.124 \neq_3 0.12345$. Assume $\alpha \in (0, 1]$ has finite precision. We write $\mathsf{prec}(\alpha)$ for the precision of α and $\mathsf{prec}(t)$ for $sup\{\mathsf{prec}(\alpha) \mid (\phi, \alpha) \in t\}$. Given an interval I, we write I_p for the set containing all $\alpha \in I$ with $\mathsf{prec}(\alpha) = p$.

Example 3. One can express (1) mutual exclusion of iron deficiency and iron overload and (2) lower necessity of iron overload to be the cause of fatigue than iron deficiency with the possibilistic KB $\{(\forall x(\mathsf{IronDef}(x) \to \neg\mathsf{IronOver}(x)), 1), (\forall x(\mathsf{IronDef}(x) \to \mathsf{Fatigue}(x)), 0.9), (\forall x(\mathsf{IronOver}(x) \to \mathsf{Fatigue}(x), 0.8)\}$. ◁

2.2 Learnability

In learning theory, examples are pieces of information that characterise an abstract target the learner wants to learn. We consider the problem of learning targets represented in decidable fragments of FO logic or in their possibilistic extensions. Examples in our case are formulas expressed in the chosen logic (in this context called 'entailments').

A *learning framework* $\mathfrak{F}$ is a pair $(\mathcal{E}, \mathcal{L})$; where $\mathcal{E}$ is a non-empty and countable set of *examples*, and $\mathcal{L}$ is a non-empty and countable set of *concept representations* (also called *hypothesis space*). Each element l of $\mathcal{L}$ is assumed to be represented using a set of symbols Σ_l (the *signature* of l). In all learning frameworks considered in this work, $\mathcal{E}$ is a set of formulas and $\mathcal{L}$ is a set of KBs (in a chosen language). We say that $e \in \mathcal{E}$ is a *positive example* for $l \in \mathcal{L}$ if $l \models e$ and a *negative example* for l if $l \not\models e$. Given a learning framework $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$, we are interested in the exact identification of a *target* $t \in \mathcal{L}$, by posing queries to oracles. Let $\mathsf{MQ}_{\mathfrak{F},t}$ be the oracle that takes as input some $e \in \mathcal{E}$ and returns 'yes' if $t \models e$ and 'no' otherwise. A *membership query* is a call to the oracle $\mathsf{MQ}_{\mathfrak{F},t}$. Given $t, h \in \mathcal{L}$, a *counterexample* for t and h is an example $e \in \mathcal{E}$ s.t. $t \models e$ and $h \not\models e$ (or vice-versa, $h \models e$ and $t \not\models e$). For every $t \in \mathcal{L}$, we denote by $\mathsf{EQ}_{\mathfrak{F},t}$ an oracle that takes as input a *hypothesis* $h \in \mathcal{L}$ and returns 'yes' if $h \equiv t$ and a counterexample otherwise. There is no assumption regarding which counterexample is chosen by the oracle. An *equivalence query* is a call to $\mathsf{EQ}_{\mathfrak{F},t}$.

Example 4. A blood test to check for vitamin B12 deficiency on patient 42 can be modelled with a call to $\mathsf{MQ}_{\mathfrak{F},t}$ with $(\mathsf{B12Def}(\mathsf{patient\_42}), \alpha)$ for some $\alpha \in (0, 1]$ as input (depending on the result and accuracy of the test). $\lhd$

A *learner* for $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$ is a deterministic algorithm that, for a fixed but arbitrary $t \in \mathcal{L}$, takes Σ_t as input, is allowed to pose queries to $\mathsf{MQ}_{\mathfrak{F},t}$ and $\mathsf{EQ}_{\mathfrak{F},t}$ (without knowing the target t), and that eventually halts and outputs some $h \in \mathcal{L}$ with $h \equiv t$. This notion of an algorithm with access to oracles can be formalised using *learning systems* (Watanabe 1990), where posing a query to an oracle means writing down the query in an (additional) communication tape, entering in a query state, and waiting. The oracle then writes the answer in the communication tape, enters in an answer state, and stops. After that, the learner resumes its execution and can now read the answer in the communication tape.

We say that $\mathfrak{F}$ is (exactly) *learnable* if there is a learner for $\mathfrak{F}$ and that $\mathfrak{F}$ is *polynomial time learnable* if it is learnable by a learner A such that at every step (the time used by an oracle to write an answer is *not* taken into account) of computation the time used by A up to that step is bounded by a polynomial $p(|t|, |e|)$, where $t \in \mathcal{L}$ is the target and $e \in \mathcal{E}$ is the largest counterexample seen so far. We denote by PTIMEL the class of learning frameworks which are polynomial time learnable and the complexity of the entailment problem is in PTIME[1]. We also consider cases in which the learner can only pose one type of query (only membership or only equivalence queries). Whenever this is the case we write this explicitly.

Let $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$ be a learning framework where $\mathcal{E}$ is a set of FO formulas and $\mathcal{L}$ is a set of FO KBs. We call such $\mathfrak{F}$ an *FO learning framework*. We say that $\mathfrak{F}$ is *non-trivial* if $\mathcal{L}$ contains a non-trivial FO KB; and that it is *safe* if $l \in \mathcal{L}$ implies that $l' \in \mathcal{L}$, for all $l' \subseteq l$. A *possibilistic extension* l_π of an FO KB l is a possibilistic KB obtained by adding a possibilistic valuation α to every formula $\varphi \in l$. The possibilistic extension $\mathfrak{F}_\pi$ of $\mathfrak{F}$ is the pair $(\mathcal{E}_\pi, \mathcal{L}_\pi)$ where $\mathcal{L}_\pi$ is the set of all possibilistic extensions of each $l \in \mathcal{L}$, and $\mathcal{E}_\pi$ is the set of all possibilistic formulas entailed by an element of $\mathcal{L}_\pi$.

We write $\mathbb{N}^+$ for the set of positive natural numbers. Given $p \in \mathbb{N}^+$, we denote by $\mathfrak{F}^p_\pi = (\mathcal{E}_\pi, \mathcal{L}^p_\pi)$ the result of removing from $\mathcal{L}_\pi$ in $\mathfrak{F}_\pi$ every $l \in \mathcal{L}_\pi$ that does not satisfy $\mathsf{prec}(l) = p$.

Remark 1. *Let $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$ be an FO learning framework and let $t \in \mathcal{L}$ be the target. If a learner A has access to $\mathsf{MQ}_{\mathfrak{F},t}$ then we can assume w.l.o.g. that all counterexamples returned by $\mathsf{EQ}_{\mathfrak{F},t}$ are* positive*: the learner can check whether each $\phi \in h$ is entailed by t. The same holds for $\mathfrak{F}_\pi$.*

3 Learnability Results

We start by studying the problem of whether there is a learner for a learning framework such that it always terminates with a hypothesis equivalent to the target. The main difficulty in learning with only membership queries (even for plain FO settings) is that the learner would 'not know' whether it has found a formula equivalent to a (non-trivial) target.

Example 5. Let $\Phi_n := \exists x_1 \ldots \exists x_n. \bigwedge_{0 \leq i < n} r(x_i, x_{i+1})$. A learner may ask membership queries of the form $\exists x_0 \Phi_n$ for an arbitrarily large n without being able to distinguish whether the target theory is $\exists x_0 \Phi_n$ or $\forall x_0 (\Phi_n \to \Phi_{n+1})$ (knowing the signature of the target theory does not help the learner). $\lhd$

For possibilistic theories, another difficulty arises even for the propositional case. As the precision of a formula can be arbitrarily high, the learner may not know when to stop (e.g., is the target $(p, 0.1)$? or $(p, 0.11)$?). Theorem 6 states that, except for trivial cases, learnability cannot be guaranteed.

Theorem 6. *Let $\mathfrak{F}$ be a non-trivial FO learning framework. $\mathfrak{F}_\pi$ is not (exactly) learnable with only membership queries.*

Sketch. The existence of a learner A for the possibilistic extension $\mathfrak{F}_\pi = (\mathcal{E}_\pi, \mathcal{L}_\pi)$ of a non-trivial learning framework $\mathfrak{F}$ would imply the existence of a procedure that terminates in n steps. A would not distinguish between the elements of $\mathcal{L}_\pi$ with precision higher than n. $\square$

If the precision of the target is known or fixed, learnability of an FO learning framework can be transferred to its possibilistic extension. We state this in Theorem 8. To show this theorem, we use the following technical result.

Lemma 7. *Let t be a possibilistic KB. Let I be a set of valuations such that $t^v \subseteq I$. If for each $\alpha \in I$ there is some FO KB k^*_α such that $k^*_\alpha \equiv t^*_\alpha$ then $t \equiv \{(\phi, \alpha) \mid \phi \in k^*_\alpha, \alpha \in I\}$.*

Proof. Let $h = \{(\phi, \alpha) \mid \phi \in k^*_\alpha, \alpha \in I\}$. Assume $h \models (\phi, \gamma)$. If $\gamma = 1$ and $\gamma \notin I$ then ϕ is a tautology. In this case, for all $\beta \in (0, 1]$, $t \models (\phi, \beta)$. Suppose this is not the case. By Points 2 and 3 of Lemma 2, $\gamma \leq \alpha$, $\alpha = \mathsf{val}(\phi, h) \in h^v \cup \{1\}$. Also, $h \models (\phi, \alpha)$. By construction of h, $h^v = I$, so $\alpha \in I$. Moreover, for every $\beta \in I$, we know that $h^*_\beta = k^*_\beta$. Therefore $k^*_\alpha \equiv h^*_\alpha$. By Point 1 of Lemma 2, $h \models (\phi, \alpha)$ implies $h^*_\alpha \models \phi$. Then, $k^*_\alpha \models \phi$. As $k^*_\alpha \equiv t^*_\alpha$, we have that $t^*_\alpha \models \phi$. Again by Point 1 (of Lemma 2), $t^*_\alpha \models \phi$ iff $t \models (\phi, \alpha)$. Since $\alpha \geq \gamma$, $t \models (\phi, \gamma)$ by Point 2. The other direction can be proved similarly. $\square$

Theorem 8. *Suppose $\mathfrak{F}$ is an FO learning framework that is learnable with only membership queries. For all $p \in \mathbb{N}^+$, $\mathfrak{F}^p_\pi = (\mathcal{E}_\pi, \mathcal{L}^p_\pi)$ is learnable with only membership queries.*

Proof. Let A be a learner for $\mathfrak{F}$ and let $t \in \mathcal{L}^p_\pi$ be the target. For each $\alpha \in (0, 1]_p$, we run an instance of A, denoted A_α. Whenever A_α calls $\mathsf{MQ}_{\mathfrak{F},t^*_\alpha}$ with ϕ as input, we

[1] In general, non-trivial algorithms need to perform entailment checks to combine the information of the examples. So polynomial time learning algorithms are normally for logics in which the entailment problem is tractable. This is the case e.g. for the Horn results mentioned in the Introduction.

call $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ with (ϕ,α) as input. By Point 1 of Lemma 2, $\mathsf{MQ}_{\mathfrak{F},t^*_\alpha}(\phi) = \mathsf{MQ}_{\mathfrak{F}_\pi,t}(\phi,\alpha)$. Since A is a learner for $\mathfrak{F}$, every A_α eventually halts and outputs a hypothesis k^*_α such that $k^*_\alpha \equiv t^*_\alpha$. Since $t \in \mathcal{L}^p_\pi$, $t^v \subseteq (0,1]_p$. By Lemma 7, $t \equiv \{(\phi,\alpha) \mid \phi \in k^*_\alpha, \alpha \in (0,1]_p\}$. Thus, we can transfer learnability of $\mathfrak{F}$ (with only membership queries) to $\mathfrak{F}^p_\pi$. □

If, e.g., $\mathsf{MQ}_{\mathfrak{F}_\pi,t}((\phi,0.01)) =$ 'yes', $\mathsf{MQ}_{\mathfrak{F}_\pi,t}((\phi,0.02)) =$ 'no', and the precision of the target is 2, then $\mathsf{val}(\phi,t) = 0.01$. So, knowing the precision is important for learning with membership queries only. If equivalence queries are allowed then a learner can build a hypothesis equivalent to the target *without knowing the precision* in advance (Theorem 9).

Theorem 9. *The possibilistic extension $\mathfrak{F}_\pi$ of a learnable FO learning framework $\mathfrak{F}$ is learnable with only equivalence queries.*

Proof. Every FO learning framework $\mathfrak{F}$ is learnable with only equivalence queries. Indeed, a naive learner A for $\mathfrak{F}$ is one that enumerates all $l \in \mathcal{L}$ built using symbols from Σ_t, taken as input, and asks the possible hypothesis to oracle $\mathsf{EQ}_{\mathfrak{F},t}$, one by one. The learner does not know the size of the target in advance but it can estimate it to be n, ask all possible hypothesis of this size, then increase to $n+1$, and so on. Since the target is finite, eventually A halts and outputs h equivalent to t. For $\mathfrak{F}_\pi$, a similar naive learner A_π exists, but in this case, it also needs to estimate the precision of the target and increase it as it navigates the search space. As the precision of the target is finite, eventually A_π also halts and outputs an equivalent hypothesis. □

If both membership and equivalence query oracles are available, learnability is guaranteed by the previous theorem (even when the precision of the target is unknown).

Corollary 1. *Let $\mathfrak{F}$ be an FO learning framework. $\mathfrak{F}$ is learnable iff $\mathfrak{F}_\pi$ is learnable.*

4 Polynomial Time Reduction

We now investigate whether results showing that an FO learning framework is in PTIMEL can be transferred to their possibilistic extensions and vice-versa. Theorem 10 shows the transferability of PTIMEL membership from the possibilistic extension $\mathfrak{F}_\pi$ of an FO learning framework $\mathfrak{F}$ to $\mathfrak{F}$.

Theorem 10. *Let $\mathfrak{F}$ be an FO learning framework. If $\mathfrak{F}_\pi$ is in* PTIMEL *then $\mathfrak{F}$ is in* PTIMEL.

Proof. In our proof, we use the following claim.

Claim 10.1. *Let k be an FO KB and let t be the possibilistic KB $\{(\phi,1) \mid \phi \in k\}$. For all (ϕ,α), $k \models \phi$ iff $t \models (\phi,\alpha)$.*

Proof. If $t \models (\phi,\alpha)$, since $t^* \models t^*_\alpha$ and $k = t^*$, $k \models \phi$. If $k \models \phi$, by construction $t^*_1 \models \phi$. By Point 1 of Lemma 2, $t^*_1 \models \phi$ iff $t \models (\phi,1)$, so, for all $\alpha \in (0,1]$, $t \models (\phi,\alpha)$. □

Let $\mathfrak{F} = (\mathcal{E},\mathcal{L})$ and let $k \in \mathcal{L}$ be the target. Since $\mathfrak{F}_\pi$ is in PTIMEL, there is a learner A_π for $\mathfrak{F}_\pi$. We start the execution of A_π that attempts to learn a hypothesis h equivalent to $t = \{(\phi,1) \mid \phi \in k\}$. By Claim 10.1, for all $\alpha \in (0,1]$, $\mathsf{MQ}_{\mathfrak{F}_\pi,t}((\phi,\alpha)) = \mathsf{MQ}_{\mathfrak{F},k}(\phi)$. Also, we can simulate a call to $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with h as input by calling $\mathsf{EQ}_{\mathfrak{F},k}$ with h^* as input. By Claim 10.1, for all $\alpha \in (0,1]$, $k \models \phi$ iff $t \models (\phi,\alpha)$, in particular, for $\alpha = 1$. By Remark 1, we can assume that all counterexamples returned by $\mathsf{EQ}_{\mathfrak{F},k}$ are positive. Whenever we receive a (positive) counterexample ϕ, we return $(\phi,1)$ to A_π. Eventually, A_π will output a hypothesis $h \equiv t$ in polynomial time w.r.t. $|t|$ and the largest counterexample received so far. Clearly, h^* is as required. □

The converse of Theorem 10 does not hold as shown by Theorem 11. Simple FO learning frameworks can become difficult to learn when extended with possibilistic valuations because algorithms also have to deal with multiple valuations.

Theorem 11. *There exists an FO learning framework $\mathfrak{F}$ such that $\mathfrak{F}$ is in* PTIMEL *but $\mathfrak{F}_\pi = (\mathcal{E}_\pi,\mathcal{L}_\pi)$ is not in* PTIMEL.

Proof. Let $\mathfrak{F} = (\mathcal{E},\mathcal{L})$ be an FO learning framework that is *not* in PTIMEL. Such $\mathfrak{F}$ exists, one can consider, for instance, the $\mathcal{EL}$ learning framework (Konev *et al.* 2018, Theorem 68)[2]. We use $\mathfrak{F}$ to define the learning framework $\mathfrak{F}^\perp = (\mathcal{E},\mathcal{L}^\perp)$ where $\mathcal{L}^\perp = \{h \cup \{\phi,\neg\phi\} \mid h \in \mathcal{L}\}$ for a fixed but arbitrary non-trivial FO formula ϕ. Even though $\mathfrak{F}$ is not learnable in polynomial time, $\mathfrak{F}^\perp$ is. The learner can learn any $l \in \mathcal{L}^\perp$ by returning the hypothesis $\{\perp\}$ (in constant time). Assume that $\mathfrak{F}^\perp_\pi = (\mathcal{E}_\pi,\mathcal{L}^\perp_\pi)$ is in PTIMEL. This means that for every target $l \in \mathcal{L}^\perp_\pi$ we can learn in polynomial time a hypothesis h such that $h \equiv l$. By construction, for every $t \in \mathcal{L}$ there is $l \in \mathcal{L}^\perp_\pi$ such that $t \equiv l^*_{\overline{\mathsf{inc}(l)}}$. By learning h such that $h \equiv l$ we have also learned a hypothesis h such that $h^*_{\overline{\mathsf{inc}(h)}} \equiv t$. By Theorem 10, $\mathfrak{F} \in$ PTIMEL, which contradicts our assumption that this is not the case. Therefore we have found an FO learning framework $\mathfrak{F}^\perp$ that is is in PTIMEL but its possibilistic extension $\mathfrak{F}^\perp_\pi$ is not in PTIMEL. □

The FO learning framework $\mathfrak{F}^\perp$ in the proof of Theorem 11 is not safe (see definition in Subsection 2.2) because, for $l \not\subseteq \{\phi,\neg\phi\}$ we have $l \in \mathcal{L}^\perp$ with $(l \setminus \{\phi,\neg\phi\}) \notin \mathcal{L}^\perp$. Intuitively, non-safe learning frameworks allow cases in which the target is easy to learn if we aim at learning the *whole* target, not a *subset* of it. In the following, we focus on FO learning frameworks that are safe[3]. The first transferability result we present is for the case in which the learner has access to only membership queries. Before showing the reduction, we define the procedure $\mathsf{FindValuation}_t$ that takes as input a precision p and a formula ϕ and returns the highest valuation β with precision p of a formula ϕ entailed by the target t (or zero if it is not entailed). That is, β is such that $\beta =_p \mathsf{val}(\phi,t)$. For any $\gamma \in [0,1]_p$ the procedure can check if $t \models (\phi,\gamma)$ by calling the oracle $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ with (ϕ,γ) as input. To compute β such that $\beta =_p \mathsf{val}(\phi,t)$, $\mathsf{FindValuation}_t$ performs a binary search on $[0,1]_p$. Lemma 12 states the correctness and the complexity of $\mathsf{FindValuation}_t$.

Lemma 12. *Let $\mathfrak{F}_\pi = (\mathcal{E}_\pi,\mathcal{L}_\pi)$ be a possibilistic learning framework and let $t \in \mathcal{L}_\pi$ be the target.* $\mathsf{FindValuation}_t$*, with*

[2] Non-polynomial query learnability is proved in (Konev *et al.* 2018, Theorem 68), which implies non-polynomial time learnability.

[3] All learning from entailment results we found in the literature could be formulated in terms of safe learning frameworks.

input a precision $p \in \mathbb{N}^+$ and $\phi \in \mathcal{E}_\pi$, runs in polynomial time in p and $|\phi|$ and outputs β such that $\beta =_p \mathsf{val}(\phi, t)$.

Sketch. By Point 2 of Lemma 2, $\mathsf{FindValuation}_t$ can determine β such that $\beta =_p \mathsf{val}(\phi, t)$ by performing a binary search on the interval of numbers $[0, 1]_p$. So the number of iterations is bounded by $log_2(10^p + 1)$, which is polynomial in p. Clearly, each iteration can be performed in polynomial time in $|\phi|$ and p (each call to the membership oracle $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ counts as one step of computation). □

In our next theorem, we show that, for safe FO learning frameworks, polynomial time results with only membership queries can be transferred to their possibilistic extensions if the precision of the target is known (recall that, by Theorem 6, we cannot remove this assumption).

Theorem 13. *Let $\mathfrak{F}$ be a safe FO learning framework. For all $p \in \mathbb{N}^+$, when only membership queries can be asked, $\mathfrak{F}$ is in* PTIMEL *iff $\mathfrak{F}^p_\pi$ is in* PTIMEL.

Proof. To show the transferability of PTIMEL membership from $\mathfrak{F}$ to $\mathfrak{F}_\pi$, we use the following claim.

Claim 13.1. *Assume $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$ is safe and in* PTIMEL *with only membership queries. For every $p \in \mathbb{N}^+$ and framework $\mathfrak{F}^p_\pi = (\mathcal{E}_\pi, \mathcal{L}^p_\pi)$ with $t \in \mathcal{L}^p_\pi$, given a valuation α with $\mathsf{prec}(\alpha) = p$, one can learn $k^*_{\overline{\alpha}}$ such that $k^*_{\overline{\alpha}} \equiv t^*_{\overline{\alpha}}$ in time polynomial w.r.t. $|t|$ with only membership queries.*

Proof. We start the execution of a polynomial time learner A for $\mathfrak{F}$. Whenever A calls $\mathsf{MQ}_{\mathfrak{F},t^*_{\overline{\alpha}}}$ with ϕ as input, we call $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ with $(\phi, \alpha + 10^{-p})$ as input and we return the same answer to A. By Point 1 of Lemma 2, $\mathsf{MQ}_{\mathfrak{F},t^*_{\overline{\alpha}}}(\phi) = \mathsf{MQ}_{\mathfrak{F}_\pi,t}(\phi, \alpha + 10^{-p})$. Since $\mathfrak{F}$ is safe, A will build a hypothesis $k^*_{\overline{\alpha}}$ such that $k^*_{\overline{\alpha}} \equiv t^*_{\overline{\alpha}}$ in polynomial time w.r.t. $|t|$. □

We set $\gamma := 0$ and $S := \emptyset$. By Claim 13.1 we can find in polynomial time w.r.t. $|t|$ a hypothesis $k^*_{\overline{\gamma}}$ such that $k^*_{\overline{\gamma}} \equiv t^*_{\overline{\gamma}}$. For every $\phi \in k^*_{\overline{\gamma}}$, we run $\mathsf{FindValuation}_t$ with $p = \mathsf{prec}(t)$ and ϕ as input to find $\mathsf{val}(\phi, t)$. In this way, by Point 3 of Lemma 2 and Lemma 12, we identify in polynomial time w.r.t. $|t|$ some $\beta \in t^v \cup \{1\}$ such that $k^*_{\overline{\gamma}} \equiv t^*_\beta$. We set $k^*_\beta := k^*_{\overline{\gamma}}$ and add k^*_β to S. Then, we update γ to the value β and apply Claim 13.1 again. For every $\phi \in k^*_{\overline{\gamma}}$, we run $\mathsf{FindValuation}_t$ again with $p = \mathsf{prec}(t)$ and ϕ as input to find $\mathsf{val}(\phi, t)$. We repeat this process until we find $k^*_{\overline{\gamma}} \equiv \emptyset$ or $\gamma + 10^{-p} > 1$. Each time we run $\mathsf{FindValuation}_t$, we identify a higher valuation in t^v. Therefore, this happens at most $|t^v|$ times. For all $\alpha \in t^v$, there is $k^*_\alpha \in S$ that satisfies $k^*_\alpha \equiv t^*_\alpha$, therefore, by Lemma 7,

$$h = \bigcup_{k^*_\alpha \in S} \{(\phi, \alpha) \mid \phi \in k^*_\alpha\}$$

is such that $h \equiv t$.

We now show the transferability of PTIMEL membership from $\mathfrak{F}_\pi$ to $\mathfrak{F}$. Let $k \in \mathcal{L}$ be the target. We start the execution of a learner A_π for $\mathfrak{F}_\pi$ that attempts to learn a hypothesis equivalent to $t = \{(\phi, 1) \mid \phi \in k\}$. By Claim 10.1 of Theorem 10, we can simulate a call to $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ with input $(\phi, 1)$ by calling $\mathsf{MQ}_{\mathfrak{F},k}$ with ϕ as input and returning the same answer to A_π. A_π terminates in polynomial time w.r.t. $|t|$ with a hypothesis h such that $h \equiv t$. As $h^* \equiv t^* = k$, h^* is as required. □

When we want to transfer learnability results from $\mathfrak{F}$ to $\mathfrak{F}_\pi$ it is important to learn one h_α such that $h_\alpha \equiv t_\alpha$ for each $\alpha \in t^v$, where t is the target (Example 14).

Example 14. *Let $t = \{(p \to q_1, 0.3), (p \to q_2, 0.7)\}$. We can use the polynomial time algorithm for propositional Horn (Frazier and Pitt 1993) to learn a hypothesis $k^* = \{p \to (q_1 \wedge q_2)\} \equiv t^*$. However, if $h = \{(\phi, \mathsf{val}(\phi, t)) \mid \phi \in k^*\}$ then $h = \{(p \to (q_1 \wedge q_2), 0.3)\} \not\equiv t$.*

A learner that has access to both membership and equivalence query oracle has a way of finding the precision of the target when it is unknown. With membership queries, we can use $\mathsf{FindValuation}_t$ to find the valuation of formulas up to a given precision. By Lemma 15, we can obtain useful information about the precision of the target with the counterexamples obtained after an equivalence query.

Lemma 15. *Assume $\mathfrak{F}_\pi = (\mathcal{E}_\pi, \mathcal{L}_\pi)$ is the possibilistic extension of a safe FO learning framework and $t \in \mathcal{L}_\pi$ is the target. Given $p \in \mathbb{N}^+$, one can determine that $p < \mathsf{prec}(t)$ or compute $h \in \mathcal{L}_\pi$ such that $h \equiv t$, in polynomial time w.r.t. $|t|$, p, and the largest counterexample seen so far.*

Proof. In our proof, we use the following claims.

Claim 15.1. *Given $h \in \mathcal{L}_\pi$ such that $t \models h$, one can construct in polynomial time in $|h|$ some $h' \in \mathcal{L}_\pi$ such that $t \models h' \models h$ and, for all $(\phi, \alpha) \in h'$, $t \models (\phi, \alpha)$ and $\alpha =_{\mathsf{prec}(h')} \mathsf{val}(\phi, t)$.*

Proof. Let h' be the set of all (ϕ, β) such that $(\phi, \alpha) \in h$ and $\mathsf{FindValuation}_t$ returns β with ϕ and $\mathsf{prec}(h)$ as input. As $t \models h$, by construction of h', $t \models h' \models h$. By Lemma 12, h' can be constructed in polynomial time in $|h|$ and is as required. □

Claim 15.2. *Let $h \in \mathcal{L}_\pi$ be such that, for all $(\phi, \alpha) \in h$, $t \models (\phi, \alpha)$ and $\alpha =_{\mathsf{prec}(h)} \mathsf{val}(\phi, t)$. If $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with input h returns (ϕ, α) then either we know that $\mathsf{prec}(t) > \mathsf{prec}(h)$ or $h^*_\beta \not\models \phi$ where $\beta =_{\mathsf{prec}(h)} \mathsf{val}(\phi, t)$.*

Proof. By Point 1 of Lemma 2, $h^*_\beta \models \phi$ iff $h \models (\phi, \beta)$. If $h \models (\phi, \beta)$ or $\beta = 0$ (note: β can be 0 because, e.g., $0.01 =_1 0$), then $\mathsf{prec}(\mathsf{val}(\phi, t)) > \mathsf{prec}(h)$. By Point 3 of Lemma 2, $\mathsf{val}(\phi, t) \in t^v \cup \{1\}$, so $\mathsf{prec}(t) > \mathsf{prec}(h)$. □

By Remark 1, we can assume at all times in this proof that any hypothesis constructed is entailed by the target (possibilistic or not). Moreover, by Claim 15.1, we can assume that, for any target and hypothesis $t, h \in \mathcal{L}_\pi$, we have that, for all $(\phi, \alpha) \in h$, $t \models (\phi, \alpha)$ and $\alpha =_{\mathsf{prec}(h)} \mathsf{val}(\phi, t)$. So we can assume at all times in our proof that the hypothesis h we construct (Equation 1) satisfies the conditions of Claim 15.2.

Let A be a polynomial time learner[4] for $\mathfrak{F}$. As in the proof of Theorem 13, we run multiple instances of A. We denote by R the set of instances of A. Each instance in R is denoted A_β and attempts to learn a hypothesis equivalent to t^*_β, where β is a valuation. We sometimes write A^n_β to indicate that the instance A_β has asked n equivalence queries so far. We denote by $k^{\beta,n}$ the hypothesis given as input by A^n_β when it asks its n-th equivalence query. For $n = 0$, we assume that $k^{\beta,n} = \emptyset$.

Initially, $\mathsf{R} := \{A^0_{10^{-p}}\}$. Whenever $A_\beta \in \mathsf{R}$ asks a membership query with input $\phi \in \mathcal{E}$, by Point 1 of Lemma 2, we can simulate $\mathsf{MQ}_{\mathfrak{F},t^*_\beta}$ by calling $\mathsf{MQ}_{\mathfrak{F}_\pi,t}$ with (ϕ, β) as input and returning the same answer to A_β. Let h_0 be $\{(\phi_\top, \alpha)\}$ where $\phi_\top$ is a tautology and α is a valuation with $\mathsf{prec}(\alpha) = p$. Whenever $A^n_\beta \in \mathsf{R}$ asks its n-th equivalence query, we leave A^n_β waiting in the query state (see description of a learning system in Subsection 2.2). When all $A^m_\alpha \in \mathsf{R}$ are waiting in the query state, we create

$$h := \bigcup_{A^m_\alpha \in \mathsf{R}} \{(\phi, \alpha) \mid \phi \in k^{\alpha,m}\} \cup h_0 \qquad (1)$$

and call $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with h as input (note: each instance $A_\alpha \in \mathsf{R}$ may have asked a different number of equivalence queries when A^n_β asks its n-th equivalence query). If the answer is 'yes', we have computed h such that $h \equiv t$ and we are done. Upon receiving a (positive) counterexample (ϕ, γ), we run $\mathsf{FindValuation}_t$ with ϕ and $\mathsf{prec}(h)$ as input and compute a valuation β such that $\beta =_{\mathsf{prec}(h)} \mathsf{val}(\phi, t)$ (Lemma 12). If $A_\beta \notin \mathsf{R}$, we start the execution of the instance A_β of algorithm A and add A_β to R. Otherwise, $A_\beta \in \mathsf{R}$ and we check whether $k^{\beta,m} \models \phi$ (assume m is the number of equivalence queries posed so far by A_β). If $k^{\beta,m} \models \phi$ then, by Claim 15.2, we know that $\mathsf{prec}(h) < \mathsf{prec}(t)$ then we are done. If $k^{\beta,m} \not\models \phi$ then ϕ is a (positive) counterexample for $k^{\beta,m}$ and t^*_β. We return ϕ to every $A^m_\alpha \in \mathsf{R}$ such that $\alpha \leq \beta$ and $k^{\alpha,m} \not\models \phi$ and these instances resume their executions. Observe that, since $h_0 \subseteq h$, by the construction of h, at all times $\mathsf{prec}(h) = p$.

We now argue that this procedure terminates in polynomial time w.r.t. $|t|$, p, and the largest counterexample seen so far. Since there is only one instance A_β in R for each valuation β such that $\beta =_p \mathsf{val}(\phi, t)$, by Point 3 of Lemma 2, we have that at all times $|\mathsf{R}|$ is linear in $|t^v|$, which is bounded by $|t|$. By Lemma 12, whenever we run $\mathsf{FindValuation}_t$ to compute a valuation with ϕ and p as input, only polynomially many steps in $|\phi|$ and p are needed. Since $\mathfrak{F}$ is safe and A is a polynomial time learner for $\mathfrak{F}$ either we can determine that $p < \mathsf{prec}(t)$ or each $A_\beta \in \mathsf{R}$ terminates, in polynomial time in the size of t^*_β and the largest counterexample seen so far, and outputs $k^{\beta,n} = h^*_\beta$ such that $h^*_\beta \equiv t^*_\beta$. In this case, by Lemma 7, $h \equiv t$ and the process terminates. □

The constructive proof of Lemma 15 delineates the steps made in Example 16 where the precision of the target is 1.

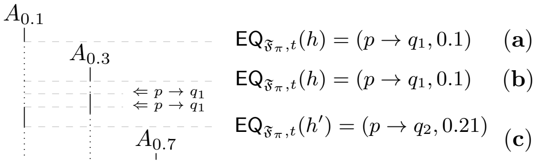

$A_{0.1}$
$A_{0.3}$
$A_{0.7}$
$\Leftarrow p \to q_1$
$\Leftarrow p \to q_1$
$\mathsf{EQ}_{\mathfrak{F}_\pi,t}(h) = (p \to q_1, 0.1)$ (**a**)
$\mathsf{EQ}_{\mathfrak{F}_\pi,t}(h) = (p \to q_1, 0.1)$ (**b**)
$\mathsf{EQ}_{\mathfrak{F}_\pi,t}(h') = (p \to q_2, 0.21)$ (**c**)

Figure 1: Multiple instances of algorithm A in Example 16. Time flows top-down. A dotted line means that the learner is waiting in query state, a continuous line means that the learner is running.

Example 16. Let $\mathfrak{F} = (\mathcal{E}, \mathcal{L})$ be the safe learning framework where $\mathcal{L}$ is the set of all propositional Horn KBs and $\mathcal{E}$ is the set of all (propositional) Horn clauses. Let $t \in \mathcal{L}_\pi$ and A be, respectively, the target and the learner of Example 14. Following our argument in Lemma 15, we start an instance $A_{0.1}$ of A. When $A_{0.1}$ is waiting in the query state, we build $h = \{(\phi_\top, 0.1)\}$ (Equation 1) and call $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with h as input (Point (a) in Figure 1). Assume we receive the positive counterexample $(p \to q_1, 0.1)$. We run $\mathsf{FindValuation}_t$ with 1 and $p \to q_1$ as input, which computes $\mathsf{val}(p \to q_1, t) = 0.3$. Since $A_{0.3} \notin \mathsf{R}$, we start $A_{0.3}$. When all learners are waiting in the query state, we call again $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with h as input (Point (b) in Figure 1). At this point, $\mathsf{R} = \{A_{0.1}, A_{0.3}\}$.

Assume we receive $(p \to q_1, 0.1)$ again. We have that $\mathsf{val}(p \to q_1, t) = 0.3$ and $A_{0.3} \in \mathsf{R}$. Since $k^{0.3,1} \not\models p \to q_1$ and $k^{0.1,1} \not\models p \to q_1$, we return $p \to q_1$ to both $A^1_{0.1}$ and $A^1_{0.3}$ and they resume their executions. All learners will eventually be waiting in query state. When this happens we call $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with $h' = \{(\phi_\top, 0.1), (p \to q_1, 0.1), (p \to q_1, 0.3)\}$ as input.

Assume the response is $(p \to q_2, 0.21)$. We run $\mathsf{FindValuation}_t$ with 1 and $p \to q_2$ as input, which returns $\mathsf{val}(p \to q_2, t) = 0.7$. As before, we start $A_{0.7}$ (Point (c) in Figure 1) and add it to R. When all learners are waiting again we call $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$ with h' as input. Assume we receive $(p \to q_2, 0.1)$. We then send $p \to q_2$ to every learner in R. Next time we call $\mathsf{EQ}_{\mathfrak{F}_\pi,t}$, with $h' \cup \{(p \to q_2, 0.7), (p \to q_2, 0.3), (p \to q_2, 0.1)\}$ as input. The answer is 'yes' and we are done. ◁

In some cases, the learner can discover if the precision of the hypothesis needs to increase (Example 17).

Example 17. Assume the target is $t = \{(p \to q, 0.32)\}$ and the learner built the hypothesis $h = \{(p \to q, 0.3)\}$. Similarly to Example 16, the precision of the hypothesis is set to 1. A future equivalence query will return the counterexample $h = \{(p \to q, \alpha)\}$ with $\alpha > 0.3$. The learner will run $\mathsf{FindValuation}_t$ with input 1 and $p \to q$, which will return 0.3. Since $h \models (p \to q, 0.3)$, this can happen only if the precision of the hypothesis is low. ◁

A direct consequence of Lemma 15 is Theorem 18.

Theorem 18. *For every safe FO learning frameworks $\mathfrak{F}$ we have, $\mathfrak{F}$ is in* PTIMEL *iff $\mathfrak{F}_\pi$ is in* PTIMEL.

Proof. One direction holds by Theorem 10. We prove the other direction. Let $\mathfrak{F}$ be a safe FO learning framework in PTIMEL and let $\mathfrak{F}_\pi = (\mathcal{E}_\pi, \mathcal{L}_\pi)$ be its possibilistic extension.

[4] Assume w.l.o.g. that A always eventually asks an equivalence query until it finds an equivalent hypothesis (but may execute other steps and ask membership queries between each equivalence query).

Consider a learner that initially estimates precision p of the target $t \in \mathcal{L}_\pi$ to be 1. Using Lemma 15, we can assume that this learner can either determine that $p < \mathsf{prec}(t)$ or find a hypothesis h such that $h \equiv t$, in time polynomial with respect to $|t|$, p and the largest counterexample seen so far. In the former case, this learner sets the estimated precision p of the target to $p + 1$. This happens at most $\mathsf{prec}(t)$ times, which is bounded by $|t|$. As a consequence, $\mathfrak{F}_\pi$ is in PTimeL. □

We end this section recalling a connection between the exact and the PAC learning models. In the PAC model, a learner receives classified examples drawn according to a probability distribution and attempts to create a hypothesis that approximates the target. It is known that polynomial time results for the exact learning model can be transferred to the PAC learning model (Valiant 1984) extended with membership queries (Theorem 19).

Theorem 19 ((Angluin 1988; Mohri *et al.* 2012))**.** *Let* PTimePL *be the class of all learning frameworks that are PAC learnable with membership queries in polynomial time. Then,* PTimeL $\subseteq$ PTimePL.

By Theorems 18 and 19, the following holds.

Corollary 2. *For all safe FO learning frameworks* $\mathfrak{F}$*, if* $\mathfrak{F} \in$ PTimeL *then* $\mathfrak{F}_\pi \in$ PTimePL.

5 Conclusion

Uncertainty is widespread in learning processes. Among different uncertainty formalisms, possibilistic logic stands out because of its ability to express preferences among worlds and model ignorance. We presented the first study on the exact (polynomial) learnability of possibilistic theories. Various algorithms designed for exact learning fragments of first-order logic can be adapted to learn their possibilistic extensions. We leave open the problem of polynomial time transferability with only equivalence queries.

Ack. We thank Andrea Mazzullo for joining the discussion.

References

Parul Agarwal and Dr. H. S. Nayal. Possibility theory versus probability theory in fuzzy measure theory. *International Journal of Engineering Research and Applications*, 5(5):37–43, 2015.

Dana Angluin, Michael Frazier, and Leonard Pitt. Learning conjunctions of horn clauses. *Machine Learning*, 9:147–164, 1992.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Hiroki Arimura. Learning acyclic first-order Horn sentences from entailment. In *International Workshop on Algorithmic Learning Theory*, pages 432–445, 1997.

Salem Benferhat, Didier Dubois, and Henri Prade. Representing default rules in possibilistic logic. In *KR*, page 673–684. Morgan Kaufmann Publishers Inc., 1992.

Yassine Djouadi, Didier Dubois, and Henri Prade. Possibility theory and formal concept analysis: Context decomposition and uncertainty handling. pages 260–269, 06 2010.

D. Dubois and H. Prade. Fuzzy sets and probability: misunderstandings, bridges and gaps. In *IEEE International Conference on Fuzzy Systems*, volume 2, pages 1059–1068, 1993.

Didier Dubois and Henri Prade. Possibility theory, probability theory and multiple-valued logics: A clarification. *Ann. Math. Artif. Intell.*, 32(1-4):35–66, 2001.

Didier Dubois and Henri Prade. Possibility theory and its applications: Where do we stand? In *Handbook of Computational Intelligence*, Springer Handbooks, pages 31–60. Springer, 2015.

D. Dubois, J. Lang, and H. Prade. *Possibilistic Logic*, page 439–513. Oxford University Press, Inc., USA, 1994.

Michael Frazier and Leonard Pitt. Learning from entailment: An application to propositional Horn sentences. In *ICML*, pages 120–127, 1993.

Montserrat Hermo and Ana Ozaki. Exact learning: On the boundary between horn and CNF. *TOCT*, 12(1):4:1–4:25, 2020.

Boris Konev, Carsten Lutz, Ana Ozaki, and Frank Wolter. Exact Learning of Lightweight Description Logic Ontologies. *Journal of Machine Learning Research*, 18(201):1–63, 2018.

Ondrej Kuzelka, Jesse Davis, and Steven Schockaert. Encoding markov logic networks in possibilistic logic. In *UAI*, pages 454–463. AUAI Press, 2015.

Ondrej Kuzelka, Jesse Davis, and Steven Schockaert. Learning possibilistic logic theories from default rules. In *IJCAI*, pages 1167–1173, 2016.

Ondrej Kuzelka, Jesse Davis, and Steven Schockaert. Induction of interpretable possibilistic logic theories from relational data. In *IJCAI*, pages 1153–1159, 2017.

Jérôme Lang. *Possibilistic Logic: Complexity and Algorithms*, pages 179–220. Springer Netherlands, 2000.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.

Simon Parsons and Anthony Hunter. *A Review of Uncertainty Handling Formalisms*, pages 8–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

Henri Prade and Mathieu Serrurier. Bipolar version space learning. *International Journal of Intelligent Systems*, 23(10):1135–1152, 2008.

Chandra Reddy and Prasad Tadepalli. Learning first-order acyclic Horn programs from entailment. *ILP*, pages 23–37, 1998.

Mathieu Serrurier and Henri Prade. Introducing possibilistic logic in ILP for dealing with exceptions. *Artif. Intell.*, 171(16-17):939–950, 2007.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

Osamu Watanabe. A formal study of learning via queries. In Michael S. Paterson, editor, *Automata, Languages and Programming*, pages 139–152, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

Certification of Iterated Belief Changes via Model Checking and its Implementation

Kai Sauerwald, **Philip Heltweg**, **Christoph Beierle**
FernUniversität in Hagen, 58084 Hagen, Germany
{kai.sauerwald,christoph.beierle}@fernuni-hagen.de, pheltweg@gmail.com

Abstract

Iterated belief change investigates principles for changes on epistemic states and their representational groundings. A common realisation of epistemic states are total preorders over possible worlds. In this paper, we consider the problem of certifying whether an operator over total preorders satisfies a given postulate. We introduce the first-order fragment $\mathrm{FO}^{\mathrm{TPC}}$ for expressing belief change postulates and present a way to encode information on changes into an $\mathrm{FO}^{\mathrm{TPC}}$-structure. As a result, the question of whether a belief change fulfils a postulate becomes a model checking problem. We present *Alchourron*, an implementation of our approach, consisting of an extensive Java library, and also of a web interface, which suits didactic purposes and experimental studies.

1 Introduction

A fundamental problem for intelligent agents is adapting their world-view to potentially new and conflicting information. Iterated belief change discusses properties of operators that model transition of currently held beliefs under newly received information. The field has a large body of literature with differentiated results for a variety of different types of operations, e.g., revision (Darwiche and Pearl 1997; Booth, Meyer, and Wong 2006), contraction (Hild and Spohn 2008; Konieczny and Pino Pérez 2017; Sauerwald, Kern-Isberner, and Beierle 2020), expansion, the area of non-prioritized change (Konieczny and Pino Pérez 2008; Booth et al. 2014; Schwind and Konieczny 2020) and many more (Schwind, Konieczny, and Marquis 2018).

The research on (iterated) belief change is focussed on propositional logic (but not limited to). Often, total preorders over interpretations (Darwiche and Pearl 1997; Konieczny and Pino Pérez 2008; Booth et al. 2014; Schwind, Konieczny, and Marquis 2018; Sauerwald, Kern-Isberner, and Beierle 2020; Schwind and Konieczny 2020; Konieczny and Pino Pérez 2017; Schwind and Konieczny 2020) or refinements thereof (Hild and Spohn 2008; Booth, Meyer, and Wong 2006) are considered as a representation formalism for epistemic states.

A common aspect of many approaches in the area of iterated belief change is that the type of an operator class is given by syntactic postulates, constraining how to change, and that representation theorems show, which semantic postulates exactly reconstruct that class of operations in the realm of total preorders. The typical structure of postulates, regardless of whether there are syntactic or semantic postulates; is that they focus on a single (but arbitrary) epistemic state and constrain the result of subsequent changes on that state. When total preorders are considered as epistemic states, then very often, the so-called faithfulness condition and a representation theorem connects the syntactic viewpoint with the semantic perspective, e.g. (Darwiche and Pearl 1997).

Given the large variety of different postulates and types of operations, it is tedious and cumbersome to check manually whether a given specific change satisfies a certain postulate, or to decide whether the change falls into a certain category of type of operation.

This leads to the general problem of checking whether a belief change operator or a singular change satisfies a given syntactic or semantic postulate, which we call the *certification problem*. The certification problem got not much attention, notable exceptions are results about the complexity for specific operations (Nebel 1998; Liberatore 1997; Schwind et al. 2020) and results about inexpressibility (Turán and Yaggie 2015). Furthermore, there seems to be no implementation for the certification problem for the area of iterated belief change.

In this paper, we propose an approach to grasp the certification problem for the case where total preorders are used as epistemic states and provide an implementation. The approach consists of defining the first-order fragment $\mathrm{FO}^{\mathrm{TPC}}$, which is meant as a language for semantic postulates. To focus on semantic postulates seems to be only a minor restriction, as, given the many representation theorems, many syntactic postulates are known to be expressible by semantic postulates in the total preorder realm. Second, we describe how an $\mathrm{FO}^{\mathrm{TPC}}$-structure can be constructed for a belief change operator and for a singular belief change, respectively. The certification problem then becomes a first-order model-checking problem. Third, we present an implementation of the approach, which is publically available on the web.

2 Belief Change on Epistemic States

Let $\mathcal{L}$ be a propositional language over a finite signature of propositional variables Σ, and Ω its corresponding set of interpretations. Following the framework of Darwiche and Pearl (Darwiche and Pearl 1997), we deal with belief changes over epistemic states and propositions. An epistemic

Predicate	Intended meaning	Exemplary appearance
$Mod(w, x)$	w is a model of x	$\omega \in \mathrm{Mod}(\Psi), \omega \in \mathrm{Mod}(\alpha)$
$LessEQ(w_1, w_2, e)$	$w_1 \leqslant w_2$ in e	$\omega_1 \leqslant_\Psi \omega_2$
$Int(w)$	w is an interpretation	$\omega \in \Omega$
$ES(e)$	e is an epistemic state	$\Psi \in \mathcal{E}$
$Form(a)$	a is a formula	$\alpha \in \mathcal{L}$
Function	Intended meaning	Exemplary appearance
$op(e_0, a)$	$op(e_0, a)$ is a result of changing e_0 by a	$\Psi * \alpha = \Psi'$
$or(a, b)$	propositional disjunction	$\mathrm{Bel}(\Psi \circ (\alpha \vee \beta)) = \ldots$
$not(a)$	propositional negation	$\neg\alpha \notin \mathrm{Bel}(\Psi \circ \alpha)$

Figure 1: Allowed predicates and functions symbols in $\mathrm{FO}^{\mathrm{TPC}}$, their intended meaning and how they are typically formulated in belief change literature.

state is an abstract entity from a set $\mathcal{E}$, where each $\Psi \in \mathcal{E}$ is equipped with a deductively closed set $\mathrm{Bel}(\Psi)$. A belief change operator is a function $\circ : \mathcal{E} \times \mathcal{L} \to \mathcal{E}$. In this paper, we only consider operators satisfying the following syntax-independence condition for each $\Psi \in \mathcal{E}$ and $\alpha, \beta \in \mathcal{L}$:

(sAGM5es*) if $\alpha \equiv \beta$, then $\Psi \circ \alpha = \Psi \circ \beta$

Here (sAGM5es*) is a stronger version of the *extensionality* postulate from the revision approach by Alchourrón, Gärdenfors and Makinson (1985) (AGM).

The framework by Darwiche and Pearl is different from the classical setup for belief revision theory by Alchourrón, Gärdenfors and Makinson (1985), where deductively closed sets (called belief sets) are used as states (Fermé and Hansson 2011). However, the richer structure of epistemic states is necessary to include the information required to capture the change strategy of iterative belief change (Darwiche and Pearl 1997).

There are many possible instantiations of $\mathcal{E}$; however, we will stick here to the very common one by total preorders. More precisely, we consider total preorders over Ω that fulfil the so-called faithfulness condition (Katsuno and Mendelzon 1992; Darwiche and Pearl 1997), stating that the minimal elements of each total preorder $\leqslant = \Psi \in \mathcal{E}$ are exactly the models of $\mathrm{Bel}(\Psi)$, i.e., $\mathrm{Mod}(\mathrm{Bel}(\Psi)) = \min(\Omega, \leqslant)$. Thus, in the scope of this paper, each total preorder $\leqslant \in \mathcal{E}$ is assumed to entirely describe an epistemic state.

3 Problem Statement

Postulates are central objects in the area of (iterative) belief change and are grouped together to define classes of belief change operators in a descriptive way. The problem we address is to check whether a given operator satisfies a postulate, i.e., belongs to a class of change operators specified by postulates. We call this particular problem the certification problem (which could be considered as a generalisation of the revision problem (Nebel 1998)):

CERTIFICATION-PROBLEM
Given: A belief change operator $\circ$ and a postulate P
Question: Does $\circ$ satisfy the postulate P?

Clearly, information about a whole belief change operator is available or even finitely representable only in few application scenarios. This gives rise to several sub-problems depending on how much information of the particular operator is known. Apart from the full operator $\circ$, we consider the certification of the following cases:

- A singular belief change from Ψ to Ψ' by α, i.e.: Does $\Psi \circ \alpha = \Psi'$ hold?
- A sequence of belief changes $\Psi_1 \circ \alpha_1 = \Psi_2$, and $\Psi_2 \circ \alpha_2 = \Psi_3$, and ...
- All singular belief changes on a state Ψ, i.e. the set $\{(\Psi_1, \alpha, \Psi_2) \in \circ \mid \Psi = \Psi_1\}$

In the next section we present a model-checking based formalisation of the CERTIFICATION-PROBLEM.

4 The Approach

In belief change, postulates are usually described by common mathematical language, which is close to (first-order) predicate logic. In the following, we use the toolset of first-order logic to formalise the CERTIFICATION-PROBLEM as a first-order model-checking problem.

Language for Postulates As an initial study, we considered several postulates from literature on iterated belief change, e.g. (Darwiche and Pearl 1997; Booth and Meyer 2006; Jin and Thielscher 2007; Booth 2002; Nayak, Pagnucco, and Peppas 2003), and selected the most common predicates and functions used. We compiled them into a fragment of first-order logic with equality over a fixed set of predicates and function symbols[1], denoted by $\mathrm{FO}^{\mathrm{TPC}}$ (Total Preorder Change), with the intention to describe changes over total preorders. Figure 1 summarises the permitted symbols and describes only the minimal required set.

Several common predicates and functions used in postulates are expressible by the means of $\mathrm{FO}^{\mathrm{TPC}}$ by employing this minimal set, e.g. logical entailment, semantic equality, the strict part of a total preorder, checking whether a formula has no model, etc. For a specific example, consider the following:

$$LogImpl(x, y) {:=} \forall w . Int(w) \to (Mod(w, x) \to Mod(w, y))$$

[1] Note that we could also use a fragment of many-sorted first-order logic. However, some predicates are "overloaded" in respect to sorts.

Universe	$U^{\mathcal{A}_C} = \Omega \cup \{\Psi_0, \Psi_1\} \cup \mathcal{P}(\Omega)$		
Predicates			
$Mod^{\mathcal{A}_C}$	$= \{(\omega, x) \mid x \in \mathcal{P}(\Omega) \cup \{\Psi_0, \Psi_1\}, \omega \in \mathrm{Mod}(x)\}\}$		
$Int^{\mathcal{A}_C}$	$= \Omega$		
$ES^{\mathcal{A}_C}$	$= \{\Psi_0, \Psi_1\}$		
$Form^{\mathcal{A}_C}$	$= \mathcal{P}(\Omega)$		
$LessEQ^{\mathcal{A}_C}$	$= \{(\omega_1, \omega_2, \Psi_i) \mid \omega_1 \leqslant_{\Psi_i} \omega_2\}$		
Functions			
$or^{\mathcal{A}_C}$	$= \lambda\alpha_1, \alpha_2 .\, \alpha_1 \cup \alpha_2$	$e_0^{\mathcal{A}_C}$	$= \Psi_0$
$not^{\mathcal{A}_C}$	$= \lambda\alpha_1 .\, \Omega \setminus \alpha_1$	$a^{\mathcal{A}_C}$	$= \mathrm{Mod}(\alpha)$
$op^{\mathcal{A}_C}$	$= (\{(\Psi, \beta, \Psi) \mid \beta \in \mathcal{P}(\Omega), \Psi \in \{\Psi_0, \Psi_1\}\} \setminus \{(\Psi_0, \alpha, \Psi_0\}) \cup \{(\Psi_0, \alpha, \Psi_1)\}$		

Figure 2: Structure $\mathcal{A}_C$, encoding a singular change $C = (\Psi_0, \alpha, \Psi_1)$

where $LogImpl(x, y)$ describes that x logically implies y.

For illustration, we consider some aspects about belief change postulates. First, belief change postulates are typically formulated with a locality aspect; every postulate focusses an initial state and a change formula α, describing a condition for this change. As prominent examples, the following postulates are an excerpt of the AGM revision postulates (Alchourrón, Gärdenfors, and Makinson 1985):

(AGM2*) $\alpha \in \mathrm{Bel}(\Psi \circ \alpha)$

(AGM7*) $\mathrm{Bel}(\Psi \circ (\alpha \wedge \beta)) \subseteq Cn(\mathrm{Bel}(\Psi \circ \alpha) \cup \{\beta\})$

In $\mathrm{FO}^{\mathrm{TPC}}$, we address this by reserving e_0 and a as special terms, where e_0 denotes the initial state and a denotes the formula representing the new information.

Postulates for (iterated) belief change typically come in two fashions: *Semantic postulates* describe changes in a semantic domain, such as faithful total preorders. For example, consider the following postulate:

(CR1) if $\omega_1, \omega_2 \in \mathrm{Mod}(\alpha)$, then $\omega_1 \leqslant_\Psi \omega_2 \Leftrightarrow \omega_1 \leqslant_{\Psi \circ \alpha} \omega_2$

This could be expressed in $\mathrm{FO}^{\mathrm{TPC}}$ by the following formula $\varphi_{\text{(CR1)}}$:

$$\begin{aligned} \varphi_{\text{(CR1)}} = {} & \forall w1, w2. \\ & (Int(w_1) \wedge Int(w_2) \wedge ES(e_0) \wedge Form(a)) \\ & \to (LessEQ(w_1, w_2, e_0) \\ & \quad \leftrightarrow LessEQ(w_1, w_2, op(e_0, a))) \end{aligned} \tag{1}$$

On the other hand, *syntactic postulates* describe changes of $\mathrm{Bel}(\Psi)$. Aside of the AGM revision postulates, prominent examples are the Darwiche-Pearl postulates for revision (Darwiche and Pearl 1997) such as:

(DP1) if $\beta \models \alpha$, then $\mathrm{Bel}(\Psi \circ \alpha \circ \beta) = \mathrm{Bel}(\Psi \circ \beta)$

Several representation results in the literature show how syntactic and semantic postulates are interrelated. For instance, it is well-known that, given $\circ$ is an AGM revision operator, (CR1) holds if and only if (DP1) holds (Darwiche and Pearl 1997). Moreover, the semantic and syntactic domains are of course related, which allows us to describe many predicates used in the syntactic realm by semantic means. For example, a statement like $\mathrm{Bel}(\Psi \circ \alpha \circ \beta) = \mathrm{Bel}(\Psi \circ \beta)$ is expressible in $\mathrm{FO}^{\mathrm{TPC}}$ by employing the following formula:

$$\begin{aligned} Bel(a, e) := {} & (Form(a) \wedge ES(e)) \\ & \to (\forall x. Mod(x, a) \leftrightarrow Mod(x, e)) \end{aligned}$$

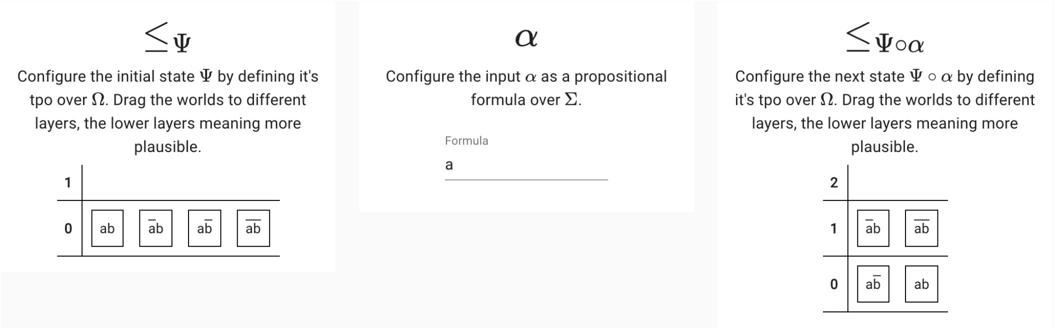
We describe now how objects like belief change operators, singular changes and so on are related to $\mathrm{FO}^{\mathrm{TPC}}$ formulas.

Encoding as Model-Checking Internally, we use the standard truth-functional semantics of first-order logic for $\mathrm{FO}^{\mathrm{TPC}}$. Therefore, we translate a belief change operator, respectively the known part of it, into a first-order structure.

The general idea is to define a structure $\mathcal{A}$ by the following pattern: The universe $U^{\mathcal{A}}$ consists of all propositional interpretations Ω, all formulas from $\mathcal{L}$ and all considered epistemic states from Ψ, i.e., the total preorders over Ω. We represent formulas by their models, i.e., by elements of[2] $\mathcal{P}(\Omega)$. The rationale is that, because of (sAGM5es*), the considered belief change operators are insensitive to syntactic differences. Additionally, predicates are interpreted in the straight-forward manner, e.g., Int is interpreted as all propositional interpretations, $Int^{\mathcal{A}} = \Omega$, and $LessEQ$ allows access to the total preorder Ψ of each epistemic state, $LessEQ^{\mathcal{A}} = \{(\omega_1, \omega_2, \Psi_i) \mid (\omega_1, \omega_2) \in \Psi_i\}$. Depending on whether a full change operator, a singular change, or another sub-problem is considered, some special treatment is necessary.

For instance, consider the signature $\Sigma = \{a, b\}$, yielding the interpretations $\Omega = \{ab, \overline{a}b, a\overline{b}, \overline{a}\overline{b}\}$. Moreover, consider the singular change $C = (\Psi_0, \alpha, \Psi_1)$, where $\Psi_0 = \leqslant_0$ is the total preorder treating every interpretation to be equally plausible, i.e., $ab =_0 a\overline{b} =_0 \overline{a}b =_0 \overline{a}\overline{b}$. Furthermore, let $\alpha = a$. The total preorder $\Psi_1 = \leqslant_1$ treats all a-models to be equally plausible, but prefers them over all non a-models, which are considered to be equally plausible, i.e. $ab =_1 a\overline{b} <_1 \overline{a}b =_1 \overline{a}\overline{b}$. We construct a structure $\mathcal{A}_C$ as follows: The universe is given by $U^{\mathcal{A}_C} = \Omega \cup \{\Psi_0, \Psi_1\} \cup \mathcal{P}(\Omega)$. The predicates and function symbols are interpreted according to Figure 2. The terms e_0 and a are interpreted as $e_0^{\mathcal{A}_C} = \Psi_0$ and $a^{\mathcal{A}_C} = \mathrm{Mod}(\alpha)$.

[2] $\mathcal{P}(\cdot)$ is the powerset function.



$\leq_\Psi$

Configure the initial state Ψ by defining it's tpo over Ω. Drag the worlds to different layers, the lower layers meaning more plausible.

1 |
0 | ab $\bar{a}b$ $a\bar{b}$ $\overline{ab}$

α

Configure the input α as a propositional formula over Σ.

Formula: a

$\leq_{\Psi\circ\alpha}$

Configure the next state $\Psi \circ \alpha$ by defining it's tpo over Ω. Drag the worlds to different layers, the lower layers meaning more plausible.

2 |
1 | $\bar{a}b$ $\overline{ab}$
0 | $a\bar{b}$ ab

Figure 3: Input fields for the change from Ψ_0 to Ψ_1 by $\alpha = a$ in Alchourron.

In summary, the CERTIFICATION-PROBLEM of whether C satisfies (CR1) is expressed as a model-checking problem for $\mathrm{FO}^{\mathrm{TPC}}$, i.e., a change C satisfies the postulate (CR1) if $\mathcal{A}_C \models \varphi_{(\mathrm{CR1})}$ holds, where $\varphi_{(\mathrm{CR1})}$ is the formula given in (1).

5 Implementation

We provide an implementation of the approach by combining independent, self-developed Java libraries. The approach is publicly accessible by a web-frontend called *Alchourron*[3], which expands on the previous work by Sauerwald and Haldimann (Sauerwald and Haldimann 2019). The currently available version allows the specification of a singular belief change using a browser-based client. First, the user decides on a propositional signature for the language of the belief change. Then a prior total preorder, an input formula, as well as the posterior total preorder is entered. Figure 3 illustrates the belief change input.

After specifying the change, Alchourron allows the user to check whether several preconfigured belief change postulates are satisfied. Optionally, a user can also enter her own postulate by defining a first-order formula using $\mathrm{FO}^{\mathrm{TPC}}$. Formulas are described in TPTP syntax (Sutcliffe 2017), e.g., the postulate (CR1) from Section 4 can be expressed as follows:

```
! [W1,W2] :
 ((int(W1) & int(W2) & mod(W1, A) & mod(W2,
    A))
=> (lesseq(W1, W2, E0)
   <=> lesseq(W1, W2, op(E0, A))))
```

Internally, Alchourron has a client-server architecture. The implementation is highly modularized, and we expect reusability of components for further projects. In particular, postulate checking via compilation into a model-checking problem as described in Section 4 is happening completely on the server side. Display of total preorders is provided by web components[4] that can also represent *ordinal conditional functions* (Spohn 1988), which for instance implement total preorders, but provide also more fine-grained representations of epistemic states. Our implementation of logic is an extensive institution-inspired implementation called *Logical Systems*[5], which allows representation and evaluation of a variety of different logics in a unified way. Preconfigured postulates are stored in TPTP syntax and parsed from there[6], mapping TPTP specified formula into our internal representation.

6 Summary and Future Work

We proposed $\mathrm{FO}^{\mathrm{TPC}}$, a first-order fragment to describe belief change postulates, complemented with a methodology to construct a finite structure for a belief change operator, employing total preorders as representation of epistemic states. With this toolset, the certification of belief change operators can be understood as a model-checking problem. We presented our implementation, which is available online[3], as a proof of concept for our approach for singular belief changes. In summary, we defined and formalized the certification problem and provide an implementation therefore.

While this is only the first proposal, we expect that this approach will be highly flexible regarding improvements and extensions. In particular, for future work we want to expand our approach to more complex representations of epistemic states. Moreover, we will work to improve the efficiency of the implementation.

[3] Visit: https://www.fernuni-hagen.de/wbs/alchourron/

[4] Heltweg, P.: Logic components, 2021. DOI: 10.5281/zenodo.4744650.

[5] Sauerwald, K.: Logical Systems, 2021, github.com/Landarzar/logical-systems.

[6] Steen, A.: Scala TPTP parser, 2021. DOI: 10.5281/zenodo.4672395.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.* 50(2):510–530.

Booth, R., and Meyer, T. A. 2006. Admissible and restrained revision. *J. Artif. Intell. Res.* 26:127–151.

Booth, R.; Fermé, E. L.; Konieczny, S.; and Pino Pérez, R. 2014. Credibility-limited improvement operators. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 123–128. IOS Press.

Booth, R.; Meyer, T. A.; and Wong, K. 2006. A bad day surfing is better than a good day working: How to revise a total preorder. In Doherty, P.; Mylopoulos, J.; and Welty, C. A., eds., *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, 230–238. AAAI Press.

Booth, R. 2002. On the logic of iterated non-prioritised revision. In Kern-Isberner, G.; Rödder, W.; and Kulmann, F., eds., *Conditionals, Information, and Inference, International Workshop, WCII 2002, Hagen, Germany, May 13-15, 2002, Revised Selected Papers*, volume 3301 of *Lecture Notes in Computer Science*, 86–107. Springer.

Darwiche, A., and Pearl, J. 1997. On the logic of iterated belief revision. *Artificial Intelligence* 89:1–29.

Fermé, E. L., and Hansson, S. O. 2011. AGM 25 years - twenty-five years of research in belief change. *J. Philosophical Logic* 40(2):295–331.

Hild, M., and Spohn, W. 2008. The measurement of ranks and the laws of iterated contraction. *Artif. Intell.* 172(10):1195–1218.

Jin, Y., and Thielscher, M. 2007. Iterated belief revision, revised. *Artif. Intell.* 171(1):1–18.

Katsuno, H., and Mendelzon, A. O. 1992. Propositional knowledge base revision and minimal change. *Artif. Intell.* 52(3):263–294.

Konieczny, S., and Pino Pérez, R. 2008. Improvement operators. In Brewka, G., and Lang, J., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, 177–187. AAAI Press.

Konieczny, S., and Pino Pérez, R. 2017. On iterated contraction: syntactic characterization, representation theorem and limitations of the Levi identity. In *Scalable Uncertainty Management - 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings*, volume 10564 of *Lecture Notes in Artificial Intelligence*. Springer.

Liberatore, P. 1997. The complexity of iterated belief revision. In Afrati, F. N., and Kolaitis, P. G., eds., *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, volume 1186 of *Lecture Notes in Computer Science*, 276–290. Springer.

Nayak, A. C.; Pagnucco, M.; and Peppas, P. 2003. Dynamic belief revision operators. *Artif. Intell.* 146(2):193–228.

Nebel, B. 1998. *How Hard is it to Revise a Belief Base?* Dordrecht: Springer Netherlands. 77–145.

Sauerwald, K., and Haldimann, J. 2019. WHIWAP: checking iterative belief changes. In Beierle, C.; Ragni, M.; Stolzenburg, F.; and Thimm, M., eds., *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI & Kognition (KIK-2019), Kassel, Germany, September 23, 2019*, volume 2445 of *CEUR Workshop Proceedings*, 14–23. CEUR-WS.org.

Sauerwald, K.; Kern-Isberner, G.; and Beierle, C. 2020. A conditional perspective for iterated belief contraction. In Giacomo, G. D.; Catalá, A.; Dilkina, B.; Milano, M.; Barro, S.; Bugarín, A.; and Lang, J., eds., *ECAI 2020 - 24nd European Conference on Artificial Intelligence, August 29th - September 8th, 2020, Santiago de Compostela, Spain*, 889–896. IOS Press.

Schwind, N., and Konieczny, S. 2020. Non-Prioritized Iterated Revision: Improvement via Incremental Belief Merging. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, 738–747.

Schwind, N.; Konieczny, S.; Lagniez, J.; and Marquis, P. 2020. On computational aspects of iterated belief change. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1770–1776.

Schwind, N.; Konieczny, S.; and Marquis, P. 2018. On belief promotion. In Thielscher, M.; Toni, F.; and Wolter, F., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, 297–307. AAAI Press.

Spohn, W. 1988. Ordinal conditional functions: a dynamic theory of epistemic states. In Harper, W., and Skyrms, B., eds., *Causation in Decision, Belief Change, and Statistics, II.* Kluwer Academic Publishers. 105–134.

Sutcliffe, G. 2017. The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning* 59(4):483–502.

Turán, G., and Yaggie, J. 2015. Characterizability in belief revision. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 3236–3242.

Revising Ontologies via Models: The $\mathcal{ALC}$-formula Case

Jandson S. Ribeiro,[1] Ricardo Guimarães,[2] Ana Ozaki[2]
[1]Universität Koblenz-Landau, Germany
[2]University of Bergen, Norway
jandson@uni-koblenz.de, ricardo.guimaraes@uib.no, ana.ozaki@uib.no

Abstract

In this work, we propose a new paradigm for Belief Change: we require that new pieces of information are represented as finite models, while the agent's body of knowledge is represented as a finite set of formulae. When the logic does not ensure that every set of models has a corresponding finite base, as it is the case of most of description logics (DLs), the standard rationality postulates of Belief Change cannot be captured. We define new Belief Change operations for this setting, and we identify the rationality postulates that emerge due to the finite representability requirement. Moreover, we instantiate our approach to the case of the description logic $\mathcal{ALC}$-formula.

1 Introduction

Processing how a rational agent should autonomously modify its current knowledge base in response to new pieces of information is the object of study of Belief Change (Hansson 1999) which is tightly connected to non-monotonic systems (Makinson and Gärdenfors 1989). The most interesting and challenging situations emerge when the new incoming information is in conflict with the agent's current knowledge base. In this case, the agent should minimally remove only the beliefs that are in conflict with the incoming information. This principle of minimal change is captured in Belief Change via rationality postulates that dictate the minimal properties of a rational change. The main paradigms of Belief Change assume that an agent's knowledge base is represented as a set of formulae expressed in some underlying logic, such as classical propositional logics; while incoming pieces of information are represented as formulae of the same underlying logic. This kind of representation system can be inconvenient in scenarios where the incoming pieces of information should be represented in other formats as, for instance, a finite model. This is the case in the paradigm of Learning from Interpretations (De Raedt 1997), where a formula needs to be created or changed to either incorporate or block certain interpretations (here called 'models') in a finite way. Example 1 illustrates the intuition behind using models as input.

Example 1. Suppose that a system, which serves a university, uses an internal logical representation of the domain with an open world behaviour and unique names. Let $\mathcal{B}$ be its current representation:

$$\mathcal{B} = \{\mathsf{Professors} : \{\mathsf{Mary}\}, \mathsf{Courses} : \{\mathsf{DL}, \mathsf{AI}\}, \\ \{\mathsf{teaches} : \{(\mathsf{Mary}, \mathsf{AI}), (\mathsf{Mary}, \mathsf{DL})\}\} .$$

Assume that a user finds mistakes in the course schedule which is caused by the wrong information that Mary teaches the DL course. The user may lack knowledge to define the issue formally. An alternative is to provide the user with an interface where one can specify, for instance, that the model $M = \{\mathsf{Professors} = \{\mathsf{Mary}\}, \mathsf{Courses} = \{\mathsf{DL}, \mathsf{AI}\}, \mathsf{teaches} = \{(\mathsf{Mary}, \mathsf{AI})\}\}$, should be accepted (in this model Mary does not teach the DL course). Given this input, the system should repair itself (semi-)automatically.

In this work, we introduce a new paradigm for Belief Change. We consider the case in which incoming pieces of information are represented as finite models, while the current knowledge of the agent is represented as a finite set of formulae. We impose the finite representability requirement because, in Computer Science, resources such as memory, are limited and the knowledge of an agent should to be represented finitely. In classical propositional logic, representing new information as a set of models is straightforward; but this not so in some more expressive non-classical logics, such as Description Logics (DLs). This problem emerges because in many DLs, not every set of models can be represented with a finite formula (which is only satisfied by such models). In other words, not every set of models has a *finite base*. As an alternative, a 'close' finitely representable knowledge base needs to be chosen instead. We identify these new postulates that arise with this requirement, and we show the belief change operations that they characterise. Also, we analyse the case of $\mathcal{ALC}$-formula using quasimodels as a way to define new belief change operations. This logic satisfies properties which facilitate the design of these operations and it is close to $\mathcal{ALC}$: a well-studied DL.

In Section 2, we briefly review some basic concepts from Belief Change and we detail the new belief change paradigm we propose. In Section 3, we investigate the new paradigm of Belief Change for the $\mathcal{ALC}$-formulae case. We identify the respective representation theorems. In Section 4, we

A version to this paper was accepted at the 34th International Workshop on Description Logics (DL 2021).

highlight studies which share similarities with our proposal and we conclude in Section 5.

2 Belief Change

2.1 The Classical Setting

Belief Change (Alchourrón, Gärdenfors, and Makinson 1985; Hansson 1999) studies the problem of how an agent should modify its knowledge in light of new information. In the original paradigm of Belief Change, the AGM theory, an agent's body of knowledge is represented as a set of formulae closed under logical consequence, called a *belief set*, and the new information is represented as a single *formula*. In the propositional classical logics, every belief set can be finitely represented by a finite set of formulae, called a *belief base* (Hansson 1999). In the AGM paradigm, when confronted with an information φ, an agent might modify its current belief set $\mathcal{B}$ in three ways:

When modifying its body of knowledge an agent should rationally modify its beliefs conserving most of its original beliefs. This principle of minimal change is captured in Belief Change via sets of rationality postulates. Each of the three operations (expansion, contraction and revision) presents its own set of rationality postulates which characterize precisely different classes of belief change constructions. The AGM paradigm was initially proposed for classical logics that satisfy specific requirements, dubbed AGM assumptions, among them *taskianicity, compactness* and deduction. See (Flouris 2006; Ribeiro 2013) for a complete list of the AGM assumptions and a discussion on the topic. Recently, efforts have been applied to extend Belief Change to logics that do not satisfy such assumptions. For instance, logics that are not closed under classical negation of formulae (such as is the case for most DLs) (Ribeiro 2013; Ribeiro and Wassermann 2014), and temporal logics and logics without compactness (Ribeiro, Nayak, and Wassermann 2018, 2019b,a).

2.2 Changing Finite Bases by Models

In this work, unlike the standard representation methods in Belief Change, we consider that an incoming piece of information is represented as a finite model. Belief Change operations defined in this format will be called model change operations. Recall that a model M is simply a structure used to give semantics to an underlying logic language. The set of all possible models is given by $\mathfrak{M}$. Moreover, we assume a semantic system that, for each set of formulae $\mathcal{B}$ of the language $\mathcal{L}$ gives a set of models $\mathrm{Mod}(\mathcal{B}) := \{M \in \mathfrak{M} \mid \forall\varphi \in \mathcal{B} : M \models \varphi\}$. Let $\mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ denote the set of all finite bases in $\mathcal{L}$. We also say that a set of models $\mathbb{M}$ is *finitely representable in* $\mathcal{L}$ if there is a finite base $\mathcal{B} \in \mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ such that $\mathrm{Mod}(\mathcal{B}) = \mathbb{M}$. Additionally, if for all $\varphi \in \mathcal{L}$ it holds that $M \models \varphi$ iff $M' \models \varphi$ then we write $M \equiv^{\mathcal{L}} M'$. We also define $[M]^{\mathcal{L}} := \{M' \in \mathfrak{M} \mid M' \equiv^{\mathcal{L}} M\}$.

The first model change operation we introduce is model contraction, which eliminates one of the models of the current base (which in Section 3 is instantiated as an ontology). Model contraction is akin to a belief expansion, where a formula is added to the belief set or base, reducing the set of models accepted. The counterpart operation, model expansion, changes the base to include a new model. This relates to belief contraction, in which a formula is removed, and thus more models are seen as plausible.

We write rationality postulates for an ideal contraction over finitely representable theories, where the incoming piece of information represented as a finite model, instead of a single formula.

Definition 2 (Model Contraction)**.** Let $\mathcal{L}$ be a language. A function $\mathrm{con} : \mathcal{P}_{\mathrm{fin}}(\mathcal{L}) \times \mathfrak{M} \mapsto \mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ is a *finitely representable model contraction function* iff for every $\mathcal{B} \in \mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ and $M \in \mathfrak{M}$ it satisfies the following postulates:

(success) $M \notin \mathrm{Mod}(\mathrm{con}(\mathcal{B}, M)) = \emptyset$,
(inclusion) $\mathrm{Mod}(\mathrm{con}(\mathcal{B}, M)) \subseteq \mathrm{Mod}(\mathcal{B})$,
(retainment) if $M' \in \mathrm{Mod}(\varphi) \setminus \mathrm{Mod}(\mathrm{con}(\varphi, M))$ then $M' \equiv^{\mathcal{L}} M$,
(extensionality) $\mathrm{con}(\mathcal{B}, M) = \mathrm{con}(\mathcal{B}, M')$, if $M \equiv^{\mathcal{L}} M'$.

We might also need to add a model to the set of models of the current base. This addition relates to classical contractions in Belief Change, which *reduces* the belief base.

Definition 3 (Model Expansion)**.** Let $\mathcal{L}$ be a language. A function $\mathrm{ex} : \mathcal{P}_{\mathrm{fin}}(\mathcal{L}) \times \mathfrak{M} \mapsto \mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ is a *finitely representable model expansion* iff for every $\mathcal{B} \in \mathcal{P}_{\mathrm{fin}}(\mathcal{L})$ and $M \in \mathfrak{M}$ it satisfies the postulates:

(success) $M \in \mathrm{Mod}(\mathrm{ex}(\mathcal{B}, M))$,
(persistence) $\mathrm{Mod}(\mathcal{B}) \subseteq \mathrm{Mod}(\mathrm{ex}(\mathcal{B}, M))$,
(vacuity) $\mathrm{Mod}(\mathrm{ex}(\mathcal{B}, M)) = \mathrm{Mod}(\mathcal{B})$, if $M \in \mathrm{Mod}(\mathcal{B})$,
(extensionality) $\mathrm{ex}(\mathcal{B}, M) = \mathrm{ex}(\mathcal{B}, M')$, if $M \equiv^{\mathcal{L}} M'$.

Definition 4. Let $\mathcal{L}$ be a language and Cn a Tarskian consequence operator defined over $\mathcal{L}$. Also let $\mathfrak{M}$ be a fixed set of models. We say that a triple $(\mathcal{L}, \mathrm{Cn}, \mathfrak{M})$ is an *ideal logical system* if the following holds.

- For every $\mathcal{B} \subseteq \mathcal{L}$ and $\varphi \in \mathcal{L}$, $\mathcal{B} \models \varphi$ (i.e. $\varphi \in \mathrm{Cn}(\mathcal{B})$) iff $\mathrm{Mod}(\mathcal{B}) \subseteq \mathrm{Mod}(\varphi)$.
- For each $\mathbb{M} \subseteq \mathfrak{M}$ there is a finite set of formulae $\mathcal{B}$ such that $\mathrm{Mod}(\mathcal{B}) = \mathbb{M}$.

Given the conditions in Definition 4, we can define a function FR such that $\mathrm{Mod}(\mathrm{FR}(\mathbb{M})) = \mathbb{M}$. Then, we can define model contraction as $\mathrm{con}(\mathcal{B}, M) = \mathrm{FR}(\mathrm{Mod}(\mathcal{B}) \setminus [M]^{\mathcal{L}})$ and expansion as $\mathrm{ex}(\mathcal{B}, M) = \mathrm{FR}(\mathrm{Mod}(\mathcal{B}) \cup [M]^{\mathcal{L}})$. An example that fits these requirements is to consider classical propositional logic with a finite signature Σ, together with its usual consequence operator and models. In this situation, we can define FR as follows:

$$\mathrm{FR}(\mathbb{M}) = \bigvee_{M \in \mathbb{M}} \left(\bigwedge_{a \in \Sigma | M \models a} a \wedge \bigwedge_{a \in \Sigma | M \models \neg a} \neg a \right).$$

Next, we show that the construction proposed with FR has the properties stated in Definitions 2 and 3.

Theorem 5. Let $(\mathcal{L}, \mathrm{Cn}, \mathfrak{M})$ be an ideal logical system as in Definition 4. Then $\mathrm{iCon}(\mathcal{B}, M) := \mathrm{FR}\left(\mathrm{Mod}(\mathcal{B}) \setminus [M]^{\mathcal{L}}\right)$ satisfies the postulates in Definition 2.

Proof. Success and inclusion are trivially satisfied. If $M \equiv^{\mathcal{L}} M'$, then $[M]^{\mathcal{L}} = [M']^{\mathcal{L}}$, thus extensionality is satisfied. Also, if $M' \in \mathrm{Mod}(\varphi) \setminus \mathrm{Mod}(\mathrm{iCon}(\varphi, M))$ then $M' \in [M]^{\mathcal{L}}$, hence the operation satisfies retainment. □

Theorem 6. Let $(\mathcal{L}, \mathrm{Cn}, \mathfrak{M})$ be an ideal logical system as in Definition 4. Then $\mathrm{iExp}(\mathcal{B}, M) := \mathrm{FR}\left(\mathrm{Mod}(\mathcal{B}) \cup [M]^{\mathcal{L}}\right)$ satisfies the postulates in Definition 3.

Proof. Success, vacuity and persistence are trivially satisfied. Extensionality also holds because whenever $M \equiv^{\mathcal{L}} M'$ we have then $[M]^{\mathcal{L}} = [M']^{\mathcal{L}}$. □

A revision operation incorporates new formulae, and removes potential conflicts in behalf of consistency. In our setting, incorporating information coincides with model contraction which could lead to an inconsistent belief state. In this case, model revision could be interpreted as a conditional model contraction: in some cases the removal might be rejected to preserve consistency. We leave the study on revision as a future work.

3 The case of $\mathcal{ALC}$-formula

The logic $\mathcal{ALC}$-formula corresponds to the DL $\mathcal{ALC}$ enriched with boolean operators over $\mathcal{ALC}$ axioms. As discussed in Section 2.2, in finite representable logics, such as the classical propositional logics, we can easily add and remove models while keeping the representation finite. For $\mathcal{ALC}$-formula, however, it is not possible to uniquely add or remove a new model M since, for instance, the language does not distinguish quantities (e.g., a model M and another model that has two duplicates of M).

Even if quantities are disregarded and our input is a class of models indistinguishable by $\mathcal{ALC}$-formulae, there are sets of formulae in this language that are not finitely representable. As for instance in the following infinite set: $\{C \sqsubseteq \exists r^n.\top \mid n \in \mathbb{N}^{>0}\}$, where $\exists r^{n+1}.\top$ is a shorthand for $\exists r.(\exists r^n.\top)$ and $\exists r^1.\top := \exists r.C$. As a workaround for the $\mathcal{ALC}$-formula case, we propose a new strategy based on the translation of $\mathcal{ALC}$-formulae into DNF.

3.1 $\mathcal{ALC}$-formulae and Quasimodels

Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be countably infinite and pairwise disjoint sets of concept names, role names, and individual names, respectively. *$\mathcal{ALC}$ concepts* are built according to the rule: $C ::= A \mid \neg C \mid (C \sqcap C) \mid \exists r.C$, where $A \in \mathsf{N_C}$. *$\mathcal{ALC}$-formulae* are defined as expressions ϕ of the form

$$\phi ::= \alpha \mid \neg(\phi) \mid (\phi \wedge \phi) \quad \alpha ::= C(a) \mid r(a, b) \mid (C = \top),$$

where C and D are concepts, $a, b \in \mathsf{N_I}$, and $r \in \mathsf{N_R}$[1]. Denote by $\mathsf{ind}(\varphi)$ the set of all individual names occurring in an $\mathcal{ALC}$-formula φ.

The semantics of $\mathcal{ALC}$-formulae and the definitions related to quasimodels are standard (Gabbay 2003, page 70). In what follows, we reproduce the essential definitions and results for this work. Let φ be an $\mathcal{ALC}$-formula. Let $\mathsf{f}(\varphi)$ and $\mathsf{c}(\varphi)$ be the set of all subformulae and subconcepts of φ closed under single negation, respectively.

A *concept type* for φ is a subset $\mathbf{c} \subseteq \mathsf{c}(\varphi)$ such that: $D \in \mathbf{c}$ iff $\neg D \notin \mathbf{c}$, for all $D \in \mathsf{c}(\varphi)$; and (2) $D \sqcap E \in \mathbf{c}$ iff $\{D, E\} \subseteq \mathbf{c}$, for all $D \sqcap E \in \mathsf{c}(\varphi)$. A *formula type* for φ is a subset $\mathbf{f} \subseteq \mathsf{f}(\varphi)$ such that: (1) $\phi \in \mathbf{f}$ iff $\neg\phi \notin \mathbf{f}$, for all $\phi \in \mathsf{f}(\varphi)$; and (2) $\phi \wedge \psi \in \mathbf{f}$ iff $\{\phi, \psi\} \subseteq \mathbf{f}$, for all $\phi \wedge \psi \in \mathsf{f}(\varphi)$.

We may omit 'for φ' if this is clear from the context. A *model candidate* for φ is a triple $(T, o, \mathbf{f})$ such that T is a set of concept types, o is a function from $\mathsf{ind}(\varphi)$ to T, $\mathbf{f}$ a formula type, and $(T, o, \mathbf{f})$ satisfies the conditions: $\varphi \in \mathbf{f}$; $C(a) \in \mathbf{f}$ implies $C \in o(a)$; $r(a, b) \in \mathbf{f}$ implies $\{\neg C \mid \neg\exists r.C \in o(a)\} \subseteq o(b)$.

Definition 7 (Quasimodel)**.** A model candidate $(T, o, \mathbf{f})$ for φ is a *quasimodel* for φ if the following holds

- for every concept type $\mathbf{c} \in T$ and every $\exists r.D \in \mathbf{c}$, there is $\mathbf{c}' \in s$ such that $\{D\} \cup \{\neg E \mid \neg\exists r.E \in \mathbf{c}\} \subseteq \mathbf{c}'$;
- for every concept type $\mathbf{c} \in T$ and every concept C, if $\neg C \in \mathbf{c}$ then this implies $(C = \top) \notin \mathbf{f}$;
- for every concept C, if $\neg(C = \top) \in \mathbf{f}$ then there is $\mathbf{c} \in T$ such that $C \notin \mathbf{c}$;
- T is not empty.

Theorem 8 motivates the decision of using quasimodels to implement our operations for finite bases described in $\mathcal{ALC}$-formulae.

Theorem 8 (Theorem 2.27 (Gabbay 2003))**.** An $\mathcal{ALC}$-formula φ is satisfiable iff there is a quasimodel for φ.

3.2 $\mathcal{ALC}$-formulae in Disjunctive Normal Form

Any $\mathcal{ALC}$-formula can be translated into an equivalent (although potentially exponentially larger) $\mathcal{ALC}$-formula made of a disjunction of conjunctions of (possibly negated) atomic formulae. Let $\mathsf{S}(\varphi)$ be the set of all quasimodels for φ.

$$\text{We define } \varphi^{\dagger} = \bigvee_{(T, o, \mathbf{f}) \in \mathsf{S}(\varphi)} \Big(\bigwedge_{\alpha \in \mathbf{f}} \alpha \wedge \bigwedge_{\neg\alpha \in \mathbf{f}} \neg\alpha \Big),$$

where α is of the form $(C = \top), C(a), r(a, b)$.

Definition 9 ((Gabbay 2003))**.** Let $\mathcal{I}$ be an interpretation and φ an $\mathcal{ALC}$-formula formula. The quasimodel of $\mathcal{I}$ w.r.t. φ, symbols $\mathrm{qm}(\varphi, \mathcal{I}) = (T, o, \mathbf{f})$, is

- $T := \{\{C \in \mathsf{c}(\varphi) \mid x \in C^{\mathcal{I}}\} \mid x \in \Delta^{\mathcal{I}}\}$,
- $o(a) := \{C \in \mathsf{c}(\varphi) \mid a \in C^{\mathcal{I}}\}$, for all $a \in \mathsf{ind}(\varphi)$,
- $\mathbf{f} := \{\psi \in \mathsf{f}(\varphi) \mid \mathcal{I} \models \psi\}$.

Theorem 10. For every $\mathcal{ALC}$-formula φ: $\varphi \equiv \varphi^{\dagger}$.

In the next subsections, we present finite base model change operations for $\mathcal{ALC}$-formulae, i.e., functions from $\mathcal{L} \times \mathfrak{M} \mapsto \mathcal{L}$. We can represent the body of knowledge as a single formula because every finite belief base of $\mathcal{ALC}$-formulae can be represented by the conjunction of its elements. We use our translation to add models in a "minimal" way by *adding disjuncts*, while removing a model amounts to *removing disjuncts*.

[1] We may omit parentheses if there is no risk of confusion. The usual concept inclusions $C \sqsubseteq D$ can be expressed with $\top \sqsubseteq \neg C \sqcup D$ and $\neg C \sqcup D \sqsubseteq \top$, which is $(\neg C \sqcup D = \top)$.

3.3 Model Contraction for $\mathcal{ALC}$-formulae

We define model contraction for $\mathcal{ALC}$-formulae using the notion of quasimodels discussed previously and a correspondence between models and quasimodels.

We use the following operator, denoted μ, to define model contraction in Definition 11. Let φ be an $\mathcal{ALC}$-formula and let M be a model. Then,

$$\mu(\varphi, M) = \mathrm{ftypes}(\varphi) \setminus \{\mathbf{f}\}, \text{ where } qm(\varphi, M) = (T, o, \mathbf{f})$$

and $\mathrm{ftypes}(\varphi)$ is $\{\mathbf{f} \mid (T, o, \mathbf{f}) \in \mathsf{S}(\varphi)\}$. Let $lit(\mathbf{f})$ be the set of all literals in a formula type $\mathbf{f}$.

Definition 11. A *finite base model contraction function* is a function $\mathrm{con} : \mathcal{L} \times \mathfrak{M} \mapsto \mathcal{L}$ such that

$$\mathrm{con}(\varphi, M) = \begin{cases} \bigvee_{\mathbf{f} \in \mu(\varphi, M)} \bigwedge lit(\mathbf{f}), & \text{if } M \models \varphi,\ \mu(\varphi, M) \neq \emptyset \\ \bot & \text{if } M \models \varphi,\ \mu(\varphi, M) = \emptyset \\ \varphi & \text{otherwise.} \end{cases}$$

As we see later in this section, there are models M, M' such that $M \not\equiv^{\mathcal{L}} M'$ but our operations based on quasimodels cannot distinguish them. Given $\mathcal{ALC}$-formulae φ, ψ, we say that ψ is *in the language of the literals of* φ, written $\psi \in \mathcal{L}_{lit}(\varphi)$, if ψ is a boolean combination of the atoms in φ. Our operations partition the models according to this restricted language. We write $M \equiv^{\varphi} M'$ instead of $M \equiv^{\mathcal{L}_{lit}(\varphi)} M'$, and $[M]^{\varphi}$ instead of $[M]^{\mathcal{L}_{lit}(\varphi)}$ for conciseness.

Theorem 12. Let M be a model and φ an $\mathcal{ALC}$-formula. A finite base model function $\mathrm{con}^*(\varphi, M)$ is equivalent to $\mathrm{con}(\varphi, M)$ iff con^* satisfies:

(success) $M \not\models \mathrm{con}^*(\varphi, M)$,

(inclusion) $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) \subseteq \mathrm{Mod}(\varphi)$,

(atomic retainment): For all $\mathbb{M}' \subseteq \mathfrak{M}$, if $\mathrm{Mod}(\mathrm{con}^*(\mathcal{B}, M)) \subset \mathbb{M}' \subseteq \mathrm{Mod}(\mathcal{B}) \setminus [M]^{\varphi}$ then $\mathbb{M}'$ is not finitely representable in $\mathcal{ALC}$-formula.

(atomic extensionality) if $M' \equiv^{\varphi} M$ then

$$\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\mathrm{con}^*(\varphi, M')).$$

The postulate of *success* guarantees that M will be indeed relinquished, while *inclusion* imposes that no model will be gained during a contraction operation. Recall that in order to guarantee finite representability, it might be necessary to remove M jointly with other models. The postulates *atomic retainment* and *atomic extensionality* capture a notion of minimal change, dictating which such models are allowed to be removed together with M.

Example 13 illustrates how con works.

Example 13. Consider the following $\mathcal{ALC}$-formula and interpretation M:

$$\begin{aligned} \varphi :=& P(Mary) \wedge C(DL) \wedge C(AI) \wedge \\ & ((teaches(Mary, DL) \wedge \neg teaches(Mary, AI)) \vee \\ & (\neg teaches(Mary, DL) \wedge teaches(Mary, AI))) \end{aligned}$$

and $M = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}} = \{m, d, a\}$, $C^{\mathcal{I}} = \{d, a\}$, $P^{\mathcal{I}} = \{m\}$, $teaches^{\mathcal{I}} = \{(m, d)\}$, $Mary^{\mathcal{I}} = m$, $AI^{\mathcal{I}} = a$, and $DL^{\mathcal{I}} = d$. Assume we want to remove M from $\mathrm{Mod}(\varphi)$. As there only two equivalence classes in $\mathrm{Mod}(\varphi)$ w.r.t. $\mathcal{L}_{lit}(\varphi)$, $\mu(\varphi, M)$ will have single formula type, whose literals are: $P(Mary)$, $C(DL)$, $C(AI)$, $\neg teaches(Mary, DL)$ and $teaches(Mary, AI)$. Hence:

$$\begin{aligned} \mathrm{con}(\varphi, M) = \neg teaches(m, a) \wedge teaches(m, d) \wedge \\ C(d) \wedge C(a) \wedge P(m). \end{aligned}$$

3.4 Model Expansion in $\mathcal{ALC}$-formulae

In this section, we investigate model expansion for $\mathcal{ALC}$-formulae. Recall that we assume that a knowledge base is represented as a single $\mathcal{ALC}$-formula φ. Expansion consists in adding an input model M to the current knowledge base φ with the requirement that the new epistemic state can be represented also as a finite formula. To show that this strategy indeed guarantees finite representability, we start by defining a new expansion operation 'ex' as shown in Definition 14.

Definition 14. Given a quasimodel $(T, o, \mathbf{f})$, we write $\bigwedge(T, o, \mathbf{f})$ as a short-cut for $\bigwedge lit(\mathbf{f})$. A *finite base model expansion* is a function $\mathrm{ex} : \mathcal{L} \times \mathfrak{M} \to \mathcal{L}$ s.t.:

$$\mathrm{ex}(\varphi, M) = \begin{cases} \varphi & \text{if } M \models \varphi \\ \varphi \vee \bigwedge qm(\neg\varphi, M) & \text{otherwise.} \end{cases}$$

Example 15 illustrates how ex works.

Example 15. Consider the interpretation M from Example 13 and $\varphi := P(Mary) \wedge C(DL) \wedge C(AI) \wedge teaches(Mary, AI) \wedge \neg teaches(Mary, DL)$. Assume we want to add M to $\mathrm{Mod}(\varphi)$ and $qm(\neg\varphi, M) = (T, o, \mathbf{f})$. Thus, $lit(\mathbf{f}) = \{\neg teaches(m, a), teaches(m, d), C(d), C(a), P(m)\}$,

$$\begin{aligned} \mathrm{ex}(\varphi, M) = \varphi \vee \bigwedge lit(\mathbf{f}) = \varphi \vee \big(\neg teaches(m, a) \wedge \\ teaches(m, d) \wedge C(d) \wedge C(a) \wedge P(m)\big). \end{aligned}$$

The operation 'ex' maps a current knowledge base represented as a single formula φ and maps it to a new knowledge base that is satisfied by the input model M. The intuition is that 'ex' modifies the current knowledge base only if M does not satisfy φ. This modification is carried out by making a disjunct of φ with a formula ψ that is satisfied by M. This guarantees that M is present in the new epistemic state and that models of φ are not discarded. The trick is to find such an appropriate formula ψ which is obtained by taking the conjunction of all the literals within the quasimodel $qm(\neg\varphi, M)$. Here, the quasimodel needs to be centred on $\neg\varphi$ because $M \not\models \varphi$, and therefore it is not possible to construct a quasimodel based on M centred on φ.

Lemma 16. For every $\mathcal{ALC}$-formula φ and model M:

$$\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}.$$

Actually, any operation that adds precisely the equivalence class of M modulo the literals is equivalent to 'ex'.

Our next step is to investigate the rationality of 'ex*'. As expected adding the whole equivalence class of M with respect to $\mathcal{L}_{lit}(\varphi)$ does not come freely, and some rationality postulates are captured, while others are lost:

Theorem 17. Let M be a model and φ an $\mathcal{ALC}$-formula. A finite base model function $\mathrm{ex}^*(\varphi, M)$ is equivalent to $\mathrm{ex}(\varphi, M)$ iff ex^* satisfies:

(success) $M \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$.

(persistence): $\mathrm{Mod}(\varphi) \subseteq \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$.

(atomic temperance): For all $\mathbb{M}' \subseteq \mathfrak{M}$, if $\mathrm{Mod}(\varphi) \cup [M]^\varphi \subseteq \mathbb{M}' \subset \mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) \cup \{M\}$ then $\mathbb{M}'$ is not finitely representable in $\mathcal{ALC}$-formula.

(atomic extensionality) if $M' \equiv^\varphi M$ then

$$\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) = \mathrm{Mod}(\mathrm{ex}^*(\varphi, M')).$$

The postulates *success* and *persistence* come from requiring that M will be absorbed, and that models will not be lost during an expansion. The *atomic extensionality* postulate states that if two models satisfy exactly the same literals within φ, then they should present the same results. *Atomic temperance* captures a principle of minimality and guarantees that when adding M, the loss of information should be minimised. Precisely, the only formulae allowed to be given up are those that are incompatible with M modulo the literals of φ. Lemma 16 and Theorem 17 prove that the 'ex' operation is characterized by the postulates: *success, persistence, atomic temperance* and *atomic extensionality*.

4 Related Work

Belief bases have been used in the literature of Belief Change with two main purposes: as a finite representation of an agent's knowledge (Nebel 1991; Dixon and Wobcke 1993), and as a way of distinguishing an agent's knowledge explicitly (Hansson 1994). The syntactic connectivity in a knowledge base has a strong consequence of how an agent should modify its knowledge (Hansson 1999). This sensitivity to syntax is also present in Ontology Repair and Evolution. Classical approaches preserve the syntactic form of the ontology as much as possible (Kalyanpur 2006; Suntisrivaraporn 2009). However, these approaches may lead to drastic loss of information, as noticed by Hansson (1993). This problem has been studied in Belief Change for pseudo-contraction (Santos et al. 2018). In the same direction, Troquard et al. (2018) proposed the repair of DL ontologies by weakening axioms using refinement operators. Building on this study, Baader et al. (2018) devised the theory of *gentle repairs*, which also aims at keeping most of the information within the ontology upon repair. In fact, gentle repairs are closely related to pseudo-contractions (Matos et al. 2019).

Other remarkable works in Belief Change in which the body of knowledge is represented in a finite way include the formalisation of revision due to Katsuno and Mendelzon (1991) and the base-generated operations by Hansson (1996). In the former, Katsuno and Mendelzon (1991) formalise traditional belief revision operations using a single formula to represent the whole belief set. This is possible because they only consider finitary propositional languages. Hansson provides a characterisation of belief change operations over finite bases but restricted for logics which satisfy all the AGM-assumptions (such as propositional classical logic). Guerra and Wassermann (2019) develop operations for rational change where an agent's knowledge or behaviour is given by a Kripke model. They also provide two characterisations with AGM-style postulates.

5 Conclusion and Future Work

In this work, we have introduced a new kind of belief change operation: belief change via models. In our approach, an agent is confronted with a new piece of information in the format of a finite model, and it is compelled to modify its current epistemic state, represented as a single finite formula, either incorporating the new model, called model expansion; or removing it, called model contraction. The price for such finite representation is that the single input model cannot be removed or added alone, and some other models must be added or removed as well. As future work, we will investigate model change operations in other DLs, still taking into account finite representability. We will also explore the effects of relaxing some constraints on Belief Base operations, allowing us to rewrite axioms with different levels of preservation in the spirit of Pseudo-Contractions, Gentle Repairs, and Axiom Weakening.

Acknowledgements

Part of this work has been done in the context of CEDAS (Center for Data Science, University of Bergen, Norway). The first author is supported by the German Research Association (DFG), project number 424710479. Ozaki is supported by the Norwegian Research Council, grant number 316022.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic* 50(2): 510–530.

Baader, F.; Kriegel, F.; Nuradiansyah, A.; and Peñaloza, R. 2018. Making Repairs in Description Logics More Gentle. In *KR 2018*. AAAI Press.

De Raedt, L. 1997. Logical settings for concept-learning. *Artificial Intelligence* 95(1): 187–201.

Dixon, S.; and Wobcke, W. 1993. The Implementation of a First-Order Logic AGM Belief Revision System. In *ICTAI 1993*, 40–47. IEEE Computer Society.

Flouris, G. 2006. *On Belief Change and Ontology Evolution*. Ph.D. thesis, University of Crete.

Gabbay, D. 2003. *Many-dimensional modal logics : theory and applications*. Amsterdam Boston: Elsevier North Holland. ISBN 0444508260.

Guerra, P. T.; and Wassermann, R. 2019. Two AGM-style characterizations of model repair. *Ann. Math. Artif. Intell.* 87(3): 233–257.

Hansson, S. O. 1993. Changes of disjunctively closed bases. *Journal of Logic, Language and Information* 2(4): 255–284.

Hansson, S. O. 1994. Taking Belief Bases Seriously. In *Logic and Philosophy of Science in Uppsala: Papers from the 9th International Congress of Logic, Methodology and*

Philosophy of Science, 13–28. Dordrecht: Springer Netherlands. ISBN 978-94-015-8311-4.

Hansson, S. O. 1996. Knowledge-Level Analysis of Belief Base Operations. *Artificial Intelligence* 82(1-2): 215–235.

Hansson, S. O. 1999. *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Applied Logic Series. Kluwer Academic Publishers.

Kalyanpur, A. 2006. *Debugging and repair of OWL ontologies*. Ph.D. thesis, University of Maryland.

Katsuno, H.; and Mendelzon, A. O. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3): 263–294.

Makinson, D.; and Gärdenfors, P. 1989. Relations between the logic of theory change and nonmonotonic logic. In *The Logic of Theory Change*, volume 465, 185–205. Springer.

Matos, V. B.; Guimarães, R.; Santos, Y. D.; and Wassermann, R. 2019. Pseudo-contractions as Gentle Repairs. In *Lecture Notes in Computer Science*, 385–403. Springer International Publishing.

Nebel, B. 1991. Belief Revision and Default Reasoning: Syntax-Based Approaches. In *KR 1991*, 417–428. Morgan Kaufmann.

Ribeiro, J. S.; Nayak, A.; and Wassermann, R. 2018. Towards Belief Contraction without Compactness. In *KR 2018*, 287–296. AAAI Press.

Ribeiro, J. S.; Nayak, A.; and Wassermann, R. 2019a. Belief Change and Non-Monotonic Reasoning Sans Compactness. In *AAAI 2019*, 3019–3026. AAAI Press.

Ribeiro, J. S.; Nayak, A.; and Wassermann, R. 2019b. Belief Update without Compactness in Non-finitary Languages. In *IJCAI 2019*, 1858–1864. ijcai.org.

Ribeiro, M. M. 2013. *Belief Revision in Non-Classical Logics*. Springer London.

Ribeiro, M. M.; and Wassermann, R. 2014. Minimal Change in AGM Revision for Non-Classical Logics. In *KR 2014*. AAAI Press.

Santos, Y. D.; Matos, V. B.; Ribeiro, M. M.; and Wassermann, R. 2018. Partial meet pseudo-contractions. *International Journal of Approximate Reasoning* 103: 11–27.

Suntisrivaraporn, B. 2009. *Polynomial time reasoning support for design and maintenance of large-scale biomedical ontologies*. Ph.D. thesis, Dresden University of Technology, Germany.

Troquard, N.; Confalonieri, R.; Galliani, P.; Peñaloza, R.; Porello, D.; and Kutz, O. 2018. Repairing Ontologies via Axiom Weakening. In *AAAI 2018*, 1981–1988. AAAI Press.

A Proofs for Section 3

Lemma 18. Let φ, ϕ be $\mathcal{ALC}$-formulae. If $\varphi \in \mathsf{f}(\phi)$ then $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$.

Proof. The proof follows by induction in the structure of ϕ.
Base: ϕ is atomic. Then, by construction $\mathsf{f}(\phi) = \{\phi, \neg\phi\}$. Thus, if $\varphi \in \mathsf{f}(\phi)$ then $\varphi = \phi$ or $\varphi = \neg\phi$. In either case, $\mathsf{f}(\varphi) = \{\varphi, \neg\varphi\}$ which implies that $\mathsf{f}(\varphi) = \{\phi, \neg\phi\}$. Thus, $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$.

In the following, assume that ϕ is not atomic.
Induction Hypothesis: by construction ϕ is defined as the conjunction of two formulae ψ and ψ' or the negation of such conjunction, that is, $\phi = \psi \wedge \psi'$ or $\varphi = \neg(\psi \wedge \psi')$. Let us assume that for all $\beta \in \{\psi, \psi'\}$, if $\varphi \in \mathsf{f}(\beta)$ then $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\beta)$.
Induction step: consider the cases (i) $\phi = \psi \wedge \psi'$ and (ii) $\phi = \neg(\psi \wedge \psi')$.

(i) $\phi = \psi \wedge \psi'$. By construction

$$\mathsf{f}(\phi) = \mathsf{f}(\psi \wedge \psi') = \{\psi \wedge \psi', \neg(\psi \wedge \psi')\} \cup \mathsf{f}(\psi) \cup \mathsf{f}(\psi'). \tag{1}$$

Thus, (a) $\varphi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$ or (b) $\varphi \in \mathsf{f}(\psi)$ or (c) $\varphi \in \mathsf{f}(\psi')$.

(a) $\varphi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$. Thus, either $\varphi = \psi \wedge \psi'$ or $\varphi = \neg(\psi \wedge \psi')$. For $\varphi = \psi \wedge \psi'$, we get that $\mathsf{f}(\varphi) = \mathsf{f}(\psi \wedge \psi') = \mathsf{f}(\phi)$ which means that $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$. For $\varphi = \neg(\psi \wedge \psi')$, we get that $\mathsf{f}(\varphi) = \mathsf{f}(\neg(\psi \wedge \psi'))$. By construction, $\mathsf{f}(\neg(\psi \wedge \psi')) = \mathsf{f}(\psi \wedge \psi')$. Therefore, $\mathsf{f}(\varphi) = \mathsf{f}(\psi \wedge \psi') = \mathsf{f}(\phi)$ and so $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$.

(b) $\varphi \in \mathsf{f}(\psi)$. By the inductive hypothesis, $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\psi)$. From (1), $\mathsf{f}(\psi) \subseteq \mathsf{f}(\phi)$. So, $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$.

(c) $\varphi \in \mathsf{f}(\psi')$. Analogous to item (b).

(ii) $\phi = \neg(\psi \wedge \psi')$. By construction, $\mathsf{f}(\neg(\psi \wedge \psi')) = \mathsf{f}(\psi \wedge \psi')$. So, $\mathsf{f}(\phi) = \mathsf{f}(\psi \wedge \psi') = \{\psi \wedge \psi', \neg(\psi \wedge \psi')\} \cup \mathsf{f}(\psi) \cup \mathsf{f}(\psi')$. Proof proceeds as in item (i). □

Lemma 19. For every $\mathcal{ALC}$-formula ϕ and formula type $\mathbf{f}$ for ϕ, if $\phi, \varphi \in \mathbf{f}$ then $\mathbf{f} \cap \mathsf{f}(\varphi)$ is a formula type for φ.

Proof. Let $\mathbf{f}_\phi$ be a fixed but arbitrary formula type for ϕ with $\phi \in \mathbf{f}_\phi$. We will show that $\mathbf{f} := \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$ is a formula type for φ. Suppose for contradiction that $\mathbf{f}$ is not a formula type for φ. Thus, as $\mathbf{f} \subseteq \mathsf{f}(\varphi)$, either condition (1) or (2) of the formula type definition is violated:

1. There are formulae $\psi, \neg\psi \in \mathsf{f}(\varphi)$ such that either (a) $\psi \notin \mathbf{f}$ and $\neg\psi \notin \mathbf{f}$, or (b) $\psi, \neg\psi \in \mathbf{f}$.

   (a) $\psi \notin \mathbf{f}$ and $\neg\psi \notin \mathbf{f}$. By hypothesis, $\varphi \in \mathbf{f}_\phi$. Thus, as $\mathbf{f} = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$, and by construction $\varphi \in \mathsf{f}(\varphi)$, we get that $\varphi \in \mathbf{f}$. Since $\mathbf{f}_\phi$ is a formula type, we have that for all $\psi' \in \mathsf{f}(\phi)$, $\psi' \in \mathbf{f}_\phi$ iff $\neg\psi' \notin \mathbf{f}_\phi$. As $\varphi \in \mathbf{f}_\phi \subseteq \mathsf{f}(\phi)$, it follows from Lemma 18 that $\mathsf{f}(\varphi) \subseteq \mathsf{f}(\phi)$. Therefore, for all $\psi' \in \mathsf{f}(\varphi)$, $\psi' \in \mathbf{f}_\phi$ iff $\neg\psi' \notin \mathbf{f}_\phi$. By hypothesis, $\neg\psi, \psi \in \mathsf{f}(\varphi)$ which implies from above that either:

   $$\psi \in \mathbf{f}_\phi \text{ and } \neg\psi \notin \mathbf{f}_\phi, \text{ or } \psi \notin \mathbf{f}_\phi \text{ and } \neg\psi \in \mathbf{f}_\phi. \tag{2}$$

   By hypothesis, $\neg\psi, \psi \in \mathsf{f}(\varphi)$ but $\neg\psi, \psi \notin \mathbf{f}$. Thus, as $\mathbf{f} = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$, we get $\neg\psi, \psi \notin \mathbf{f}_\phi$, contradicting (2).

   (b) $\psi, \neg\psi \in \mathbf{f}$. By hypothesis, $\mathbf{f}_\phi$ is a formula type which implies that for all $\psi' \in \mathbf{f}_\phi$, $\psi', \neg\psi' \notin \mathbf{f}_\phi$. Therefore, as $\mathbf{f} \subseteq \mathbf{f}_\phi$, we get that $\psi, \neg\psi \notin \mathbf{f}$, a contradiction.

2. Let $\psi \wedge \psi' \in \mathsf{f}(\varphi)$. We will show that $\psi \wedge \psi' \in \mathbf{f}$ iff $\{\psi, \psi'\} \subseteq \mathbf{f}$ which contradicts the hypothesis that condition (2) from the formula type definition is violated. We

split the proof in two cases: either (a) $\psi \wedge \psi' \in \mathbf{f}$ or (b) $\psi \wedge \psi' \notin \mathbf{f}$. If $\psi \wedge \psi' \in \mathbf{f}$, as $\mathbf{f} = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$, we get that $\psi \wedge \psi' \in \mathbf{f}_\phi$. Since $\mathbf{f}_\phi$ is a formula type, we have that $\{\psi, \psi'\} \subseteq \mathbf{f}_\phi$. By definition of $\mathsf{f}(\varphi)$, if $\psi \wedge \psi' \in \mathsf{f}(\varphi)$ then $\{\psi, \psi'\} \subseteq \mathsf{f}(\varphi)$. Hence, $\{\psi, \psi'\} \in \mathbf{f} = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$.

Otherwise, $\psi \wedge \psi' \notin \mathbf{f}$. As $\mathbf{f} = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$ and $\psi \wedge \psi' \in \mathsf{f}(\varphi)$, we get that $\psi \wedge \psi' \notin \mathbf{f}_\phi$. Thus, as $\mathbf{f}_\phi$ is a formula type, we get that $\{\psi, \psi'\} \not\subseteq \mathbf{f}_\phi$. Therefore, as $\mathbf{f} \subseteq \mathbf{f}_\phi$, we get that $\{\psi, \psi'\} \not\subseteq \mathbf{f}$. From (a) and (b) we conclude that $\psi \wedge \psi' \in \mathbf{f}$ iff $\{\psi, \psi'\} \subseteq \mathbf{f}$. But this contradicts the hypothesis that condition (2) from the formula type definition is violated.

Therefore, we conclude that $\mathbf{f}$ is a formula type. □

Lemma 20. For every $\mathcal{ALC}$-formula φ, $\mathsf{f}(\varphi) = \mathsf{f}(\neg\varphi)$ [2].

Proof. By construction φ is a subformula of $\neg\varphi$. We can see that $\mathsf{f}(\neg\varphi) = \mathsf{f}(\varphi) \cup \{\neg\varphi\}$. Since $\mathsf{f}(\varphi)$ is closed under single negation and, by construction, $\varphi \in \mathsf{f}(\varphi)$, we have that $\neg\varphi \in \mathsf{f}(\varphi)$. Thus, $\mathsf{f}(\varphi) = \mathsf{f}(\neg\varphi)$. □

Definition 21. Let φ be an $\mathcal{ALC}$-formula. The set of of formula types for φ that has φ is given by the set

$$\tau(\varphi) = \{\mathbf{f} \subseteq \mathsf{f}(\varphi) \mid \mathbf{f} \text{ is a formula type for } \varphi \text{ and } \varphi \in \mathbf{f}\}.$$

Lemma 22. For every $\mathcal{ALC}$-formula ϕ and formula type $\mathbf{f}$ for ϕ, if $\phi \in \mathbf{f}$ and $\varphi \in \mathsf{f}(\phi)$ then $\mathbf{f} \cap \mathsf{f}(\varphi) \in \tau(\varphi) \cup \tau(\neg\varphi)$.

Proof. Let $\mathbf{f}_\phi$ be a fixed but arbitrary formula type for ϕ with $\phi \in \mathbf{f}_\phi$. As $\mathbf{f}_\phi$ is a formula type (for ϕ) and $\varphi \in \mathsf{f}(\phi)$, either (i) $\varphi \in \mathbf{f}_\phi$ or $\neg\varphi \in \mathbf{f}_\phi$:

(i) $\varphi \in \mathbf{f}_\phi$. Thus, by Lemma 19, we have that $\mathbf{f}_\phi \cap \mathsf{f}(\varphi)$ is a formula type of φ. Also, $\varphi \in \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$. Therefore, $\mathbf{f}_\phi \cap \mathsf{f}(\varphi) \in \tau(\varphi)$ which means that $\mathbf{f}_\phi \cap \mathsf{f}(\varphi) \in \tau(\varphi) \cup \tau(\neg\varphi)$.

(ii) $\neg\varphi \in \mathbf{f}_\phi$. Thus, by Lemma 19, we have that $\mathbf{f}_\phi \cap \mathsf{f}(\neg\varphi)$ is a formula type for $\neg\varphi$. Also, $\neg\varphi \in \mathbf{f}_\phi \cap \mathsf{f}(\neg\varphi)$. Therefore, $\mathbf{f}_\phi \cap \mathsf{f}(\neg\varphi) \in \tau(\neg\varphi)$ which means that $\mathbf{f}_\phi \cap \mathsf{f}(\neg\varphi) \in \tau(\varphi) \cup \tau(\neg\varphi)$. By Lemma 20, we have that $\mathsf{f}(\varphi) = \mathsf{f}(\neg\varphi)$ which implies that $\mathbf{f}_\phi \cap \mathsf{f}(\neg\varphi) = \mathbf{f}_\phi \cap \mathsf{f}(\varphi)$. Therefore, $\mathbf{f}_\phi \cap \mathsf{f}(\varphi) \in \tau(\varphi) \cup \tau(\neg\varphi)$. □

Lemma 23. For every $\mathcal{ALC}$-formula φ, $\mathbf{f} \in \tau(\varphi)$ iff $\mathbf{f}$ is a formula type for φ and

1. if φ is atomic then $\mathbf{f} = \{\varphi\}$;
2. if $\varphi = \psi \wedge \psi'$ then $\mathbf{f} = \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$, for some $\mathbf{f}_\psi \in \tau(\psi)$ and $\mathbf{f}_{\psi'} \in \tau(\psi')$;
3. if $\varphi = \neg(\psi \wedge \psi')$ then $\mathbf{f} = \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$, for some $\mathbf{f}_\psi \in \tau(\psi) \cup \tau(\neg\psi)$, $\mathbf{f}_{\psi'} \in \tau(\psi') \cup \tau(\neg\psi')$ such that either $\mathbf{f}_\psi \in \tau(\neg\psi)$ or $\mathbf{f}_{\psi'} \in \tau(\neg\psi')$.

Proof. The direction "$\Leftarrow$" is trivial, so we focus only on the "$\Rightarrow$" direction. Let $\mathbf{f} \in \tau(\varphi)$. Thus, $\varphi \in \mathbf{f}$ and $\mathbf{f}$ is a formula type for φ. By construction, (I) either φ is atomic or (II) $\varphi = \psi \wedge \psi'$ or (III) $\varphi = \neg(\psi \wedge \psi')$:

(I) φ is atomic. Thus, by construction $\mathbf{f} = \{\varphi\}$ or $\mathbf{f} = \{\neg\varphi\}$. Thus, as $\varphi \in \mathbf{f}$, we get $\mathbf{f} = \{\varphi\}$.

(II) $\varphi = \psi \wedge \psi'$. As $\varphi \in \mathbf{f}$, we get that $\psi \wedge \psi' \in \mathbf{f}$. Moreover, as $\mathbf{f}$ is a formula type for φ and $\psi \wedge \psi' \in \mathbf{f}$, it follows that $\psi, \psi' \in \mathbf{f}$.

Let $\mathbf{f}_\psi := \mathbf{f} \cap \mathsf{f}(\psi)$ and $\mathbf{f}_{\psi'} := \mathbf{f} \cap \mathsf{f}(\psi')$. As $\psi, \psi' \in \mathbf{f}$ and $\mathbf{f}$ is a formula type for $\varphi = \psi \wedge \psi'$, by Lemma 19, $\mathbf{f}_\psi = \mathbf{f} \cap \mathsf{f}(\psi)$ is a formula type for ψ and $\mathbf{f}_{\psi'} = \mathbf{f} \cap \mathsf{f}(\psi')$ is a formula type for ψ'. We have that $\psi \in \mathsf{f}(\psi)$ and $\psi' \in \mathsf{f}(\psi')$ which means that $\psi \in \mathbf{f}_\psi$ and $\psi' \in \mathbf{f}_{\psi'}$. Thus, $\mathbf{f}_\psi \in \tau(\psi)$ and $\mathbf{f}_{\psi'} \in \tau(\psi')$. We still need to show that $\mathbf{f} = \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$. For this, we will show that (i) $\mathbf{f} \subseteq \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$ and (ii) $\{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'} \subseteq \mathbf{f}$. The case (ii) is trivial, so we focus only on case (i). Let $\phi \in \mathbf{f}$. As $\mathbf{f}$ is a formula type for $\varphi = \psi \wedge \psi'$, we get that

$$\phi \in \mathbf{f} \subseteq \mathsf{f}(\psi \wedge \psi') = \{\psi \wedge \psi', \neg(\psi \wedge \psi')\} \cup \mathsf{f}(\psi) \cup \mathsf{f}(\psi').$$

Therefore, (a) $\phi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$ or (b) $\phi \in \mathsf{f}(\psi)$ or (c) $\phi \in \mathsf{f}(\psi')$.

(a) $\phi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$. As $\mathbf{f}$ is a formula type and $\varphi = \psi \wedge \psi' \in \mathbf{f}$, we get that $\neg(\psi \wedge \psi') \notin \mathbf{f}$. Thus, as $\phi \in \mathbf{f}$, we have that $\phi \neq \neg(\psi \wedge \psi')$. Hence, $\phi = \psi \wedge \psi'$, which implies that $\phi \in \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

(b) $\phi \in \mathsf{f}(\psi)$. Thus, as $\phi \in \mathbf{f}$, we get that $\phi \in \mathbf{f}_\psi = \mathbf{f} \cap \mathsf{f}(\psi)$ which implies that $\phi \in \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

(c) $\phi \in \mathsf{f}(\psi')$. Thus, as $\phi \in \mathbf{f}$, we get that $\phi \in \mathbf{f}_{\psi'} = \mathbf{f} \cap \mathsf{f}(\psi')$ which implies that $\phi \in \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

Thus, $\phi \in \{\psi \wedge \psi'\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

(III) $\varphi = \neg(\psi \wedge \psi')$. As $\varphi \in \mathbf{f}$, we get that $\neg(\psi \wedge \psi') \in \mathbf{f}$. Let $\mathbf{f}_\psi := \mathbf{f} \cap \mathsf{f}(\neg\psi)$ and $\mathbf{f}_{\psi'} := \mathbf{f} \cap \mathsf{f}(\neg\psi')$.

As $\neg\psi, \neg\psi' \in \mathsf{f}(\varphi = \neg(\psi \wedge \psi'))$, by Lemma 22, we have that $\mathbf{f}_\psi \in \tau(\psi) \cup \tau(\neg\psi)$ and $\mathbf{f}_{\psi'} \in \tau(\psi') \cup \tau(\neg\psi')$.

Moreover, as $\mathbf{f}$ is a formula type for φ and $\varphi = \neg(\psi \wedge \psi') \in \mathbf{f}$, it follows that $\psi \wedge \psi' \notin \mathbf{f}$. Therefore, $\{\psi, \psi'\} \not\subseteq \mathbf{f}$. Thus, either $\psi \notin \mathbf{f}$ or $\psi' \notin \mathbf{f}$. Thus, as $\mathbf{f}$ is a formula type, either (i) $\neg\psi \in \mathbf{f}$ or (ii) $\neg\psi' \in \mathbf{f}$.

(i) $\neg\psi \in \mathbf{f}$. Thus, as $\mathbf{f}$ is a formula type for $\varphi = \neg(\psi \wedge \psi')$, by Lemma 19, $\mathbf{f}_\psi = \mathbf{f} \cap \mathsf{f}(\neg\psi)$ is a formula type for $\neg\psi$. We have that $\neg\psi \in \mathsf{f}(\neg\psi)$. So $\neg\psi \in \mathbf{f}_\psi$. Thus, $\mathbf{f}_\psi \in \tau(\neg\psi)$.

(ii) $\neg\psi' \in \mathbf{f}$. Analogously to item (i): $\mathbf{f}_{\psi'} \in \tau(\neg\psi')$.

Thus, $\mathbf{f}_\psi \in \tau(\neg\psi)$ or $\mathbf{f}_{\psi'} \in \tau(\neg\psi')$.

We still need to show that $\mathbf{f} = \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$. For this we need to show that (i) $\mathbf{f} \subseteq \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$ and (ii) $\{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'} \subseteq \mathbf{f}$. The case (ii) is trivial. So we focus only on case (i).

Let $\phi \in \mathbf{f}$. As $\mathbf{f}$ is a formula type for $\varphi = \neg(\psi \wedge \psi')$, we get that $\phi \in \mathbf{f} \subseteq \mathsf{f}(\neg(\psi \wedge \psi')) = \mathsf{f}(\psi \wedge \psi') = \{\psi \wedge \psi', \neg(\psi \wedge \psi')\} \cup \mathsf{f}(\psi) \cup \mathsf{f}(\psi')$. Therefore, (a) $\phi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$ or (b) $\phi \in \mathsf{f}(\psi)$ or (c) $\phi \in \mathsf{f}(\psi')$.

(a) $\phi \in \{\psi \wedge \psi', \neg(\psi \wedge \psi')\}$. As $\mathbf{f}$ is a formula type and $\varphi = \neg(\psi \wedge \psi') \in \mathbf{f}$, we get that $(\psi \wedge \psi') \notin \mathbf{f}$. Since $\phi \in \mathbf{f}$, we have that $\phi \neq (\psi \wedge \psi')$. Therefore, $\phi = \neg(\psi \wedge \psi')$, which implies that $\phi \in \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

(b) $\phi \in \mathsf{f}(\psi)$. By Lemma 20, we get $\mathsf{f}(\psi) = \mathsf{f}(\neg\psi)$. Therefore, $\phi \in \mathsf{f}(\neg\psi)$. Thus, as $\phi \in \mathbf{f}$, we get that $\phi \in \mathbf{f}_\psi = \mathbf{f} \cap \mathsf{f}(\neg\psi)$ which implies that $\phi \in \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$

[2] We silently remove double negation and treat $\neg\neg\phi$ as equal to ϕ.

(c) $\phi \in \mathsf{f}(\psi')$. Analogously to item (b), we get $\phi \in \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$.

Thus, $\phi \in \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}$. □

Definition 24 (Formula degree)**.** The degree of an $\mathcal{ALC}$-formula ϕ, denoted $degree(\phi)$, is

- 1 if ϕ is an atomic $\mathcal{ALC}$-formula;
- $degree(\varphi) + 1$ if $\phi = \neg\varphi$; and
- $degree(\varphi) + degree(\psi)$ if $\phi = \varphi \wedge \psi$.

Lemma 25 ((Gabbay 2003))**.** If $\mathcal{I} \models \varphi$ then $\mathrm{qm}(\varphi, \mathcal{I})$ is a quasimodel for φ.

Lemma 26. Let φ be an $\mathcal{ALC}$-formula. If $\mathbf{f} \in \tau(\varphi)$ then

$$\left(\bigwedge lit(\mathbf{f})\right) \models \varphi.$$

Proof. The proof follows by induction in the degree of ϕ.

Base: $degree(\phi) = 1$. Then ϕ is atomic. This implies from Lemma 23 that $\mathbf{f} = \{\phi\}$. Thus, $\bigwedge lit(\mathbf{f}) = \phi$. For $\mathcal{ALC}$, this means that $\bigwedge lit(\mathbf{f}) \models \phi$.

Induction Hypothesis: For every formula φ, and formula type $\mathbf{f}_\varphi$ for φ, if $\varphi \in \mathbf{f}_\varphi$ and $degree(\varphi) < degree(\phi)$ then $\bigwedge lit(\mathbf{f}_\varphi) \models \varphi$.

Induction Step: If $degree(\phi) > 1$ then ϕ is of the form $\varphi \wedge \psi$ or $\neg\varphi$, for some $\mathcal{ALC}$-formulae φ and ψ:

1. $\phi = \varphi \wedge \psi$. Thus, from Lemma 23,

   $$\mathbf{f} = \{\varphi \wedge \psi\} \cup \mathbf{f}_\varphi \cup \mathbf{f}_\psi, \text{ such that } \mathbf{f}_\varphi \in \tau(\varphi), \mathbf{f}_\psi \in \tau(\psi).$$

   Note that $lit(\mathbf{f}) = lit(\mathbf{f}_\varphi) \cup lit(\mathbf{f}_\psi)$. Therefore,

   $$\bigwedge lit(\mathbf{f}) = \left(\bigwedge lit(\mathbf{f}_\varphi)\right) \wedge \left(\bigwedge lit(\mathbf{f}_\psi)\right)$$

   By the definition of degree, we get that $degree(\phi) = degree(\varphi \wedge \psi) = degree(\varphi) + degree(\psi)$ and $1 \leq degree(\varphi)$ and $1 \leq degree(\psi)$. Therefore, $degree(\varphi) < degree(\phi)$ and $degree(\psi) < degree(\phi)$. By the inductive hypothesis, $\bigwedge lit(\mathbf{f}_\varphi) \models \varphi$ and $\bigwedge lit(\mathbf{f}_\psi) \models \psi$. Therefore, $\bigwedge lit(\mathbf{f}) = \bigwedge lit(\mathbf{f}_\varphi) \wedge \bigwedge lit(\mathbf{f}_\psi) \models \varphi \wedge \psi$. Thus, as $\phi = \varphi \wedge \psi$, we get $\bigwedge lit(\mathbf{f}) \models \phi$.

2. $\phi = \neg\varphi$. By construction, either: (a) φ is atomic, or (b)$\varphi = \psi \wedge \psi'$.

   (a) φ is atomic. We get from Lemma 23 that $\mathbf{f} = \{\neg\varphi\}$, which implies that $lit(\mathbf{f}) = \{\neg\varphi\}$, and analogous to the base case, we get that $\bigwedge lit(\mathbf{f}) \models \neg\varphi$ that is, $\bigwedge lit(\mathbf{f}) \models \phi$.

   (b) $\varphi = \psi \wedge \psi'$. By Lemma 23, we get that

   $$\mathbf{f} = \{\neg(\psi \wedge \psi')\} \cup \mathbf{f}_\psi \cup \mathbf{f}_{\psi'}, \tag{3}$$

   where $\mathbf{f}_\psi \in \tau(\psi) \cup \tau(\neg\psi), \mathbf{f}_{\psi'} \in \tau(\psi') \cup \tau(\neg\psi')$ such that either (i) $\mathbf{f}\psi \in \tau(\neg\psi)$ or (ii) $\mathbf{f}_{\psi'} \in \tau(\neg\psi')$.

   i. $\mathbf{f}_\psi \in \tau(\neg\psi)$. By the definition of degree, we get that $degree(\phi) = degree(\neg(\psi \wedge \psi')) = degree(\psi) + degree(\psi') + 1$, and $degree(\psi) \geq 1$ and $degree(\psi') \geq 1$ and $degree(\neg\psi) = degree(\psi) + 1$. Thus, $degree(\phi) = degree(\neg(\psi \wedge \psi')) = degree(\neg\psi) + degree(\psi')$. Thus, as $degree(\psi') \geq 1$ we get $degree(\neg\psi) < degree(\phi)$. Thus, by the inductive hypothesis, $\bigwedge lit(\mathbf{f}_\psi) \models \neg\psi$. Note that for every formula β, $\neg\psi \models \neg(\psi \wedge \beta)$. Therefore, for $\beta = \psi'$: $\bigwedge lit(\mathbf{f}_\psi) \models \neg(\psi \wedge \psi')$.
   From (3), we get that

   $$\bigwedge lit(\mathbf{f}) = \bigwedge lit(\mathbf{f}_\psi) \wedge \bigwedge lit(\mathbf{f}_{\psi'}).$$

   Thus, as $\bigwedge lit(\mathbf{f}_\psi) \models \neg(\psi \wedge \psi')$, we get that $\bigwedge lit(\mathbf{f}_\psi) \wedge \bigwedge lit(\mathbf{f}_{\psi'}) \models \neg(\psi \wedge \psi')$ which implies from above that $\bigwedge lit(\mathbf{f}) \models \neg(\psi \wedge \psi')$ that is,

   $$\bigwedge lit(\mathbf{f}) \models \phi.$$

   ii. $\mathbf{f}_\psi \in \tau(\neg\psi')$. Analogous to item (i). □

Theorem 10. For every $\mathcal{ALC}$-formula φ: $\varphi \equiv \varphi^\dagger$.

Proof. Let φ be an $\mathcal{ALC}$-formula and $\mathcal{I}$ an interpretation. First, suppose that $\mathcal{I} \models \varphi$. From Lemma 25 we know that $\mathrm{qm}(\varphi, \mathcal{I}) = (T, o, \mathbf{f})$ is a quasimodel of φ. Therefore, there is a disjunct ψ of $\varphi^\dagger$ which is the conjunction of all atomic formulae in $\mathbf{f}$. By Definition 9 $\mathcal{I} \models \mathbf{f}$, thus we can conclude that $\mathcal{I} \models \varphi^\dagger$. Now, assume that $\mathcal{I} \models \varphi^\dagger$. This means that there is one disjunct ψ of $\varphi^\dagger$ such that $\mathcal{I} \models \psi$. By construction, this disjunct is a conjunction of atomic formulae in the formula type of a quasimodel $(T, o, \mathbf{f})$ for φ. Using Lemma 26 we can conclude that $\mathcal{I} \models \mathbf{f}$. As $\varphi \in \mathbf{f}$ we get that $\mathcal{I} \models \varphi$. Hence, $\mathcal{I} \models \varphi$ iff $\mathcal{I} \models \varphi^\dagger$, i.e., $\varphi \equiv \varphi^\dagger$. □

Corollary 27 is a direct consequence of the definition of a formula type.

Corollary 27. Let $(T, o, \mathbf{f})$ and $(T', o', \mathbf{f}')$ be quasimodels for an $\mathcal{ALC}$-formula φ. Then, $lit(\mathbf{f}) = lit(\mathbf{f}')$ iff $\mathbf{f} = \mathbf{f}'$.

Given $\mathcal{ALC}$-formulae φ, ψ, we say that ψ is in the language of the literals of φ, written $\psi \in \mathcal{L}_{lit}(\varphi)$, if ψ is a boolean combination of the atoms in φ.

Lemma 28. Let M, M' be models and φ an $\mathcal{ALC}$-formula. Also let $(T, o, \mathbf{f}) := \mathrm{qm}(\varphi, M)$ and $(T', o', \mathbf{f}') := \mathrm{qm}(\varphi, M')$. Then, $[M]^\varphi = [M']^\varphi$ iff $\mathbf{f} = \mathbf{f}'$.

Proof. First, assume that $[M]^\varphi = [M']^\varphi$. Then we know that for every $\alpha \in \mathcal{L}_{lit}(\varphi)$, $M \models \alpha$ iff $M' \models \alpha$. With Corollary 27 we can conclude that $\mathbf{f} = \mathbf{f}'$. Now, assume that $\mathbf{f} = \mathbf{f}'$. Corollary 27 implies that $lit(\mathbf{f}) = lit(\mathbf{f}')$. That is, for every atomic subformula $\alpha \in \mathcal{L}_{lit}(\varphi)$ we have that $M \models \alpha$ iff $M' \models \alpha$, i.e., $[M]^\varphi = [M']^\varphi$. □

Lemma 29. Let M be a model and φ an $\mathcal{ALC}$-formula. Then, the following holds: $\mathrm{Mod}(\varphi) \setminus [M]^\varphi = \mathrm{Mod}(\mathrm{con}(\varphi, M))$.

Proof. Let $(T, o, \mathbf{f}) := \mathrm{qm}(\varphi, M)$ and $(T', o', \mathbf{f}') := \mathrm{qm}(\varphi, M')$. First, suppose that $M' \in \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$. We know that $M' \models \varphi$ and by Lemma 25 we get that $\mathrm{qm}(\varphi, M')$ is a quasimodel for φ. We also know that $M' \notin [M]^{\varphi}$. Thus, from Lemma 28, we obtain $\mathbf{f} \neq \mathbf{f}'$. Therefore, $\mathbf{f}' \in \mu(\varphi, M)$. Hence, $M' \in \mathrm{Mod}(\mathrm{con}(\varphi, M))$ and so $\mathrm{Mod}(\varphi) \setminus [M]^{\varphi} \subseteq \mathrm{Mod}(\mathrm{con}(\varphi, M))$.

Now, let $M' \in \mathrm{Mod}(\mathrm{con}(\varphi, M))$. This means that there is at least one $\mathbf{f}'' \in \mu(\varphi, M)$ such that $M' \models \bigwedge lit(\mathbf{f}'')$. But as consequence of the definition of formula type, this implies that $M' \in \mathrm{Mod}(\varphi)$ and thus $(T', o', \mathbf{f}') \in \mathsf{S}(\varphi)$. We also know that $M \notin [M]^{\varphi}$, otherwise $\mathbf{f}' = \mathbf{f}$ due to Lemma 28. Therefore, $M' \in \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and we can conclude that $\mathrm{Mod}(\mathrm{con}(\varphi, M)) \subseteq \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$.

Therefore, $\mathrm{Mod}(\mathrm{con}(\varphi, M)) = \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$. □

Proof. Assume $M \in \mathrm{Mod}(\varphi)$ and $M' \in [M]^{\varphi}$. Also let $(T, o, \mathbf{f}) := \mathrm{qm}(\varphi, M)$ and $(T', o', \mathbf{f}') := \mathrm{qm}(\varphi, M')$. As $M' \in [M]^{\varphi}$, it follows that $[M]^{\varphi} = [M']^{\varphi}$. Due to Lemma 28 we get that $\mathbf{f} = \mathbf{f}'$. Since $\mathbf{f} = \mathbf{f}'$, M and M' are indistinguishable for the $\mathcal{ALC}$-formula φ and $M' \in \mathrm{Mod}(\varphi)$ (see Definition 9). Since M' was an arbitrary model in $\mathrm{Mod}(\varphi)$ it follows that $[M]^{\varphi} \subseteq \mathrm{Mod}(\varphi)$. The other direction is straightforward. □

Theorem 12. Let M be a model and φ an $\mathcal{ALC}$-formula. A finite base model function $\mathrm{con}^*(\varphi, M)$ is equivalent to $\mathrm{con}(\varphi, M)$ iff con^* satisfies:

(success) $M \not\models \mathrm{con}^*(\varphi, M)$,

(inclusion) $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) \subseteq \mathrm{Mod}(\varphi)$,

(atomic retainment): For all $\mathbb{M}' \subseteq \mathfrak{M}$, if $\mathrm{Mod}(\mathrm{con}^*(\mathcal{B}, M)) \subset \mathbb{M}' \subseteq \mathrm{Mod}(\mathcal{B}) \setminus [M]^{\varphi}$ then $\mathbb{M}'$ is not finitely representable in $\mathcal{ALC}$-formula.

(atomic extensionality) if $M' \equiv^{\varphi} M$ then

$$\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\mathrm{con}^*(\varphi, M')).$$

Proof. Assume that $\mathrm{con}^*(\varphi, M) \equiv \mathrm{con}(\varphi, M)$. From Lemma 29 we have that $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$, hence success and inclusion are immediately satisfied. To prove atomic retainment, assume that $M' \notin \mathrm{Mod}(\mathrm{con}^*(\varphi, M))$ and that there is a set of models $\mathbb{M}'$ with $M' \in \mathbb{M}'$, $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) \subset \mathbb{M}' \subseteq \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and that is finitely representable in $\mathcal{ALC}$-formula. Lemma 29 implies that $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$. Hence, $M' \in [M]^{\varphi}$, a contradiction as we assumed that $\mathbb{M}' \subseteq \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$. Therefore, no such $\mathbb{M}'$ could exist, and thus, con^* satisfies atomic retainment.

Let $M' \equiv^{\varphi} M$. Since $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and $[M']^{\varphi} = [M]^{\varphi}$, we have that: $\mathrm{Mod}(\varphi) \setminus [M]^{\varphi} = \mathrm{Mod}(\varphi) \setminus [M']^{\varphi} = \mathrm{Mod}(\mathrm{con}^*(\varphi, M'))$. Hence, atomic extensionality is also satisfied.

On the other hand, suppose that $\mathrm{con}^*(\varphi, M)$ satisfies the postulates stated. Let $M' \in \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and assume that $M' \notin \mathrm{Mod}(\mathrm{con}^*(\varphi, M))$. Due to atomic retainment, this means that there is no set $\mathbb{M}'$ finitely representable in $\mathcal{ALC}$-formula such that $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) \subset \mathbb{M}' \subseteq \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and $M' \in \mathbb{M}'$. But we know from Lemma 29 that $\mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ is finitely representable in $\mathcal{ALC}$-formula and includes M' by assumption, a contradiction. Thus, no such M' could exist and $\mathrm{Mod}(\varphi) \setminus [M]^{\varphi} \subseteq \mathrm{Mod}(\mathrm{con}^*(\varphi, M))$.

Now, let $M' \in \mathrm{Mod}(\mathrm{con}^*(\varphi, M))$. By inclusion $M' \in \mathrm{Mod}(\varphi)$ and by success $M' \neq M$. We will show that $M' \notin [M]^{\varphi}$. By contradiction, suppose that $M' \in [M]^{\varphi}$. Due to atomic extensionality $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) = \mathrm{Mod}(\mathrm{con}^*(\varphi, M'))$, but success implies that $M' \notin \mathrm{Mod}(\mathrm{con}^*(\varphi, M'))$. This contradicts our initial assumption that $M' \in \mathrm{Mod}(\mathrm{con}^*(\varphi, M))$. Therefore $M' \in \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$ and we can conclude that $\mathrm{Mod}(\mathrm{con}^*(\varphi, M)) \subseteq \mathrm{Mod}(\varphi) \setminus [M]^{\varphi}$.

Hence, Lemma 29 yields $\mathrm{con}(\varphi, M) \equiv \mathrm{con}^*(\varphi, M)$. □

Proposition 30. Let $\mathbf{f}$ be a formula type. If $M \models \bigwedge lit(\mathbf{f})$, $\psi \in \mathcal{L}_{lit}(\mathbf{f})$ and $M \models \psi$ then $\bigwedge lit(\mathbf{f}) \models \psi$.

Proof. Let $\mathbf{f}$ be a formula type, M a model such that $M \models \bigwedge lit(\mathbf{f})$, and ψ and $\mathcal{ALC}$-formula such that $M \models \psi$. The proof is by induction on the degree of ψ.
Base: $degree(\psi) = 1$. Thus, from its definition, ψ has to be an atomic formula. As $\mathbf{f}$ is a formula type, we have that $\varphi \in \mathbf{f}$ iff $\neg\varphi \notin \mathbf{f}$. Let us suppose for contradiction that $\psi \notin \mathbf{f}$. Thus, $\neg\psi \in \mathbf{f}$. This implies that $\bigwedge \mathbf{f} \models \neg\psi$. Thus, as $M \models \bigwedge lit(\mathbf{f})$, we have that $M \models \neg\psi$. This contradicts the hypothesis that $M \models \psi$. Thus, we conclude that $\psi \in \mathbf{f}$. Therefore, $\bigwedge lit(\mathbf{f}) \models \psi$.
Induction Hypothesis: For every formula φ, if $degree(\varphi) < degree(\psi)$ and $M \models \varphi$ then $\bigwedge lif(\mathbf{f}) \models \varphi$.
Induction Step: Let $degree(\psi) > 1$. By construction, ψ is of the form (1) $\varphi \wedge \varphi'$ or (2) $\neg\varphi$, for $\mathcal{ALC}$-formulae φ, φ':

(1) $\psi = \varphi \wedge \varphi'$. By definition, $degree(\varphi \wedge \varphi') = degree(\varphi) + degree(\varphi')$. Recall from the definition of $degree$ that $degree(\beta) > 1$, for every formula β. Therefore,
$degree(\varphi) < degree(\varphi \wedge \psi')$ and $degree(\varphi') < degree(\varphi \wedge \varphi')$. This means that $degree(\varphi) < degree(\psi)$ and $degree(\varphi') < degree(\psi)$. From hypothesis, $M \models \psi = \varphi \wedge \varphi'$. Thus, $M \models \varphi$ and $M \models \psi$. This implies from IH that $\bigwedge lit(\mathbf{f}) \models \varphi$ and $\bigwedge lit(\mathbf{f}) \models \varphi'$ Therefore, $\bigwedge lit(\mathbf{f}) \models \varphi \wedge \varphi' = \psi$.
(2) $\psi = \neg\varphi$. We have two cases, either (i) φ is an atomic formula or (ii) $\varphi = (\beta \wedge \beta')$. For the first case, analogous to the base case, we get that $\bigwedge lit(\mathbf{f}) \models \psi$. So we focus only on the second case. From the definition of $degree$, we get that $degree(\neg\beta) < degree(\psi)$ and $degree(\neg\beta') < degree(\psi)$. As $M \models \psi = \neg(\beta \wedge \beta')$, we get that either (a) $M \models \neg\beta$ or (b) $M \models \neg\beta'$.

(a) $M \models \neg\beta$. From above, $degree(\neg\beta) < degree(\psi)$. Thus, from IH, we get that $\bigwedge lit(\mathbf{f}) \models \neg\beta$. Thus, $\bigwedge lit(\mathbf{f}) \models \neg(\beta \wedge \beta') = \psi$.

(b): $M \models \neg\beta'$. Analogous to case (a). □

Lemma 31. If $M \models \varphi$, $\mathbf{f} \in \mathrm{ftypes}(\varphi)$ and $M \models \bigwedge lit(\mathbf{f})$ then $\mathrm{Mod}(\bigwedge lit(\mathbf{f})) = [M]^{\varphi}$.

Proof. We prove that $M' \in \mathrm{Mod}(\bigwedge lit(\mathbf{f}))$ iff $M' \in [M]^{\varphi}$.

"⇒". $M' \in \mathrm{Mod}(\bigwedge lit(\mathbf{f}))$. To show that $M' \in [M]^{\varphi}$, it suffices to show that $M' \equiv^{\varphi} M$. Let $\psi \in \mathcal{L}_{lit}(\varphi)$, we need to show that $M \models \psi$ iff $M \models \psi$.

(a) "⇒" $M \models \psi$. From Proposition 30, we have that $\bigwedge lit(\mathbf{f}) \models \psi$. This jointly with $M' \models \bigwedge lit(\mathbf{f})$ implies that $M' \models \psi$.

(b) "⇐" $M' \models \psi$. Analogous to item (a).

"⇐" $M' \in [M]^{\varphi}$. Note that $\bigwedge lit(\mathbf{f}) \in \mathcal{L}_{lit}(\varphi)$. Thus as $M' \equiv^{\varphi} M$, and $M \models \bigwedge lit(\mathbf{f})$, we get that $M' \models \bigwedge lit(\mathbf{f})$. □

Lemma 16. For every $\mathcal{ALC}$-formula φ and model M:

$$\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}.$$

Proof. We have two cases: either (i) $M \models \varphi$ or (ii)$M \not\models \varphi$. (i) $M \models \varphi$. Then, by the definition of ex, we have that $\mathrm{ex}(\varphi, M) = \varphi$ which implies that $\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi)$. As $M \models \varphi$, we get that $[M]^{\varphi} \subseteq \mathrm{Mod}(\varphi)$. Therefore, $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} = \mathrm{Mod}(\varphi)$. This implies that

$$\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}.$$

(ii)$M \not\models \varphi$. Thus, by the definition of ex, we get that

$$\mathrm{ex}(\varphi, M) = \varphi \vee \bigwedge lit(\mathbf{f}), \text{ where } qm(\neg\varphi, M) = (T, o, \mathbf{f}).$$

This implies that $\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi \vee \bigwedge lit(\mathbf{f}))$. Note that $\mathrm{Mod}(\varphi \vee \bigwedge lit(\mathbf{f})) = \mathrm{Mod}(\varphi) \cup \mathrm{Mod}(\bigwedge lit(\mathbf{f}))$.

As $qm(\neg\varphi, M) = (T, o, \mathbf{f})$, it follows from the definition of qm that $\mathbf{f} \in \mathrm{ftypes}(\neg\varphi)$ and $M \models \bigwedge lit(\mathbf{f})$. In summary, $M \models \neg\varphi$, $\mathbf{f} \in \mathrm{ftypes}(\neg\varphi)$ and $M \models \bigwedge lit(\mathbf{f})$. Thus, from Lemma 31, we have that $\mathrm{Mod}(\bigwedge lit(\mathbf{f})) = [M]^{\varphi}$. Therefore, $\mathrm{Mod}(\mathrm{ex}(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$. □

Theorem 17. Let M be a model and φ an $\mathcal{ALC}$-formula. A finite base model function $\mathrm{ex}^*(\varphi, M)$ is equivalent to $\mathrm{ex}(\varphi, M)$ iff ex^* satisfies:

(success) $M \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$.

(persistence): $\mathrm{Mod}(\varphi) \subseteq \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$.

(atomic temperance): For all $\mathbb{M}' \subseteq \mathfrak{M}$, if $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} \subseteq \mathbb{M}' \subset \mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) \cup \{M\}$ then $\mathbb{M}'$ is not finitely representable in $\mathcal{ALC}$-formula.

(atomic extensionality) if $M' \equiv^{\varphi} M$ then

$$\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) = \mathrm{Mod}(\mathrm{ex}^*(\varphi, M')).$$

Proof. First, assume that $\mathrm{ex}^*(\varphi, M) \equiv \mathrm{ex}(\varphi, M)$. From Lemma 16 we have that $\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$, hence success and persistence are immediately satisfied. To prove atomic temperance, assume that $M' \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$ and that there is a set of models $\mathbb{M}'$ with M' that is finitely representable in $\mathcal{ALC}$-formula and such that $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} \subseteq \mathbb{M}' \subset \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$. Lemma 16 implies that $\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$. Hence, $M' \notin [M]^{\varphi}$, a contradiction as we assumed that $\mathbb{M}' \supseteq \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$. Therefore, no such $\mathbb{M}'$ could exist, and thus, ex^* satisfies atomic temperance.

Let $M' \equiv^{\varphi} M$. Since $\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) = \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$ and $[M']^{\varphi} = [M]^{\varphi}$, we have that: $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} = \mathrm{Mod}(\varphi) \cup [M']^{\varphi} = \mathrm{Mod}(\mathrm{ex}^*(\varphi, M'))$. Hence, atomic extensionality is also satisfied.

On the other hand, suppose that $\mathrm{ex}^*(\varphi, M)$ satisfies the postulates stated. Let $M' \in \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$. If $M' \in \mathrm{Mod}(\varphi)$ then success ensures that $M' \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$. Otherwise, we have $M' \equiv^{\varphi} M$, and as consequence of success and atomic extensionality we also obtain $M' \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$. Therefore, $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} \subseteq \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$.

Now, let $M' \in \mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$ and assume that $M' \notin \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$. Success, persistence and atomic extensionality imply that $\mathrm{Mod}(\mathrm{ex}^*(\varphi, M))$. Atomic temperance states that there is no set of models $\mathbb{M}'$ that is finitely representable in $\mathcal{ALC}$-formula with $\mathrm{Mod}(\varphi) \cup [M]^{\varphi} \subseteq \mathbb{M}' \subset \mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) \cup \{M\}$. But we know from Lemma 16 that $\mathrm{Mod}(\varphi) \cup [M]^{\varphi}$ is finitely representable in $\mathcal{ALC}$-formula and does not include M' by assumption, a contradiction. Thus, no such M' could exist and $\mathrm{Mod}(\mathrm{ex}^*(\varphi, M)) \subseteq \mathrm{Mod}(\varphi) \cup [M]^{\varphi}$.

Hence, Lemma 16 yields $\mathrm{ex}^*(\varphi, M) \equiv \mathrm{ex}(\varphi, M)$. □

Belief Base Contraction by Cutting Connections

Matthew James Lynn , **James P. Delgrande**
Simon Fraser University, Canada
mlynn@cs.sfu.ca, jim@cs.sfu.ca

Abstract

This paper presents a methodology for constructing belief base contraction operators which preserve the syntactic structure of the initial belief base. We believe that preserving the structure of the initial belief base is an important requirement for practical belief base contraction. In our approach, individual occurrences of propositional variables are differentiated by the introduction of tags. Using the characterisation of unsatisfiability provided by the connection method, we may identify which specific variable occurrences result in a belief being entailed, and thus apply selective substitutions for these occurrences in order to block that entailment by effectively *cutting connections*. The resulting belief base has a structure almost identical to that of the initial belief base. We demonstrate that these contraction operators satisfy a number of desirable properties. Next, we present an algorithm for path-contraction with complexity **DP**. Finally, we introduce the notion of path-entailment to capture precisely what is preserved after a contraction, and show that the class of *regular path-contraction operators* satisfy an analogue of Parikh's postulate.

1 Introduction

Belief contraction is a form of belief change which occurs whenever an agent realises that it holds a belief which is no longer justified, and subsequently must modify its existing beliefs to ensure that this specified belief is no longer entailed by those beliefs it decides to retain. The challenge is to preserve as many of its existing beliefs as possible. This process is formalised as a *belief contraction operator* $-$ which takes a belief state κ alongside an existing belief ϕ and produces a contracted belief state $\kappa - \phi$.

Our contention is that belief contraction operators, and belief change functions more generally, should satisfy a *principle of structural preservation* analogous to the principle of categorical matching, which requires that the structure of the contracted beliefs should resemble the structure of the initial beliefs to the greatest extent possible. This is in conflict with purely semantic approaches to belief change, such as the Katsuno–Mendelzon approach, which require syntax-independence. This is also in conflict with approaches such as prime implicate based belief revision (Bienvenu, Herzig, and Qi 2008), and with approaches relying on disjunctive normal forms (Hunter and Agapeyev 2019) which generally involve an exponential cost as knowledge bases generally have the form of a large conjunction of small beliefs. We believe that pursuing the principle of structural preservation will help close the gap between practical belief representation and the representations convenient for naive implementations of belief change functions.

In this paper, we introduce the class of *path-contraction operators* which satisfy the principle of structural preservation for formulae in negation normal form. As conversion to negation normal form largely preserves the structure of a formula, we consider this a reasonable restriction. These operators work by tagging every occurrence of a propositional variable within the existing belief base with a unique tag, and then applying the connection method (Bibel 1981) to determine which particular occurrences contribute to the unwanted belief being entailed. Using this information, a process of selective substitution of $\top$ or $\bot$ for these particular occurrences, which we call *attenuation*, is employed to produce the resulting contracted belief base. The nature of this construction means that these path-contraction operators preserve the initial structure to a great extent.

This can be understood from a tableaux perspective: in order to compute $\kappa - \phi$ we proceed as follows. Assuming $\kappa \vdash \phi$ it follows that $\kappa \wedge \neg\phi$ is unsatisfiable, and therefore we may construct a closed fully expanded tableaux for $\kappa \wedge \neg\phi$. At this point, we select one or more branches, and selectively remove literals appearing along these branches which originate in κ until at least one branch is open. This results in a formula κ' obtained from κ by our process of *attenuation*, with the property that $\kappa \vdash \kappa'$ and $\kappa' \nvdash \phi$. Then, define $\kappa - \phi$ as κ'.

In Section 2 we present background material on propositional matrices, the connection method, existing approaches to belief base contraction, and on the distinction between explicit an implicit beliefs. Section 3 introduces the technique of attenuation, the notion of a cutting, and uses these to define the class of path-contraction operators which are shown to satisfy a handful of desirable properties. Section 4 presents a concrete algorithm for performing path-contraction alongside a complexity analysis. In Section 5 we introduce the notion of path-entailment and path-independence, which allow for characterising the preservation properties of path-contraction operators, and show an analogue of Parikh's Postulate to be satisfied by all *regular* path-contraction operators. We next compare this to existing literature in Section 6, and finally offer a summary of our contributions in Section 7.

2 Background Material

2.1 Propositional Logic

Let $V = \{p, q, r, \dots\}$ be a finite set of propositional variables. The corresponding propositional language L is constructed from V by applying the propositional connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. We use $\phi, \psi, \kappa, \dots$ to range over propositional formulae in L. We write $V(\phi)$ to denote the set of propositional variables occurring within ϕ.

Propositional formulae of the form $\neg p$ or p are called *literals*. When every negation occurring in ϕ is the negation of a variable we say that ϕ is in *negation normal form*. When ϕ is a disjunction of conjunctions of literals we say it is in *disjunctive normal form*, and when ϕ is a conjunction of disjunctions of literals we say it is in *conjunctive normal form*.

Functions $\nu, \mu : V \rightarrow \{T, F\}$ are referred to as *truth-value assignments* or just as *assignments*. Given a propositional formula ϕ we write $[\phi]$ for the set of assignments satisfying ϕ, with $\phi \vdash \psi$ indicating that $[\phi] \subseteq [\psi]$, and $\phi \equiv \psi$ indicating that $[\phi] = [\psi]$. In the case $[\phi] \neq \varnothing$ we say that ϕ is **satisfiable**, which is denoted by writing $\vdash \phi$. Similarly, we write $\phi \nvdash \psi$ if it is not the case that $\phi \vdash \psi$, and $\nvdash \phi$ if it is not the case that $\vdash \phi$.

2.2 Belief Base Contraction Operators

Belief contraction operators were formalised by Alchourron, Gärdenfors, and Makinson (1985) as binary functions $-$ which map a belief state κ alongside a belief to contract ϕ into a new contracted belief state $\kappa - \phi$ such that $\kappa - \phi \nvdash \phi$. Working with a finite vocabulary, both the belief state κ and the belief ϕ may be represented as propositional formulae along the lines of (Katsuno and Mendelzon 1991). Among the many postulates discussed in the aforementioned, there is an assumption that whenever $\kappa_1 \equiv \kappa_2$ then $\kappa_1 - \phi \equiv \kappa_2 - \phi$ meaning that belief contraction is meant to be syntax-independent. Rejecting this assumption leads to the subject of *belief base contraction*.

There are a number of approaches to belief base contraction in literature, where, usually, belief bases are represented as arbitrary sets of formulae. For instance, using selection functions to combine *remainders* of κ by ϕ which are maximal sets of beliefs from κ which do not entail ϕ (Hansson 1991), or by using incision functions to combine *kernels* of κ for ϕ which are minimal sets of beliefs from κ which do entail ϕ (Hansson 1994). Additional approaches are summarised nicely in (Peppas 2008).

In (Hansson 1999) a number of different properties are proposed that a belief base contraction operator may be required to satisfy. For our purposes, where we consider belief bases as comprising a single formula, we work with the following subset of those postulates discussed in (Caridroit, Konieczny, and Marquis 2017).

Definition 2.1. *A binary function $- : L \times L \rightarrow L$ is a **belief base contraction operator** iff it satisfies the following postulates:*

C1. *If $\nvdash \phi$ then $\kappa - \phi \nvdash \phi$.*

C2. *If $\vdash \phi$ then $\kappa - \phi = \kappa$.*

C3. $\kappa \vdash \kappa - \phi$.

C4. *If $\kappa \nvdash \phi$ then $\kappa - \phi = \kappa$.*

Postulate (C1) states that whenever ϕ is not a tautology, then $\kappa - \phi$ must not entail ϕ. Postulate (C2) states that whenever ϕ is a tautology, then $\kappa - \phi$ should not change anything as there is nothing which can be done to stop the entailment of ϕ anyways. Postulate (C3) states that $\kappa - \phi$ must be a consequence of κ, so that the process of contraction cannot result in new beliefs being adopted. Finally, postulate (C4) states that whenever ϕ is not a consequence of κ then contracting κ by ϕ should result in nothing being changed. We regard these postulates as serving to demarcate the broadest class of functions worth considering as belief base contraction operators, as the postulates capture very little of the requirement of minimal change.

In addition to these, we consider *Parikh's relevance postulate* (Parikh 1999) which further captures the requirement of minimal change by requiring that when some beliefs κ are being revised by a new belief ϕ then those beliefs in κ irrelevant to ϕ should remain unchanged. For example, beliefs about automobiles are irrelevant to those about birds. Thus, when contracting our beliefs about bird and automobiles so as to no longer entail that birds can fly, say in order to accommodate penguins, there should be no reason for any of our beliefs about automobiles to be altered in this process.

Irrelevance is formalised by considering decompositions $\kappa \equiv \kappa_1 \wedge \kappa_2$ with $V(\kappa_1) \cap V(\kappa_2) = \varnothing$ called syntax-splittings and considering κ_1 as irrelevant when revising κ by any ϕ with $V(\kappa_1) \cap V(\phi) = \varnothing$. For our purposes of studying belief base contraction, rather than belief revision, we use the following formulation of this:

Definition 2.2. *A belief base contraction operator $-$ satisfies **Parikh's relevance postulate** if and only if for any formulae κ_1, κ_2, and ϕ such that $V(\kappa_1) \cap (V(\kappa_2) \cup V(\phi)) = \varnothing$ then it follows that*

P. $(\kappa_1 \wedge \kappa_2) - \phi \equiv \kappa_1 \wedge (\kappa_2 - \phi)$.

2.3 Propositional Matrices

Our approach to belief contraction relies on the selective substitution of $\top$ or $\bot$ for propositional variables appearing within the belief base κ. In order to facilitate this, we attach distinct *tags* to each separate occurrence of a propositional variable in κ. Our examples use positive integers for tags, but the choice is arbitrary. We refer to propositional formulae in negation normal form which have been annotated with tags as *propositional matrices*, and use the variables $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{K}, \dots$ to range over them.

Definition 2.3. *A **matrix** is an expression constructed via the following rules:*

1. *The symbol $\top$ is a matrix.*
2. *If p is a variable in V and i is a tag then p^i and $\neg p^i$ are matrices.*
3. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices with no tags in common, then $(\mathcal{A} \wedge \mathcal{B})$ is a matrix.*

4. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices with no tags in common, then $(\mathcal{A} \vee \mathcal{B})$ is a matrix.*
5. *Nothing else is a matrix.*

We write M for the set of matrices.

It is worth noting that the use of the term “matrix” for a formula in negation normal form within the connection method literature is motivated by a graphical notation wherein disjunctions are represented by vertical juxtaposition, and conjunctions are represented by horizontal juxtaposition, or vice versa depending on the author. To illustrate, one might write

$$p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4 = \begin{bmatrix} p^1 & \begin{bmatrix} \neg p^2 \\ q^3 \end{bmatrix} & \neg q^4 \end{bmatrix}.$$

Our introduction of tags into the definition of a matrix amounts to a slight simplification of the approach in (Kreitz and Otten 1999; Otten 2011) which instead associates every subformula with a position label of its own.

Although matrices are required to be in negation normal form, we write $\neg\mathcal{A}$ to refer to the matrix obtained by temporarily treating $\mathcal{A}$ as a formula, and computing the negation normal form of $\neg\mathcal{A}$ by pushing negations down while retaining the tags. For example, $\neg(p^1 \vee q^2)$ refers to the matrix $\neg p^1 \wedge \neg q^2$

Definition 2.4. *We write $T(\mathcal{A})$ for the set of tags occurring in $\mathcal{A}$, and say that matrices $\mathcal{A}$ and $\mathcal{B}$ are **tag-disjoint** when $T(\mathcal{A}) \cap T(\mathcal{B}) = \varnothing$.*

Definition 2.5. *If p is a propositional variable then a matrix of the form p^i or $\neg p^i$ is called a **literal**. In the case the literal p^i or $\neg p^i$ appears in a matrix $\mathcal{A}$ we say that i **tags** the variable p.*

Definition 2.6. *The **detagging** of a matrix $\mathcal{A}$ is the propositional formula ϕ obtained by deleting the tags from $\mathcal{A}$, which we denote by $\epsilon(\mathcal{A})$.*

Example 2.1. *The matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4$ has detagging $p \wedge (\neg p \vee q) \wedge \neg q$.*

When working with matrices we say a truth-value assignment ν satisfies $\mathcal{A}$ when ν satisfies $\epsilon(\mathcal{A})$. We also say that $\mathcal{A}$ entails $\mathcal{B}$ and write $\mathcal{A} \vdash \mathcal{B}$ when $\epsilon(\mathcal{A})$ entails $\epsilon(\mathcal{B})$.

2.4 Connections in Propositional Matrices

Unsatisfiability of propositional matrices may be characterised in terms of *paths* and *connections*, where paths correspond roughly to the disjuncts of a disjunctive normal form of a formula, and connections correspond to pairs of complementary literals in those disjuncts. In the context of automated reasoning, this has become known as the *connection method* which originates with (Bibel 1981) and (Andrews 1976). Our presentation below is a variation on that of (Wallen 1987) and (Otten 2011).

Definition 2.7. *A **path** is a set $\mathfrak{p}$ of literal matrices such that each tag occurring in $\mathfrak{p}$ occurs exactly once. If there exists a variable p and tags i and j such that $\mathfrak{p}$ contains p^i and $\neg p^j$ then $\{p^i, \neg p^j\}$ is called a **connection** in $\mathfrak{p}$, and $\mathfrak{p}$ is said to be **connected**. If $\mathfrak{p}$ contains no connection, then $\mathfrak{p}$ is **unconnected**.*

In order to construct the set of paths through a particular matrix, we employ the following two functions defined on sets of paths:

Definition 2.8. *If X and Y are sets of paths, then $X \oplus Y$ and $X \otimes Y$ are defined as follows:*

$$X \oplus Y := X \cup Y,$$
$$X \otimes Y := \{\mathfrak{p} \cup \mathfrak{q} \mid \mathfrak{p} \in X \text{ and } \mathfrak{q} \in Y\}.$$

Definition 2.9. *If $\mathcal{A}$ is a matrix then the set of **paths through** $\mathcal{A}$, denoted by $[\![\mathcal{A}]\!]$, is defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then $[\![\mathcal{A}]\!] = \{\varnothing\}$.*
2. *If $\mathcal{A}$ is p^i or $\neg p^i$ then $[\![\mathcal{A}]\!] = \{\{\mathcal{A}\}\}$.*
3. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$.*
4. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$.*

Computing the paths through a matrix effectively amounts to converting the matrix into a disjunctive normal form. We can also think of paths as representing branches in a fully expanded tableau for the formula underlying the matrix.

Example 2.2. *Consider the matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4$, which is unsatisfiable and has the following paths:*

$$\begin{aligned} &[\![p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4]\!] \\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![q^3]\!]) \otimes [\![\neg q^4]\!] \\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\{q^3\}\}) \otimes \{\{\neg q^4\}\} \\ &= \{\{p^1\}\} \otimes \{\{\neg p^2\}, \{q^3\}\} \otimes \{\{\neg q^4\}\} \\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, q^3, \neg q^4\}\}. \end{aligned}$$

Observe that the first path contains the connection $\{p^1, \neg p^2\}$ whereas the second path contains the connection $\{q^3, \neg q^4\}$, so that every path is connected. Recalling the graphical notation

$$p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4 = \begin{bmatrix} p^1 & \begin{bmatrix} \neg p^2 \\ q^3 \end{bmatrix} & \neg q^4 \end{bmatrix},$$

we see that the paths through a matrix correspond to horizontal lines drawn across the matrix which intersect one literal from every column.

That every path through our example matrix is connected, and the matrix itself is unsatisfiable, is not a coincidence. At the heart of the connection method in (Bibel 1981; Andrews 1976) is a theorem stating that a matrix is unsatisfiable if and only if every path through the matrix is connected. Although this characterisation is well known, given our modified definitions for matrices and paths, we take a moment to prove this result for the convenience of the reader. We start with the following lemma:

Lemma 2.1. *An interpretation ν satisfies a matrix $\mathcal{A}$ if and only if for some path $\mathfrak{p}$ through $\mathcal{A}$ it follows that ν satisfies every element of $\mathfrak{p}$.*

Proof. Suppose $\mathcal{A}$ is a matrix and ν is an interpretation, and proceed by induction on the complexity of $\mathcal{A}$ followed by case analysis on the primary connective of the matrix underlying $\mathcal{A}$.

1. In the case $\mathcal{A}$ is $\top$ then it follows that ν satisfies $\mathcal{A}$, and ν satisfies every element of the single path $\varnothing \in [\![\mathcal{A}]\!]$.
2. In the case $\mathcal{A}$ is a literal p^i or $\neg p^i$, then ν satisfies $\mathcal{A}$ if and only if it satisfies every element of the path $\{\mathcal{A}\}$. As $[\![\mathcal{A}]\!] = \{\{\mathcal{A}\}\}$ the conclusion follows.
3. In the case $\mathcal{A}$ is a disjunction $(\mathcal{B} \vee \mathcal{C})$ suppose ν is a valuation satisfying $\mathcal{A}$. It follows that ν satisfies either $\mathcal{B}$ or $\mathcal{C}$, and therefore by the induction hypothesis there either exists a path $\mathfrak{p} \in [\![\mathcal{B}]\!]$ such that ν satisfies every element of $\mathfrak{p}$, or there exists a path $\mathfrak{p} \in [\![\mathcal{C}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$, it follows in either case that there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$ as required. Conversely, suppose that ν is a valuation such that there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$ it follows that either $\mathfrak{p} \in [\![\mathcal{B}]\!]$ in which case the induction hypothesis shows that ν satisfies $\mathcal{B}$, or $\mathfrak{p} \in [\![\mathcal{C}]\!]$ in which case the induction hypothesis shows that ν satisfies $\mathcal{C}$. In either case, it follows that ν satisfies $\mathcal{A} = (\mathcal{B} \vee \mathcal{C})$ so the conclusion follows.
4. In the case $\mathcal{A}$ is a conjunction $(\mathcal{B} \wedge \mathcal{C})$ suppose ν is a valuation satisfying $\mathcal{A}$. It follows that ν satisfies both $\mathcal{B}$ and $\mathcal{C}$, and therefore by the induction hypothesis there exists a path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that ν satisfies every element of $\mathfrak{q}$, and there also exists a path $\mathfrak{r} \in [\![\mathcal{C}]\!]$ such that ν satisfies every element of $\mathfrak{r}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$ it follows that $\mathfrak{p} = \mathfrak{q} \cup \mathfrak{r}$ is a path through $\mathcal{A}$ such that ν satisfies every element of $\mathfrak{p}$, as required. Conversely, suppose that ν is a valuation for which there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$ it follows that there exist paths $\mathfrak{q} \in [\![\mathcal{B}]\!]$ and $\mathfrak{r} \in [\![\mathcal{C}]\!]$ such that $\mathfrak{p} = \mathfrak{q} \cup \mathfrak{r}$. Therefore, as ν satisfies every element of $\mathfrak{q}$ and every element of $\mathfrak{r}$, by applying the induction hypothesis it follows that ν satisfies $\mathcal{B}$ and $\mathcal{C}$, which is to say ν satisfies $\mathcal{A} = (\mathcal{B} \wedge \mathcal{C})$ so the conclusion follows.

□

Theorem 2.1. *A matrix $\mathcal{A}$ is unsatisfiable if and only if every path through $\mathcal{A}$ is connected.*

Proof. Suppose that $\mathcal{A}$ is unsatisfiable. Assume for the sake of contradiction that there exists a path $\mathfrak{p}$ through $\mathcal{A}$ which contains no connection. Then consider an interpretation ν such that $\nu(p) = T$ if $p^i \in \mathfrak{p}$, $\nu(p) = F$ if $\neg p^i \in \mathfrak{p}$, with $\nu(p)$ chosen arbitrarily otherwise. It follows that ν satisfies every element of the path $\mathfrak{p}$, and hence by the prior Lemma 2.1 ν satisfies $\mathcal{A}$. This is a contradiction, so it must be that every path through $\mathcal{A}$ was connected.

Conversely, suppose every path through $\mathcal{A}$ is connected and assume for the sake of contradiction that ν is an interpretation satisfying $\mathcal{A}$. It follows that there exists a path $\mathfrak{p}$ such that ν satisfies every element of $\mathfrak{p}$. However, because every path is connected, $\mathfrak{p}$ is connected and therefore there exists a variable p alongside tags i and j such that $\mathfrak{p}$ contains both p^i and $\neg p^j$. However, this means that $\nu(p) = T$ and $\nu(p) = F$ which is a contradiction. Therefore, $\mathcal{A}$ must be unsatisfiable. □

2.5 Explicit and Implicit Beliefs

Requiring belief contraction operators to be invariant under logical equivalence is unreasonable when studying resource-limited agents. It becomes impossible to differentiate between the beliefs the agent holds, and the logical consequences of those beliefs. Effectively, every consequence of its beliefs must be treated as if it is instantaneously known, and any contradictory beliefs results in every sentence being believed.

These sorts of concerns motivated (Levesque 1984) to differentiate between the *explicit beliefs* which an agent possesses, and those *implicit beliefs* which it would be able to conclude based off of inferences from its explicit beliefs given adequate time.

One concern is that explicitly believing $\mathcal{A} \wedge \mathcal{B}$ seems to imply one should explicitly believe $\mathcal{A}$ as well. Hence, there is a need for an intermediate approach, wherein certain immediate consequences of explicit beliefs are regarded as among the explicit beliefs, while consequences involving more elaborate inferences are relegated to the category of implicit belief.

Definition 2.10. *Matrices $\mathcal{A}$ and $\mathcal{B}$ are **path-equivalent** iff $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$.*

Example 2.3. *The matrices $p^1 \vee (q^2 \vee r^3)$ and $r^3 \vee (q^2 \vee p^1)$ are path-equivalent, whereas the matrices $p^1 \vee \neg p^2$ and $\top$ are not path-equivalent.*

In our approach, almost everything is invariant under path-equivalence, or can be chosen to be so. It follows from Lemma 2.1 that path-equivalence implies logical equivalence, however path-equivalence is far more restrictive. We consider path-equivalence to offer an intermediary between completely syntax-insensitive approaches which fail to differentiate implicit and explicit beliefs, and completely syntax-sensitive approaches which risk becoming ad-hoc.

3 Path-Contraction via Matrix Attenuation

In this section we introduce the class of *path-contraction operators* which operate by applying selective substitutions of $\top$ or $\bot$ for particular occurrences of literals variables within a matrix $\mathcal{K}$ in order to construct a matrix $\mathcal{K}'$ which does not entail another matrix $\mathcal{A}$. We refer to this process of selective substitution as *matrix attenuation*. An advantage of matrix attenuation is that it amounts to a straightforward edit to the original knowledge base, without any requirement of a costly conversion to a conjunctive or disjunctive normal form. Hence, the path-contraction operators we obtain leave the structure of the knowledge bases being contracted relatively unchanged.

Definition 3.1. *The **attenuation of** a matrix $\mathcal{A}$ **at** a tag i is the matrix $\mathcal{A}_i$ defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then $\mathcal{A}_i = \top$.*
2. *If $\mathcal{A}$ is p^j or $\neg p^j$ and $i \neq j$ then $\mathcal{A}_i = \mathcal{A}$.*
3. *If $\mathcal{A}$ is p^j or $\neg p^j$ and $i = j$ then $\mathcal{A}_i = \top$.*
4. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then $\mathcal{A}_i = (\mathcal{B}_i \vee \mathcal{C}_i)$.*
5. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then $\mathcal{A}_i = (\mathcal{B}_i \wedge \mathcal{C}_i)$.*

Simply, $\mathcal{A}_i$ is the matrix obtained from $\mathcal{A}$ by replacing the literal containing the variable tagged by i in $\mathcal{A}$ with $\top$, or by doing nothing when the tag i does not appear in $\mathcal{A}$. We think of this as replacing the specific variable occurrence tagged by i in $\mathcal{A}$ with $\top$ if it appears positively, or with $\bot$ if it appears negatively.

Example 3.1. *Suppose $\mathcal{K}$ is the matrix $p^1 \wedge (\neg p^2 \vee q^3)$ which logically entails q. If we attenuate at the tag 2 we get $\mathcal{K}_2 = p^1 \wedge (\top \vee q^3)$ which no longer entails q. Note that $\mathcal{K} \vdash \mathcal{K}_2$, so we can think of $\mathcal{K}_2$ as a contraction of $\mathcal{K}$ by q.*

It turns out that $\mathcal{A}$ always entails $\mathcal{A}_i$. This can be proven directly via an induction, or as a consequence of Theorems 5.2 and 5.1 below. However, for now we merely state this as an observation:

Observation 3.1. *If $\mathcal{A}$ is a matrix and i is a tag then $\mathcal{A} \vdash \mathcal{A}_i$.*

It is possible to characterise the paths through an attenuation of a matrix as being attenuations of the paths through the matrix itself, where attenuations of paths are defined as follows:

Definition 3.2. *The **attenuation of** a path $\mathfrak{p}$ **at** a tag i is the path $\mathfrak{p}_i$ consisting of those literals in $\mathfrak{p}$ not containing the tag i.*

Theorem 3.1. *If $\mathcal{A}$ is a matrix and i is a tag then $[\![\mathcal{A}_i]\!] = \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}$.*

Proof. Proceed by induction on the complexity of $\mathcal{A}$, and within the induction by case analysis.

1. If $\mathcal{A}$ is $\top$ then $[\![\mathcal{A}]\!] = \{\varnothing\}$ and $\mathcal{A}_i = \mathcal{A}$ so it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\top]\!]\\ &= \{\varnothing\}\\ &= \{\varnothing_i\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\varnothing\}\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

2. If $\mathcal{A}$ is p^j or $\neg p^j$ then there are two cases. In the case $i = j$ then $\mathcal{A}_i = \top$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\top]\!]\\ &= \{\varnothing\}\\ &= \{\{\mathcal{A}\}_i\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\{\mathcal{A}\}\}\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

In the case $i \neq j$ then $\mathcal{A}_i = \mathcal{A}$ and it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= \{\{\mathcal{A}\}\}\\ &= \{\{\mathcal{A}\}_i\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\{\mathcal{A}\}\}\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

In either case, our choice satisfies the requirement.

3. If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then by the induction hypothesis $[\![\mathcal{B}_i]\!] = \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\}$ and $[\![\mathcal{C}_i]\!] = \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}$. Observing $\mathcal{A}_i = (\mathcal{B}_i \vee \mathcal{C}_i)$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\mathcal{B}_i \vee \mathcal{C}_i]\!]\\ &= [\![\mathcal{B}_i]\!] \cup [\![\mathcal{C}_i]\!]\\ &= \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\} \cup \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{B}]\!] \cup [\![\mathcal{C}]\!]\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

Thus, the requirement is satisfied.

4. If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then by the induction hypothesis $[\![\mathcal{B}_i]\!] = \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\}$ and $[\![\mathcal{C}_i]\!] = \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}$. Observing $\mathcal{A}_i = (\mathcal{B}_i \wedge \mathcal{C}_i)$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\mathcal{B}_i \wedge \mathcal{C}_i]\!]\\ &= [\![\mathcal{B}_i]\!] \otimes [\![\mathcal{C}_i]\!]\\ &= \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\} \otimes \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}\\ &= \{\mathfrak{q}_i \cup \mathfrak{r}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!], \mathfrak{r} \in [\![\mathcal{C}]\!]\}\\ &= \{(\mathfrak{q} \cup \mathfrak{r})_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!], \mathfrak{r} \in [\![\mathcal{C}]\!]\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]\}\\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

Thus, the requirement is satisfied.

□

Example 3.2. *Consider the matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)$ which logically entails r, and has the following paths:*

$$\begin{aligned}&[\![p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)]\!]\\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![q^3]\!]) \otimes ([\![\neg q^4]\!] \oplus [\![r^5]\!])\\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\{q^3\}\}) \otimes (\{\{\neg q^4\}\} \oplus \{\{r^5\}\})\\ &= \{\{p^1\}\} \otimes \{\{\neg p^2\}, \{q^3\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}\\ &= \{\{p^1, \neg p^2\}, \{p^1, q^3\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}\\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, q^3, \neg q^4\},\\ &\qquad \{p^1, \neg p^2, r^5\}, \{p^1, q^3, r^5\}\}.\end{aligned}$$

Attenuating this matrix at the tag 3, we obtain the following paths:

$$\begin{aligned}&[\![p^1 \wedge (\neg p^2 \vee \top) \wedge (\neg q^4 \vee r^5)]\!]\\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![\top]\!]) \otimes ([\![\neg q^4]\!] \oplus [\![r^5]\!])\\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\varnothing\}) \otimes (\{\{\neg q^4\}\} \oplus \{\{r^5\}\})\\ &= \{\{p^1\}\} \otimes \{\{\neg p^2\}, \varnothing\} \otimes \{\{\neg q^4\}, \{r^5\}\}\\ &= \{\{p^1, \neg p^2\}, \{p^1\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}\\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, \neg q^4\}, \{p^1, \neg p^2, r^5\}, \{p^1, r^5\}\}\end{aligned}$$

Notice that it is possible to build a valuation satisfying $\{p^1, \neg q^4\}$ but not r. Therefore, it follows that via attenuation we have prevented the logical entailment of r.

In the subsequent development we make use of iterated attenuations:

Definition 3.3. *If $I = \{i_1, i_2, \ldots, i_k\}$ is a finite set of tags and $\mathcal{A}$ is a matrix then the* ***attenuation of*** $\mathcal{A}$ ***by*** I *is defined as the iterated attenuation $(\ldots((\mathcal{A}_{i_1})_{i_2})\ldots)_{i_k}$.*

That this definition is well-defined follows from the following observation:

Observation 3.2. *If i and j are tags and $\mathcal{A}$ is a matrix then $(\mathcal{A}_i)_j = (\mathcal{A}_j)_i$ and $(\mathcal{A}_i)_i = \mathcal{A}_i$.*

Path-contraction operators will compute a contraction of $\mathcal{K}$ by $\mathcal{A}$ via attenuating a number of tags within a matrix $\mathcal{K}$ to obtain a matrix $\mathcal{K}_I$ which does not entail a matrix $\mathcal{A}$. We refer to these sets of tags I as *cuttings*.

Definition 3.4. *If $\mathcal{K}$ and $\mathcal{A}$ are tag-disjoint matrices then a* ***cutting*** *of $\mathcal{K}$ by $\mathcal{A}$ is either $\varnothing$ in the case $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$, or a subset I of $T(\mathcal{K})$ such that $\mathcal{K}_I \nvdash \mathcal{A}$ otherwise. It is a* ***regular cutting*** *iff every tag in I tags a variable in $\mathcal{K}$ which also appears in $\mathcal{A}$.*

Example 3.3. *In our previous example of $\mathcal{K} = p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)$ it follows that $\{3\}$ is a non-regular cutting of $\mathcal{K}$ by r whereas $\{5\}$ is a regular cutting of $\mathcal{K}$ by r.*

We show in Section 5 that working with regular cuttings results in an analogue of Parikh's Postulate being satisfied. It is conceivable that additional restrictions on cuttings may prove desirable. For instance, whenever I and J are sets of tags with $I \subseteq J$ then $\mathcal{K}_I \vdash \mathcal{K}_J$ and hence in the case $\mathcal{K}_I \nvdash \mathcal{A}$ it follows that $\mathcal{K}_J \nvdash \mathcal{A}$ as well. Seeking belief base contraction operators which result in minimal change, i.e. which preserve as many of the existing beliefs as possible, suggests that we should always prefer $\mathcal{K}_I$ to $\mathcal{K}_J$, in effect imposing a requirement that a cutting must be minimal with respect to set inclusion. However, this would increase the complexity of computing a cutting, and thus we do not take this to be a defining feature of our approach. We leave the question of additional restrictions on cuttings to the designers of concrete path-contraction operators.

Definition 3.5. *A binary function $- : M \times M \to M$ is a* ***path-contraction operator*** *iff for all satisfiable $\mathcal{K}$ and $\mathcal{A}$ it follows that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some cutting I of $\mathcal{K}$ by $\mathcal{A}$. It is* ***regular*** *in the case $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some regular cutting I.*

Note our restriction to satisfiable formulae in the prior definition. Although in practice sufficiently-complex knowledge bases will likely have inconsistencies which require repair, we consider this a separate issue which path-contraction operators will not be responsible for addressing.

Path-contraction operators preserve the syntactic structure of the original knowledge base for the reason that matrix attenuation preserves the syntactic structure of a matrix, and $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some cutting I of $\mathcal{K}$ by $\mathcal{A}$. Additionally, path-contraction operators satisfy analogues of the postulates for belief base contraction in Definition 2.1.

Theorem 3.2. *Suppose that $-$ is a path-contraction operator, then the following properties are satisfied:*

C1. If $\nvdash \mathcal{A}$ then $\mathcal{K} - \mathcal{A} \nvdash \mathcal{A}$.

C2. If $\vdash \mathcal{A}$ then $\mathcal{K} - \mathcal{A} = \mathcal{K}$.

C3. $\mathcal{K} \vdash \mathcal{K} - \mathcal{A}$.

C4. If $\mathcal{K} \nvdash \mathcal{A}$ then $\mathcal{K} - \mathcal{A} = \mathcal{K}$.

Proof.

1. In the case $\nvdash \mathcal{A}$ let I be a cutting of $\mathcal{K}$ by $\mathcal{A}$ such that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$. Being that $\nvdash \mathcal{A}$ it follows from the definition of a cutting that I is a set of tags such that $\mathcal{K}_I \nvdash \mathcal{A}$, which is to say $\mathcal{K} - \mathcal{A} \nvdash \mathcal{A}$.
2. In this case $\vdash \mathcal{A}$ it follows by definition that $\varnothing$ is the only cutting of $\mathcal{K}$ by $\mathcal{A}$, and hence $\mathcal{K} - \mathcal{A} = \mathcal{K}_\varnothing = \mathcal{K}$.
3. It follows that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ and therefore $\mathcal{K} \vdash \mathcal{K}_I = \mathcal{K} - \mathcal{A}$ by Theorem 5.2 and Theorem 5.1 below.
4. In the case $\mathcal{K} \nvdash \mathcal{A}$ then $\varnothing$ is the only cutting of $\mathcal{K}$ by $\mathcal{A}$. Letting I be the cutting such that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ it then follows that $I = \varnothing$ showing $\mathcal{K} - \mathcal{A} = \mathcal{K}_\varnothing = \mathcal{K}$.

□

It follows that every path-contraction operator produces a belief base contraction operator in the following manner. Given κ and ϕ choose tag-disjoint matrices $\mathcal{K}$ and $\mathcal{A}$ such that κ is the detagging of $\mathcal{K}$ and ϕ is the detagging of $\mathcal{A}$. Apply the path-contraction operator to compute $\mathcal{K} - \mathcal{A}$, then define $\kappa - \phi$ as the detagging of $\mathcal{K} - \mathcal{A}$. It follows by Theorem 3.2 that the binary function $-$ on propositional formulae defined above satisfies the requirements to be a belief base contraction operator. Example 4.3 in the next section shows this in action.

4 An Algorithm for Regular Path-Contraction

In this section we present an algorithm for implementing a regular path-contraction operator $-$, and show that this algorithm can be used to decide whether $\mathcal{K} - \mathcal{A} \vdash \mathcal{B}$ with complexity **DP**, where **DP** $= \{L_1 \cap L_2 \mid L_1 \in$ **NP**$, L_2 \in$ **coNP**$\}$. Further, in the case $\mathcal{A}$ is not a tautology, computing $\mathcal{K} - \mathcal{A}$ has complexity **FNP**[1]. Our algorithm makes use of the notion of an *extracted path*, and the notion of a *cross-cut*, which we now present.

Definition 4.1. *If $\mathcal{A}$ is a satisfiable matrix with satisfying assignment ν, then the* ***extracted path*** $\mathrm{ext}(\mathcal{A}, \nu)$ *is defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then $\mathrm{ext}(\mathcal{A}, \nu) = \varnothing$.*
2. *If $\mathcal{A}$ is p^i or $\neg p^i$ then $\mathrm{ext}(\mathcal{A}, \nu) = \{\mathcal{A}\}$.*
3. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{B}, \nu) \cup \mathrm{ext}(\mathcal{C}, \nu)$.*
4. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ and ν satisfies $\mathcal{B}$ then $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{B}, \nu)$, otherwise $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{C}, \nu)$.*

Example 4.1. *The matrix $\mathcal{K}$ defined as $(p^1 \wedge \neg q^2) \vee (\neg p^3 \wedge q^4)$ has the paths $\{p^1, \neg q^2\}$ and $\{\neg p^3, q^4\}$. The satisfying assignment μ for $\mathcal{K}$ given by setting $\mu(p) = T$ and $\mu(q) = F$ results in $\mathrm{ext}(\mathcal{K}, \nu) = \{p^1, \neg q^2\}$ whereas the satisfying assignment ν for $\mathcal{K}$ given by setting $\nu(p) = F$ and $\nu(q) = T$ results in $\mathrm{ext}(\mathcal{K}, \nu) = \{\neg p^3, q^4\}$.*

Lemma 4.1. *If ν is an assignment satisfying $\mathcal{A}$ then $\mathrm{ext}(\mathcal{A}, \nu)$ is an unconnected path through $\mathcal{A}$.*

[1] **FNP** is the function problem version of **NP** consisting, roughly, of those functions which may be evaluated in polynomial-time on a non-deterministic Turing machine. For example, **FSAT**, the problem of finding a satisfying assignment, is **FNP**-complete.

Definition 4.2. *If $\mathfrak{p}$ and $\mathfrak{q}$ are unconnected paths then the* ***cross-cut****, denoted by $\mathrm{cr}(\mathfrak{p}, \mathfrak{q})$, is the set of tags i in $\mathfrak{p}$ for which there exists a tag j in $\mathfrak{q}$ alongside a propositional variable p such that either $\{p^i, \neg p^j\}$ or $\{\neg p^i, p^j\}$ is a connection in $\mathfrak{p} \cup \mathfrak{q}$.*

Example 4.2. *Consider the matrices $\mathcal{K}$ and $\mathcal{A}$ defined as $p^1 \wedge (\neg p^2 \vee q^3)$ and q^4 respectively. It follows that $\mathfrak{p} = \{p^1, q^3\}$ is an unconnected path through $\mathcal{K}$. It also follows that $\mathfrak{q} = \{\neg q^4\}$ is an unconnected path through $\neg\mathcal{A} = \neg q^4$. In this case their cross-cut is $\mathrm{cr}(\mathfrak{p}, \mathfrak{q}) = \{3\}$, and it follows that $\mathcal{K}_3 = p^1 \wedge (\neg p^2 \vee \top)$ which does not entail $\mathcal{A} = q^4$.*

Lemma 4.2. *If $\mathcal{K} \vdash \mathcal{A}$, $\mathfrak{p}$ is an unconnected path through $\mathcal{K}$, and $\mathfrak{q}$ is an unconnected path through $\neg\mathcal{A}$ then $\mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$.*

Proof. Suppose $\mathfrak{p}$ is an unconnected path through $\mathcal{K}$ and $\mathfrak{q}$ is an unconnected path through $\neg\mathcal{A}$. It follows that $\mathfrak{r} = \mathfrak{p} \cup \mathfrak{q}$ is a path through $\mathcal{K} \wedge \neg\mathcal{A}$. By the assumption that $\mathfrak{p}$ and $\mathfrak{q}$ are unconnected, it follows that every connection in $\mathfrak{r}$ may be written as $\{\ell^i, \ell^j\}$ where ℓ^i is in $\mathcal{K}$ and ℓ^j is in $\neg\mathcal{A}$. Enumerating these connections as $\{\ell^{i_1}, \ell^{j_1}\}, \{\ell^{i_2}, \ell^{j_2}\}, \ldots, \{\ell^{i_n}, \ell^{j_n}\}$ it follows that $\mathrm{cr}(\mathfrak{p}, \mathfrak{q}) = \{i_1, i_2, \ldots, i_n\}$. Letting $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ it follows that $\mathfrak{r}_I$ is a unconnected path through $(\mathcal{K} \wedge \neg\mathcal{A})_I$. Being that I contains only tags appearing in $\mathcal{K}$ it follows that $(\mathcal{K} \wedge \neg\mathcal{A})_I = \mathcal{K}_I \wedge \neg\mathcal{A}$. Therefore $\mathfrak{r}_I$ is an unconnected path through $\mathcal{K}_I \wedge \neg\mathcal{A}$. By Theorem 2.1 it follows that $\mathcal{K}_I \wedge \neg\mathcal{A}$ is satisfiable, showing that $\mathcal{K}_I \nvdash \mathcal{A}$. If $\vdash \mathcal{A}$ then there would exist no unconnected path through $\neg\mathcal{A}$, contradicting our hypotheses. If $\mathcal{K} \nvdash \mathcal{A}$ this would also contradict our hypothesis. Hence, $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a cutting of $\mathcal{K}$ by $\mathcal{A}$. Furthermore, as every tag in I appears in $\mathcal{K}$, it follows that I is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$. □

With these tools in hand, we can now present the Path-Contraction Algorithm. Given matrices $\mathcal{K}$ and $\mathcal{A}$ such that $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$, the Path-Contraction Algorithm will choose an unconnected path $\mathfrak{p}$ through $\mathcal{K}$ alongside an unconnected path $\mathfrak{q}$ through $\neg\mathcal{A}$. This is accomplished by using a satisfiability solver to construct satisfying assignments for $\mathcal{K}$ and $\neg\mathcal{A}$, then defining $\mathfrak{p}$ and $\mathfrak{q}$ as the extracted paths corresponding to their respective satisfying assignments. Having chosen these paths, the algorithm computes the cross-cut $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ which is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$, then it returns $\mathcal{K}_I$. In pseudo-code, we have the following:

Algorithm 1 Path-Contraction Algorithm

Input: Initial belief base $\mathcal{K}$
Input: Belief to contract $\mathcal{A}$
Output: The contracted belief base $\mathcal{K} - \mathcal{A}$

```
if $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$ then
    return $\mathcal{K}$
else
    $\nu$ := a satisfying assignment for $\mathcal{K}$
    $\mu$ := a satisfying assignment for $\neg\mathcal{A}$
    $\mathfrak{p} := \mathrm{ext}(\mathcal{K}, \nu)$
    $\mathfrak{q} := \mathrm{ext}(\neg\mathcal{A}, \mu)$
    $I := \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$
    return $\mathcal{K}_I$
end if
```

Depending on the satisfiability solver used, this algorithm produces different path-contraction operators. Further, if the satisfiability solver is non-deterministic, as many systems with randomised restarts are, then the path-contraction operator produced is non-deterministic. However, fixing a deterministic satisfiability solver, we obtain a deterministic algorithm, and thus a well-defined path-contraction operator. Before turning to the correctness and complexity of this algorithm, we present the following example:

Example 4.3. *Consider a propositional vocabulary where p symbolises that Tweety is a penguin, b symbolises that Tweety is a bird, f symbolises that Tweety flies, and n symbolises that Tweety builds nests. Consider the naive belief base κ defined as $(p \to b) \wedge (b \to f) \wedge (b \to n)$ which has the undesirable consequence $\phi := (p \to f)$ suggesting that were Tweety a penguin then Tweety could fly. In order to apply our Path-Contraction Algorithm to compute $\kappa - \phi$, we first convert κ and ϕ to negation normal form, and tag everything to obtain the matrices $\mathcal{K} := (\neg p^1 \vee b^2) \wedge (\neg b^3 \vee f^4) \wedge (\neg b^5 \vee n^6)$ and $\mathcal{A} := (\neg p^7 \vee f^8)$.*

As $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$ our algorithm must do some work. We start by choosing a truth-value assignment ν satisfying $\mathcal{K}$ such as the one satisfying $p \wedge b \wedge f \wedge n$, alongside a truth-value assignment μ satisfying $\neg\mathcal{A} = p^7 \wedge \neg f^8$ such as the one satisfying $p \wedge b \wedge \neg f \wedge n$. Using these assignments, we extract the path $\mathfrak{p} = \{b^2, f^4, n^6\}$ through $\mathcal{K}$ alongside the path $\mathfrak{q} = \{p^7, \neg f^8\}$ through $\neg\mathcal{A}$.

Computing the cross-cut, we find $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q}) = \{4\}$. Our algorithm now returns

$$\mathcal{K}_I = (\neg p^1 \vee b^2) \wedge (\neg b^3 \vee \top) \wedge (\neg b^5 \vee n^6)$$

whose detagging is $(\neg p \vee b) \wedge (\neg b \vee n)$. This is just the negation normal form of $(p \to b) \wedge (b \to n)$. Note that the beliefs regarding penguins being birds, and birds building nests, have been preserved.

Theorem 4.1. *The Path-Contraction Algorithm defines a regular path-contraction operator.*

Proof. For any satisfiable matrices $\mathcal{K}$ and $\mathcal{A}$ let $\mathcal{K} - \mathcal{A}$ be defined as the result returned by the Path-Contraction Algorithm. This is well-defined as the Path-Contraction Algorithm is deterministic once we fix deterministic satisfiability solvers, and always terminates regardless. We must show that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some regular cutting I of $\mathcal{K}$ by $\mathcal{A}$. In the case $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$ then, by definition, $I = \varnothing$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$, and furthermore the algorithm returns $\mathcal{K} = \mathcal{K}_I$. Otherwise, we proceed under the assumption $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$. It follows that there exists a truth value assignment ν satisfying $\mathcal{K}$ as well as a truth value assignment μ satisfying $\neg\mathcal{A}$. By Lemma 4.1 it follows that $\mathfrak{p} = \mathrm{ext}(\nu, \mathcal{K})$ is an unconnected path through $\mathcal{K}$, and $\mathfrak{q} = \mathrm{ext}(\mu, \neg\mathcal{K})$ is an unconnected path through $\mathcal{A}$. Further, recalling our assumption that $\mathcal{K} \vdash \mathcal{A}$ it follows by Lemma 4.2 that $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$. In this case, the algorithm returns with $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$. Hence, in every case the algorithm returns $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some regular cutting I of $\mathcal{K}$ by $\mathcal{A}$, and therefore it follows that $-$ is a regular path-contraction operator. □

Theorem 4.2. *If $-$ is a regular path-contraction operator defined by the Path-Contraction Algorithm, then the decision problem of deciding whether $\mathcal{K} - \mathcal{A} \vdash \mathcal{B}$ is in the class* ***DP****.*

Proof. Let n be the sum of the size of $\mathcal{K}$, the size of $\mathcal{A}$, the size of $\mathcal{B}$, and the number of variables in the language. We start by applying the Path-Contraction Algorithm to compute $\mathcal{K} - \mathcal{A}$. Initially the algorithm checks whether $\mathcal{K} \nvdash \mathcal{A}$ using a satisfiability solver, and if this is the case then returns $\mathcal{K} - \mathcal{A} = \mathcal{K}$. Otherwise, the algorithm next checks whether $\vdash \mathcal{A}$ using a theorem prover, and if this is the case it also returns $\mathcal{K} - \mathcal{A} = \mathcal{K}$. In the case $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$, the algorithm proceeds to call a satisfiability solver to obtain satisfying assignments ν and μ on lines (4) and (5). Recursively labelling every subformula of $\mathcal{K}$ and $\mathcal{A}$ by its value assigned under ν and μ respectively costs $O(n \log n)$ time, and after the paths $\mathfrak{p} = \mathrm{ext}(\nu, \kappa)$ and $\mathfrak{q} = \mathrm{ext}(\mu, \neg\phi)$ are computed on lines (6) and (7) using only $O(n)$ time. Computing $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ on line (8) can be done in time $O(n^2)$, and computing $\mathcal{K}_I$ on line (9) can be done in time $O(n \log n)$, after which the algorithm returns $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$.

In total, a satisfiability solver has been called at most three times, and a theorem prover has been called at most once. Deciding whether $\mathcal{K} - \mathcal{A} \vdash \mathcal{B}$ may now be accomplished with one more call to a theorem prover. In total, this decision procedure shows the decision problem to be in the complexity class **DP**. □

It is worthwhile observing that when $\mathcal{A}$ is assumed to not be a tautology, then the check for whether $\vdash \mathcal{A}$ on line (1) of the Path-Contraction Algorithm can be omitted, so that it computes $\mathcal{K} - \mathcal{A}$ with only three calls to a satisfiability solver and additional work taking only polynomial-time on a deterministic machine. This gives an overall complexity of **FNP** for computing $\mathcal{K} - \mathcal{A}$ when $\mathcal{A}$ is assumed to not be a tautology. The existence of a regular path-contraction operator with a worst-case time complexity of **FNP** is in striking contrast to other concrete belief change functions discussed in (Eiter and Gottlob 1992) whose complexity is often Π_p^2-complete. Though, it must be pointed out that path-contraction operators comprise a class of operators of which the Path-Contraction Algorithm contributes only a small subset, and it may ultimately prove desirable to sacrifice some additional performance in order to ensure stronger guarantees over the properties of the overall path-contraction operator.

5 Path-Entailment and Path-Independence

In this section we introduce the notion of *path-entailment* which strengthens logical entailment to a structural property of matrices. Our central results are that every matrix path-entails its attenuations, and further that attenuation preserves the path-entailment of matrices not containing the attenuated tag. Using path-entailment, we then show that regular path-contraction operators satisfy preservation properties analogous to Parikh's relevance postulate.

Definition 5.1. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices such that for every path $\mathfrak{p} \in \llbracket\mathcal{A}\rrbracket$ there exists a path $\mathfrak{q} \in \llbracket\mathcal{B}\rrbracket$ with $\mathfrak{p} \supseteq \mathfrak{q}$ then $\mathcal{A}$* ***path-entails*** *$\mathcal{B}$, which we indicate by writing $\mathcal{A} \Vdash \mathcal{B}$. We also say that $\mathcal{B}$ is a* ***path-consequence*** *of $\mathcal{A}$.*

Example 5.1. *Consider the matrix $\mathcal{K}$ defined as $(\neg p^1 \vee q^2) \wedge (\neg q^3 \vee r^4)$. The paths through $\mathcal{K}$ are as follows: $\{\neg p^1, \neg q^3\}$, $\{\neg p^1, r^4\}$, $\{q^2, \neg q^3\}$, and finally $\{q^2, r^4\}$. As the paths through $\neg p^1 \vee q^2$ are $\{\neg p^1\}$ and $\{q^2\}$ it follows that $\mathcal{K}$ path-entails $\neg p^1 \vee q^2$, as each path through $\mathcal{K}$ contains either $\neg p^1$ or q^2. However, despite $\neg p^1 \vee r^4$ being logically entailed by $\mathcal{K}$, this is not a path-consequence of $\mathcal{K}$ for the reason that the path $\{q^2, \neg q^3\}$ contains no path through $\neg p^1 \vee r^4$.*

Theorem 5.1. *If $\mathcal{A} \Vdash \mathcal{B}$ then $\mathcal{A} \vdash \mathcal{B}$.*

Proof. Suppose that $\mathcal{A} \Vdash \mathcal{B}$, and consider an interpretation ν which satisfies $\mathcal{A}$. By Lemma 2.1 this means that there exists a path $\mathfrak{p} \in \llbracket\mathcal{A}\rrbracket$ such that ν satisfies every element of $\mathfrak{p}$. Under our assumption that $\mathcal{A} \Vdash \mathcal{B}$ it follows that there exists some path $\mathfrak{q} \in \llbracket\mathcal{B}\rrbracket$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. However, this means that ν satisfies every element of $\mathfrak{q}$, which by Lemma 2.1 implies ν satisfies $\mathcal{B}$. With ν being arbitrary, it follows that $\mathcal{A} \vdash \mathcal{B}$. □

Theorem 5.2.

1. $\mathcal{A} \Vdash \mathcal{A}$.
2. If $\mathcal{A} \Vdash \mathcal{B}$ and $\mathcal{B} \Vdash \mathcal{C}$ then $\mathcal{A} \Vdash \mathcal{C}$.
3. If $\mathcal{A} \Vdash \mathcal{C}$ and $\mathcal{B}$ is tag-disjoint with $\mathcal{A}$ then $\mathcal{A} \wedge \mathcal{B} \Vdash \mathcal{C}$.
4. If $\mathcal{A}$ is a matrix and i is a tag then $\mathcal{A} \Vdash \mathcal{A}_i$.
5. If $\mathcal{A} \Vdash \mathcal{B} \wedge \mathcal{C}$ then $\mathcal{A} \Vdash \mathcal{B}$.
6. If $\mathcal{A} \Vdash \mathcal{B}$ then $\mathcal{A} \Vdash \mathcal{B} \vee \mathcal{C}$.

Proof.

1. Immediate.
2. Suppose $\mathcal{A} \Vdash \mathcal{B}$ and $\mathcal{B} \Vdash \mathcal{C}$. Suppose that $\mathfrak{p} \in \llbracket\mathcal{A}\rrbracket$ and observe that $\mathcal{A} \Vdash \mathcal{B}$ implies there exists some $\mathfrak{q} \in \llbracket\mathcal{B}\rrbracket$ such that $\mathfrak{p} \supseteq \mathfrak{q}$, and furthermore observe that $\mathcal{B} \Vdash \mathcal{C}$ implies there exists some $\mathfrak{r} \in \llbracket\mathcal{C}\rrbracket$ with $\mathfrak{q} \supseteq \mathfrak{r}$. By transitivity it follows that $\mathfrak{p} \supseteq \mathfrak{r}$. With $\mathfrak{p}$ being arbitrary, it follows that $\mathcal{A} \Vdash \mathcal{C}$.
3. Suppose $\mathfrak{p}$ is a path through $\mathcal{A} \wedge \mathcal{B}$ and observe there exist paths $\mathfrak{q}_1 \in \llbracket\mathcal{A}\rrbracket$ and $\mathfrak{q}_2 \in \llbracket\mathfrak{q}\rrbracket$ with $\mathfrak{p} = \mathfrak{q}_1 \cup \mathfrak{q}_2$. As $\mathcal{A} \Vdash \mathcal{C}$ there exists a path $\mathfrak{r} \in \llbracket\mathcal{C}\rrbracket$ such that $\mathfrak{q}_1 \supseteq \mathfrak{r}$. As $\mathfrak{p} = \mathfrak{q}_1 \cup \mathfrak{q}_2$ it follows that $\mathfrak{p} \supseteq \mathfrak{r}$. With $\mathfrak{p}$ being arbitrary, it then follows that $\mathcal{A} \wedge \mathcal{B} \Vdash \mathcal{C}$.
4. Suppose $\mathfrak{p}$ is a path through $\mathcal{A}$, then it follows that $\mathfrak{p} \supseteq \mathfrak{p}_i$ and $\mathfrak{p}_i \in \llbracket\mathcal{A}_i\rrbracket$. Hence, $\mathcal{A} \Vdash \mathcal{A}_i$.
5. Suppose $\mathcal{A} \Vdash \mathcal{B} \wedge \mathcal{C}$ and consider a path $\mathfrak{p} \in \llbracket\mathcal{A}\rrbracket$. It follows that there exists a path $\mathfrak{q} \in \llbracket\mathcal{B} \wedge \mathcal{C}\rrbracket$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. However, as $\llbracket\mathcal{B} \wedge \mathcal{C}\rrbracket$ it follows that $\mathfrak{q} = \mathfrak{q}_1 \cup \mathfrak{q}_2$ for some $\mathfrak{q}_1 \in \llbracket\mathcal{B}\rrbracket$ and $\mathfrak{q}_2 \in \llbracket\mathcal{B}_2\rrbracket$, showing that $\mathfrak{p} \supseteq \mathfrak{q}_1$ where $\mathfrak{q}_1 \in \llbracket\mathfrak{q}_1\rrbracket$. With $\mathfrak{p}$ being arbitrary, it then follows that $\mathcal{A} \Vdash \mathcal{B}$.

6. Suppose $\mathfrak{p}$ is a path through $\mathcal{A}$. It follows from the assumption that $\mathcal{A} \Vdash \mathcal{B}$ then there exists a path $\mathfrak{q} \in \llbracket \mathcal{B} \rrbracket$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. As $\llbracket \mathcal{B} \vee \mathcal{C} \rrbracket = \llbracket \mathcal{B} \rrbracket \cup \llbracket \mathcal{C} \rrbracket$, this implies that there exists a path $\mathfrak{q} \in \llbracket \mathcal{B} \vee \mathcal{C} \rrbracket$ with $\mathfrak{p} \supseteq \mathfrak{q}$. Hence, $\mathcal{A} \Vdash \mathcal{B} \vee \mathcal{C}$.

□

Note that properties (1), (2), and (3) of Theorem 5.2 correspond to the requirements for path-entailment to be a Tarskian consequence relation, modulo the proviso of the tag-disjointness for (3). Regardless, the motivation for studying path-entailment is that we can easily formulate a criterion for attenuation to preserve an individual path-consequence, whereas in the case of logical entailment the situation is not straightforward.

Theorem 5.3. *If $\mathcal{A} \Vdash \mathcal{B}$ and i is a tag not occurring in $\mathcal{B}$ then $\mathcal{A}_i \Vdash \mathcal{B}$.*

Proof. Suppose that $\mathfrak{p} \in \llbracket \mathcal{A}_i \rrbracket$. Then there exists a path $\mathfrak{p}' \in \llbracket \mathcal{A} \rrbracket$ such that $\mathfrak{p} = \mathfrak{p}'_i$, and as $\mathcal{A} \Vdash \mathcal{B}$ there exists a path $\mathfrak{q} \in \llbracket \mathcal{B} \rrbracket$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. However, as i does not occur in $\mathcal{B}$ it follows that $\mathfrak{p} = \mathfrak{p}'_i \supseteq \mathfrak{q}$. With $\mathfrak{p}$ being arbitrary, it follows that $\mathcal{A}_i \Vdash \mathcal{B}$. □

Using Theorem 5.3 it is possible to demonstrate that regular path-contraction operators satisfy a structural analogue of Parikh's relevance postulate. Rather than consider formulae κ logically equivalent to some conjunction $\kappa_1 \wedge \kappa_2$ with $V(\kappa_1) \cap V(\kappa_2) = \varnothing$, we consider matrices $\mathcal{K}$ which are path-equivalent to a conjunction $\mathcal{K}_1 \wedge \mathcal{K}_2$ with $V(\mathcal{K}_1) \cap V(\mathcal{K}_2) = \varnothing$. We refer to these decompositions as *path-splittings*:

Definition 5.2. *If X and Y are disjoint subsets of V with $V = X \cup Y$ then X and Y are **path-independent modulo** $\mathcal{K}$ iff there exist tag-disjoint matrices $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $V(\mathcal{K}_1) \subseteq X$, $V(\mathcal{K}_2) \subseteq Y$, and $\llbracket \mathcal{K} \rrbracket = \llbracket \mathcal{K}_1 \wedge \mathcal{K}_2 \rrbracket$. In this case we say that $(\mathcal{K}_1, \mathcal{K}_2)$ is a **path-splitting** of $\mathcal{K}$.*

Example 5.2. *Although the matrices $p^1 \wedge (\neg p^2 \vee q^3)$ and $p^1 \wedge q^3$ are logically equivalent, and it follows that $\{p\}$ and $\{q\}$ are path-independent modulo $p^1 \wedge q^3$, it follows that $\llbracket p^1 \wedge (\neg p^2 \vee q^3) \rrbracket = \{\{p^1, \neg p^2\}, \{p^1, q^3\}\}$ cannot be expressed as $\llbracket \mathcal{K}_1 \wedge \mathcal{K}_2 \rrbracket = \llbracket \mathcal{K}_1 \rrbracket \otimes \llbracket \mathcal{K}_2 \rrbracket$ for any $\mathcal{K}_1$ and $\mathcal{K}_2$ with $V(\mathcal{K}_1) = \{p\}$ and $V(\mathcal{K}_2) = \{q\}$ showing that $\{p\}$ and $\{q\}$ are not path-independent modulo $p^1 \wedge (\neg p^2 \vee q^3)$.*

Definition 5.3. *A path-contraction operator $-$ satisfies the **path-independence postulate** if and only if given a matrix $\mathcal{K}$ with path-splitting $(\mathcal{K}_1, \mathcal{K}_2)$ alongside a matrix $\mathcal{A}$ with $V(\mathcal{K}_1) \cap V(\mathcal{A}) = \varnothing$, it follows that*

$$\mathcal{K} - \mathcal{A} \Vdash \mathcal{K}_1.$$

In order words, a path-contraction operator satisfies the path-independence postulate when for any matrix $\mathcal{K}$ such that X and Y are path-independent modulo $\mathcal{K}$, after contraction by a belief over the vocabulary Y it follows that the beliefs in the X-component of $\mathcal{K}$ are preserved after contraction.

Theorem 5.4. *Every regular path-contraction operator satisfies the path-independence postulate.*

Proof. Suppose $-$ is a regular path-contraction operator and $\mathcal{K}$ is a matrix with path-splitting $(\mathcal{K}_1, \mathcal{K}_2)$ such that $V(\mathcal{K}_1) \cap V(\mathcal{K}_2) = \varnothing$. Given a matrix $\mathcal{A}$ with $V(\mathcal{K}_1) \cap V(\mathcal{A}) = \varnothing$ it follows that there exists a regular cutting I of $\mathcal{K}$ by $\mathcal{A}$ such that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$. As $(\mathcal{K}_1, \mathcal{K}_2)$ is a path-splitting of $\mathcal{K}$ it follows that $\llbracket \mathcal{K} \rrbracket = \llbracket \mathcal{K}_1 \wedge \mathcal{K}_2 \rrbracket$. By Theorem 5.2 it follows that $\mathcal{K} \Vdash \mathcal{K}_1$. Therefore, as I is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$ and $V(\mathcal{K}_1) \cap V(\mathcal{A}) = \varnothing$, it follows by Theorem 5.2 that $\mathcal{K}_I \Vdash \mathcal{K}_1$. As $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ this means that $\mathcal{K} - \mathcal{A} \Vdash \mathcal{K}_1$. Therefore, the path-independence postulate is satisfied. □

6 Discussion

6.1 Related Work

There are attempts to leverage the connection method and similar techniques for belief contraction already in literature, however we believe our work to be unique in utilising matrix attenuation to preserve the structure of the original formula.

In (Bienvenu, Herzig, and Qi 2008), knowledge bases are converted into prime implicate normal form, resulting in a syntax-independent but nevertheless syntactic belief revision function.

In (Schwind 2010) and (Schwind 2012) belief revision functions are introduced which operate on implicants, which are taken there to be roughly paths through matrices with tags erased. These functions are required to satisfy the AGM postulates, and thus correspond to belief revision rather than belief base revision.

In (Gabbay, Rodrigues, and Russo 2010) a version of the connection method for knowledge bases in conjunctive normal form is used to repair inconsistent knowledge bases. This is accomplished by replacing the knowledge base with the disjunction formed by the conjunctions associated with each maximally consistent subset of those paths through the knowledge base. Their approach can be adapted to belief revision by prioritising which maximally consistent subsets of paths to use. This is accomplished by tagging every variable based on the clause it originated from, placing a priority order on those tags, and then selecting the maximal consistent subpaths which retain the highest priority tags if at all possible. Our approach differs in that we do not require conversion to conjunctive normal form, we focus on belief base contraction rather than revision or repair, and our path-contraction operators comply with the principle of structural preservation.

6.2 Future Work

Clarifying the connection between path-contraction operators and other approaches to belief base contraction via hitting sets and incision functions, as well as attempting to obtain versions of properties such as core-retainment suitable for path-contraction remains an open problem. We believe that investigating "path-remainders" of $\mathcal{K}$ modulo $\mathcal{A}$, defined as those logically strongest $\mathcal{K}'$ such that $\mathcal{K} \Vdash \mathcal{K}'$ yet $\mathcal{K}' \nVdash \mathcal{A}$ will prove illuminating.

Variants of the connection method have been developed for intuitionistic and modal logics (Wallen 1987), as well as for the description logic $\mathcal{ALC}$ (Freitas and Otten 2016).

We believe that the theory of path-contraction operators introduced here will generalise well to these formalisms. It would also be interesting to investigate whether this approach also extends to tableaux methods for non-monotonic logics (Olivetti 1999) such as sceptical default reasoning (Bonatti and Olivetti 1997b) or circumscription (Bonatti and Olivetti 1997a).

We are also interested in conducting an empirical study of the performance of our Path-Contraction Algorithm.

7 Conclusion

In this paper we introduced the class of path-contraction operators, which utilise the process of matrix attenuation to carry out a form of belief base contraction in a manner which leaves the syntactic structure of the original belief base minimally changed. We have presented an algorithm for implementing a path-contraction operator and shown it to have complexity **DP**, and better yet **FNP** under reasonable restrictions. We have further introduced the notion of path-entailment, and shown that regular path-contraction operators satisfy an analogue of Parikh's relevance postulate, which further substantiates the claim that path-contraction operators are carrying out only minimal changes to the original formula. Finally, we discussed where our approach fits in with other related approaches to belief change.

Acknowledgements

We would like to thank our reviewers for their insightful comments, as well as the Natural Sciences and Engineering Research Council of Canada for financial support.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 510–530.

Andrews, P. B. 1976. Refutations by matings. *IEEE Trans. Computers* 25(8):801–807.

Bibel, W. 1981. On matrices with connections. *Journal of the ACM (JACM)* 28(4):633–645.

Bienvenu, M.; Herzig, A.; and Qi, G. 2008. Prime implicate-based belief revision operators. In *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, 741–742. IOS Press.

Bonatti, P. A., and Olivetti, N. 1997a. A sequent calculus for circumscription. In Nielsen, M., and Thomas, W., eds., *International Workshop on Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*, 98–114. Aarhus, Denmark: Springer.

Bonatti, P. A., and Olivetti, N. 1997b. A sequent calculus for skeptical default logic. In Galmiche, D., ed., *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, volume 1227 of *Lecture Notes in Computer Science*, 107–121. Springer.

Caridroit, T.; Konieczny, S.; and Marquis, P. 2017. Contraction in propositional logic. *International Journal of Approximate Reasoning* 80:428–442.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial intelligence* 57(2-3):227–270.

Freitas, F., and Otten, J. 2016. A connection calculus for the description logic $\mathcal{ALC}$. In Khoury, R., and Drummond, C., eds., *Canadian Conference on Artificial Intelligence*, volume 9673 of *Lecture Notes in Computer Science*, 243–256. Springer.

Gabbay, D. M.; Rodrigues, O. T.; and Russo, A. 2010. *Revision, Acceptability and Context: Theoretical and Algorithmic Aspects*. Springer Science & Business Media.

Hansson, S. O. 1991. Belief contraction without recovery. *Studia Logica* 50(2):251–260.

Hansson, S. O. 1994. Kernel contraction. *Journal of Symbolic Logic* 845–859.

Hansson, S. O. 1999. *A textbook of belief dynamics: Theory change and database updating*. Kluwer Academic Publishers.

Hunter, A., and Agapeyev, J. 2019. An efficient solver for parametrized difference revision. In *Australasian Joint Conference on Artificial Intelligence*, 143–152. Springer.

Katsuno, H., and Mendelzon, A. O. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3):263–294.

Kreitz, C., and Otten, J. 1999. Connection-based theorem proving in classical and non-classical logics. *Journal of Universal Computer Science* 5(3):88–112.

Levesque, H. J. 1984. A logic of implicit and explicit belief. In Brachman, R. J., ed., *Proceedings of the National Conference on Artificial Intelligence*, 198–202. Austin, TX, USA: AAAI Press.

Olivetti, N. 1999. Tableaux for nonmonotonic logics. In *Handbook of Tableau Methods*. Springer. 469–528.

Otten, J. 2011. A non-clausal connection calculus. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 226–241. Springer.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, Language and Computation* 2(96):266–268.

Peppas, P. 2008. Belief revision. *Foundations of Artificial Intelligence* 3:317–359.

Schwind, C. 2010. From inconsistency to consistency: Knowledge base revision by tableaux opening. In *Ibero-American Conference on Artificial Intelligence*, 120–132. Springer.

Schwind, C. 2012. Belief base revision as a binary operation on implicant sets: a finitary approach. In *Proceedings of the 14th International Workshop on Non-Monotonic Reasoning*.

Wallen, L. A. 1987. *Automated proof search in non-classical logics: efficient matrix proof methods for modal and intuitionistic logics*. Ph.D. Dissertation, The University of Edinburgh.

Merging Conditional Beliefs: Approaches Based on Syntax and Semantic

Meliha Sezgin[1] , **Gabriele Kern-Isberner**[1]
[1]Department of Computer Science, TU Dortmund University, Germany
meliha.sezgin@tu-dortmund.de, gabriele.kern-isberner@cs.uni-dortmund.de

Abstract

We extend the challenge of merging possibly conflicting conditional belief bases to consistent beliefs of agents. Since conditional beliefs are logically more intertwined, the merging process offers a lot of challenges and also leaves room for different interpretations of reasonable merging processes. In this paper, we present one conditional-based merging operator and two conditional merging operators that take the agents' epistemic states into account. Moreover, we define quality criteria for general conditional merging operators that can be seen as a guideline for reasonable merging processes and discuss two well-known subclasses of merging operators for conditional merging.

1 Introduction

Belief merging aims at combining several pieces of information when there are no strict precedence between them, for example if an agents learns different rules or mechanisms to handle a problem from different sources of information, which are equally reliable. In order to make the information from different contexts usable an agent has to perform a merging process, and it seems quite natural that this information does not solely consist of propositional beliefs, but rather of conditional beliefs. A variety of merging operators, dealing with conflicting information have been studied in the literature, e.g. in (Baral, Kraus, and Minker 1991; Revesz 1997; Liberatore and Schaerf 1998). Among the most influential works in this field, Konieczny and Pino Pérez's logical characterization of propositional merging operators in (Konieczny and Pérez 1998; Konieczny and Pérez 2002) can be found. Yet, the information provided by conditional beliefs is quite different from propositional ones. Conditionals are three-valued logical entities and they display rather an agent's preferences and reasoning patterns than static knowledge. So, we cannot simply transfer the properties and operators for propositional merging to the framework of conditional beliefs. The following example illustrates conditional belief merging as planning with uncertainty:

Example 1. *Suppose we want to speculate on the stock market and ask three financial experts about their instruction for action in the event of a stock bubble. The first one states, that bubbles (B) lead to crashes (R) in the stock market. If we recognize a bubble, we should sell (S) our shares. Yet, in general, in the event of a crash, we should hold our shares. The second one shares the opinion that bubbles lead to crashes, but, recommends to hold the shares in case the bubble bursts. The third states that it is best to sell shares if we recognize a bubble, but hold shares when the bubble has already burst. These experts are equally reliable, so what should we do now with our shares?*

$$\begin{tikzcd}
 & C = \{\Delta_i\}_{i=1}^{m} \arrow[dl] \arrow[drr] & & & \\
\mathcal{M}_C \arrow[dd, "\mathrm{S}"'] & & & E = \{\kappa_{\Delta_i}\}_{i=1}^{m} \arrow[dl] \arrow[dr] & \\
 & & K(E) \arrow[d] & & K^c(E) \arrow[d] \\
\Gamma_S^{pm}(C)C & & \Gamma_K^{co}(C) & & \Gamma_{K^c}^{coc}(C)
\end{tikzcd}$$

Figure 1: Schematic representation of the three merging operators in this paper.

We propose in this paper a conditional-based approach to conditional merging, that selects a consistent set of conditionals from, possibly conflicting, sets of conditionals. However, this approach does not take into account the role of conditional beliefs as representations of an agent's preferences and reasoning patterns, i.e., the close relationship between conditional beliefs and epistemic states. So, we also propose two more fine-grained conditional merging operators that are based on the epistemic state of an agent, representing semantic approaches of conditional belief merging. We call these approaches to conditional merging, epistemic conditional merging. A set of quality criteria defined in this paper enables us to compare conditional merging operators, also we distinguish between two common subclasses of them. In Figure 1, we give a schematic representation of the three merging operators presented in this paper. The rest of this paper is organized as follows: We start with some formal preliminaries in Section 2. Section 3 introduces the logical framework for conditional merging operators. In Section 4 we present a conditional-based merging operator and then in Section 5 epistemic conditional merging is presented and two ways of epistemic conditional merging are presented in the following subsections. We conclude and discuss the results of this paper in Section 6.

2 Formal Preliminaries

We start by recalling basics of propositional and conditional logic together with epistemic states represented as ranking functions.

2.1 Propositional Logic

Let $\mathcal{L}$ be a finitely generated propositional language over an alphabet Σ with atoms $a, b, c, \ldots$ and with formulas $A, B, C, \ldots$. For conciseness of notation, we will omit the logical *and*-connector, writing AB instead of $A \wedge B$, and overlining formulas will indicate negation, i.e. $\overline{A}$ means $\neg A$. The set of all propositional interpretations over Σ is denoted by Ω_Σ. As the signature will be fixed throughout the paper, we will usually omit the subscript and simply write Ω. $\omega \models A$ means that the propositional formula $A \in \mathcal{L}$ holds in the possible world $\omega \in \Omega$; then ω is called a *model* of A, and the set of all models of A is denoted by $Mod(A)$. For propositions $A, B \in \mathcal{L}$, $A \models B$ holds iff $Mod(A) \subseteq Mod(B)$, as usual. By slight abuse of notation, we will use ω both for the model and the corresponding conjunction of all positive or negated atoms. This will allow us to ease notation a lot. Since $\omega \models A$ means the same for both readings of ω, no confusion will arise.

2.2 Conditionals and Ranking Functions

We extend $\mathcal{L}$ to a conditional language $(\mathcal{L} \,|\, \mathcal{L})$ by introducing a conditional operator $|$: $(\mathcal{L} \,|\, \mathcal{L}) = \{(B \,|\, A) | A, B \in \mathcal{L}\}$. $(\mathcal{L}|\mathcal{L})$ is a flat conditional language, no nesting of conditionals is allowed. A is called the antecedent or the premise of $(B \,|\, A)$, and B is the consequence. $(B \,|\, A)$ expresses *"If A then plausibly B"*. According to de Finetti (1975), conditionals can be regarded as three-valued logical entities on possible worlds $\omega \in \Omega$, distinguishing between verification $\omega \models AB$, falsification $\omega \models A\overline{B}$ and neutrality $\omega \models \overline{A}$. Two conditionals $(B \,|\, A)$ and $(B' \,|\, A')$ are *equivalent*, denoted by $(B \,|\, A) \equiv (B' \,|\, A')$, if they have the same verification and the same falsification behavior, i.e. $AB \equiv A'B'$ and $A\overline{B} \equiv A'\overline{B'}$. We presuppose in this paper that all conditionals are not self-contradicting, i.e., it holds that $AB \not\equiv \bot$ for each conditional $(B \,|\, A)$. A *conditional belief base* is a finite set of conditionals $\Delta = \{(B_1 \,|\, A_1), \ldots, (B_n \,|\, A_n)\}$. We denote the set of all verifying models for a conditional belief base $Mod(\{AB \,|\, (B \,|\, A) \in \Delta\})$ as Ω^v_Δ. In the same manner we denote by Ω^f_Δ the set of all worlds falsifying the conditionals in Δ, i.e. $\Omega^f_\Delta = Mod(\{A\overline{B} \,|\, (B \,|\, A) \in \Delta\})$.

To give an appropriate semantics to conditional belief bases, we need richer semantic structures like epistemic states in the sense of Halpern (2005). In this paper, we build upon ordinal conditional functions (Spohn 1988, 2014), which are a representation of epistemic states. *Ordinal conditional functions* (OCFs, also called *ranking functions*) $\kappa : \Omega \to \mathbb{N} \cup \{\infty\}$, with $\kappa^{-1}(0) \neq \emptyset$, assign to each world ω an implausibility rank $\kappa(\omega)$. The higher $\kappa(\omega)$, the less plausible ω is, and the normalization constraint requires that there are worlds having maximal plausibility. We have $\kappa(A) := \min\{\kappa(\omega) \mid \omega \models A\}$, and in particular, $\kappa(\bot) = \infty$. Due to $\kappa^{-1}(0) \neq \emptyset$, at least one of $\kappa(A)$ and $\kappa(\overline{A})$ must be 0. A proposition A is believed if $\kappa(\overline{A}) > 0$. A conditional $(B \,|\, A)$ is accepted by κ, denoted by $\kappa \models (B \,|\, A)$, if $\kappa(AB) < \kappa(A\overline{B})$. κ accepts a conditional belief base Δ, $\kappa \models \Delta$, iff $\kappa \models (B \,|\, A)$ for each $(B \,|\, A) \in \Delta$, then κ is called *admissible with respect to* Δ. Vice versa, a conditional belief base Δ is *consistent*, iff there exists an OCF κ s.t. $\kappa \models \Delta$.

A *conditional profile* is a nonempty multi-set of conditional bases $C = \{\Delta_j\}_{j=1,\ldots,m}$ with $\Delta_j \neq \emptyset$ (hence different agents are allowed to exhibit identical bases), and represents a group of m agents. We denote by $\bigcup C$ the union of bases of C, i.e. $\bigcup C = \Delta_1 \cup \ldots \cup \Delta_m$ and by $\bigcap C$ the intersection of bases of C, i.e. $\bigcap C = \Delta_1 \cap \ldots \cap \Delta_m$. A profile C is said to be *consistent* if and only if $\bigcup C$ is consistent.

Remark 1. *Since the world views of agents shall be consistent, we will presuppose in the rest of the paper that all belief bases in a profile are consistent.*

The multiset union is denoted by $\sqcup$. By abuse of notation, we will write $\Delta \sqcup C$ instead of $\{\Delta\} \sqcup C$. We denote by Δ^n the profile in which Δ appears n times, more precisely $\Delta^n = \underbrace{\Delta \sqcup \ldots \sqcup \Delta}_{n}$. Two sets of conditionals Δ, Δ' are *elementwise equivalent* ($\Delta \equiv_e \Delta'$) iff for every conditional in each set there is an equivalent conditional in the other set.

Definition 1. *Let C, C' be conditional belief profiles. C and C' are* equivalent*, noted $C \equiv_c C'$, iff there exists a bijection f from $C = \{\Delta_j\}_{j=1,\ldots,m}$ to $C' = \{\Delta'_{j'}\}_{j'=1,\ldots,m'}$ s.t. for any $\Delta_j \in C$ there exists a $\Delta'_{j'} \in C'$, s.t. $f(\Delta_j) \equiv_e \Delta'_{j'}$.*

Note that the relation $\equiv_c$ is an equivalence relation on belief profiles. As for conditional belief bases, we define the set of verifying resp. falsifying worlds for a conditional profile as $\Omega^v_{\bigcup C}$ resp. $\Omega^f_{\bigcup C}$. To ease notation, we write Ω^v_C instead of $\Omega^v_{\bigcup C}$ and Ω^f_C instead of $\Omega^f_{\bigcup C}$.

3 Conditional Merging Operators

In this section, we give a logical definition of conditional merging operators and provide a set of postulates that define good behavior concerning the merging. Furthermore, we introduce two subclasses of conditional merging operators.

A *conditional merging operator* defines a mapping between a conditional belief profile C and a consistent set of conditionals:

Definition 2. *Let $C = \{\Delta_1, \ldots, \Delta_m\}$ be a conditional profile. A* conditional merging operator *$\Gamma : C \mapsto \Gamma(C)$ assigns to each conditional belief profile a consistent set of conditionals $\Gamma(C)$ with $\Gamma(C) \subseteq \bigcup C$.*

Note that, in our definition of conditional merging, we take only conditionals from $\bigcup C$ into account. This puts our operators more in the vicinity of selection operators as defined by Liberatore and Schaerf for propositions (Liberatore and Schaerf 1998). Now that we have general conditional merging operators $\Gamma(C)$ for a conditional profile C, we provide a set of desirable quality criteria in order for a conditional merging operator to behave rationally in the process of merging.

(CM*) $\Gamma(C) \subseteq \bigcup C$
(CM0) $\Gamma(C)$ is consistent
(CM1) If $\bigcup C$ is consistent, then $\Gamma(C) = \bigcup C$
(CM2) If $C_1 \equiv_c C_2$, then $\Gamma(C_1) \equiv_c \Gamma(C_2)$
(CM3) $\bigcap C \subseteq \Gamma(C)$

The first postulate (CM*), is a special one. It states that for the merging process we only take conditional information that is already given in one of the conditional bases $\Delta_i \in C$ into consideration. This postulate is inherent in our definition of conditional merging operators. Since more general conditional merging operators that also include conditionals outside of $\bigcup C$ are also conceivable we include this postulate in our listing of postulates. The other postulates define basic properties a conditional merging operator should satisfy: (CM0) requires the belief base obtained to be satisfiable, this is actually a restatement of the definition of conditional merging operators. (CM1) assures that if it is possible to retain all the information contained in a conditional belief profile, then we should do so. Since we consider only consistent conditional belief bases (CM1) implies for singleton profiles $C = \{\Delta\}$ that $\Gamma(C) = \Gamma(\Delta) = \Delta$. (CM2) demands irrelevance of syntax and, since we demand elementwise-equivalence, commits us to commutativity, i.e. the result of the merging operation is independent of any order of elements in the belief profile. (CM3) states if a conditional is included in every base Δ_i in C, this information should also be present in the merged belief base $\Gamma(C)$.

Now, we turn to two (common) subclasses of merging operations, *majority and arbitration merging operators*. So called *majority conditional merging operators* strive to satisfy a maximum of protagonists, i.e. if a conditional belief set has a large audience, then it will be included in the opinion of the group.

(Maj) For all Δ there exists $n \in \mathbb{N}$, s.t. $\Delta \subseteq \Gamma(C \sqcup \Delta^n)$

In contrast to majority merging operators, *arbitration conditional merging operators* implement the idea of independence from the cardinality of opinions in the merging process. Conditional operators from this subclass prefer median possible choices, i.e., they strive to satisfy each protagonist to the best possible degree in the merging process and therefore minimize individual dissatisfaction, whereas majority merging operators minimize global dissatisfaction.

(Arb) For C, Δ, it holds that $\Gamma(C \sqcup \Delta^n) = \Gamma(C \sqcup \Delta)$ for all $n \in \mathbb{N}$

Note that, this is a more strict version of arbitration, which resembles the majority independence postulate from (Konieczny and Pérez 2002) for integrity constraints merging operators. We prefer this version, which is closer to the intuition of arbitration. The following theorem proves the incompatibility of merging behavior in the sense of majority and arbitration conditional merging:

Theorem 1. *A conditional merging operator Γ cannot satisfy both (Maj) and (Arb).*

Proof. Let Δ_1, Δ_2 be conditional belief bases, s.t. $\Delta_1 \cup \Delta_2$ is inconsistent. From (Arb), if follows that $\Gamma(\Delta_1 \sqcup \Delta_2) = \Gamma(\Delta_1 \sqcup \Delta_2^n) = \Gamma(\Delta_1^n \sqcup \Delta_2)$ and with (Maj), we can follow that $\Delta_1 \subseteq \Gamma(\Delta_1 \sqcup \Delta_2)$ and $\Delta_2 \subseteq \Gamma(\Delta_1 \sqcup \Delta_2)$, thus, $\Gamma(\Delta_1 \sqcup \Delta_2)$ is inconsistent, which contradicts that Γ is a conditional merging operator. □

4 Conditional-based Merging Operators

Conditional-based merging operators for conditional profiles are sensitive to the syntax of conditionals in the conditional belief bases to be merged. They are depicted on the left-hand side of Figure 1. A straightforward approach for conditional-based merging is adapted from partial-meet contraction (Alchourrón, Gärdenfors, and Makinson 1985). Via the intersection of selected maximal consistent sets of conditionals (for set inclusion) from $\bigcup C$ we determine a consistent set of conditionals in C.

Definition 3. *Let C be a conditional profile. A set M is called a* maximal consistent set of *C, if it holds that:*

- $\bigcap C \subseteq M \subseteq \bigcup C$
- *M is consistent*
- *For every set M' with $M \subseteq M' \subseteq \bigcup C$, $M \neq M'$, it holds that M' is inconsistent.*

The set of all maximal consistent sets of C is denoted by $\mathcal{M}_C$.

A *selection function* $S : \{\Lambda_1 \dots, \Lambda_k\} \mapsto \{\Lambda'_1, \dots, \Lambda'_{k'}\} \subseteq \{\Lambda_1, \dots, \Lambda_k\}$ is a mapping between sets of consistent conditional belief bases Λ_i to a set of consistent conditional belief bases $\Lambda'_{i'}$. In order to realize our conditional merging operator we take $\mathcal{M}_C$ as input for S, s.t.

$$S(\mathcal{M}_C) \subseteq \mathcal{M}_C$$

with $S(\mathcal{M}_C) \neq \emptyset$ if $\mathcal{M}_C \neq \emptyset$. Using this selection function we are able to define a partial meet conditional merging operator:

Definition 4. *Let $C = \{\Delta_1, \dots, \Delta_m\}$ be a conditional profile and $\mathcal{M}_C$ be the set of all maximal consistent sets of C. For a selection function S, we define a* partial meet conditional merging operator *as*

$$\Gamma_S^{pm}(C) = \bigcap S(\mathcal{M}_C).$$

The following example illustrates the behavior of the partial meet merging operator:

Example 2 (Continue Example 1)**.** *The three experts' advices can be represented by the following conditional belief bases: $\Delta_1 = \{(R \,|\, B), (\overline{S} \,|\, R), (S \,|\, B)\}$, $\Delta_2 = \{(R \,|\, B), (\overline{S} \,|\, BR)\}$ and $\Delta_3 = \{(S \,|\, B), (\overline{S} \,|\, BR)\}$. We have $C = \{\Delta_1, \Delta_2, \Delta_3\}$ and $\bigcup C = \{(R \,|\, B), (\overline{S} \,|\, R), (S \,|\, B), (\overline{S} \,|\, BR)\}$ is inconsistent. There are three maximal consistent subsets of C, $\mathcal{M}_C = \{M_1, M_2, M_3\}$ with*

$$M_1 = \{(\overline{S} \,|\, R), (S \,|\, B), (\overline{S} \,|\, BR)\}$$
$$M_2 = \{(\overline{S} \,|\, R), (R \,|\, B), (\overline{S} \,|\, BR)\}$$
$$M_3 = \{(\overline{S} \,|\, R), (R \,|\, B), (S \,|\, B)\}.$$

Note that, C is not consistent, since there is no world $\omega \in \Omega$ that verifies one of the conditionals in $\{(R \,|\, B), (S \,|\, B), (\overline{S} \,|\, BR)\} \subseteq \bigcup C$ and does not falsify *any of the others. There are several options for selection functions, e.g. a cautious one, that takes all agents possible maximal sets into account $S_1 = \{M_1, M_2, M_3\}$ therefore implementing a full-meet strategy. Or a maxichoice strategy, i.e. a selection function that takes only the opinion of one agent, e.g., Δ_1, into account, i.e. $S_2 = \{M_1\}$. It is also possible that the decision maker in the merging process explicitly wants to exclude agent 2, i.e. $S_3 = \{M_1, M_3\}$. We get the following corresponding merging results:*

$$\Gamma^{pm}_{S_1}(C) = \{(\bar{S} \,|\, R)\}$$
$$\Gamma^{pm}_{S_2}(C) = \{(\bar{S} \,|\, R), (S \,|\, B), (\bar{S} \,|\, BR)\}$$
$$\Gamma^{pm}_{S_3}(C) = \{(\bar{S} \,|\, R), (S \,|\, B)\}.$$

The following theorem investigates the behavior of $\Gamma^{pm}_S(C)$ in the light of the postulates from Section 3:

Theorem 2. *Let $C = \{\Delta_i\}_{i=1}^m$ be a conditional profile. The partial meet conditional merging $\Gamma^{pm}_S(C) = \bigcap S(\mathcal{M}_C)$ defined via a selection function S satisfies (CM*), (CM0), (CM1), (CM2), (CM3).*

Proof. (CM*) follows straightforward from the first property for maximal consistent sets in Definition 3. $\Gamma^{pm}_S(C)$ is defined via the intersection of maximally consistent sets of conditionals, i.e., there exists a set $M \in \mathcal{M}_C$, s.t. $\Gamma^{pm}_S(C) = \bigcap S(\mathcal{M}_C) \subseteq M$ and therefore $\Gamma^{pm}_S(C)$ is consistent and Γ^{pm}_S always satisfies (CM0). Since it holds that $\mathcal{M}_C = \bigcup C \neq \emptyset$, if $\bigcup C$ is consistent, (CM1) holds for $\Gamma^{pm}_S(C)$ for any selection functions $S(\mathcal{M}_C)$. (CM2), (CM3) follow immediately from Definition 3 since we define sets, i.e. we are bound to commutativity and irrelevance of syntax. From first property of maximal consistent set of C it follows that $\bigcap C \subseteq M \in \mathcal{M}_C$, i.e. for any S it holds that $\bigcap C \subseteq \bigcap S(\mathcal{M}_C) = \Gamma^{pm}_S(C)$. □

Note that the result of Γ^{pm}_S is maximally consistent, only for a selection function that satisfies a maxichoice strategy, i.e., picks just one of the maximally consistent sets from $\mathcal{M}_C$. The next theorem classifies Γ^{pm}_S as an arbitration conditional merging operator:

Theorem 3. *Every partial meet conditional merging operator Γ^{pm}_S satisfies (Arb).*

Proof. Let $n \in \mathbb{N}$ and $C = \{\Delta_1, \ldots, \Delta_m\}$ be a conditional profile and Δ be a conditional belief base. It holds that $\mathcal{M}_{C \sqcup \Delta} = \mathcal{M}_{C \sqcup \Delta^n}$ and therefore $\Gamma^{pm}_C(C \sqcup \Delta) = \bigcap S(\mathcal{M}_{C \sqcup \Delta}) = \bigcap S(\mathcal{M}_{C \sqcup \Delta^n}) = \Gamma^{pm}_S(C \sqcup \Delta^n)$, i.e., Γ^{pm}_S satisfies (Arb). □

As we have seen, merging via partial-meet merging satisfies desirable properties for conditional merging operators. However, conditional bases make only part of the agents' epistemic states explicit. Taking the full epistemic states into account might provide more information that proves helpful for merging.

5 Epistemic Conditional Merging Operators

In this Section, we discuss two approaches to merging that are depicted on the right-hand side of Figure 1. Therefore, we introduce a mapping between conditional belief profiles and ranking functions in Subsection 5.1.

5.1 Coalescent Assignments

Conditionals and epistemic states are strongly connected. In this paper epistemic states are represented via ranking functions. On the one hand, it holds that for each consistent set of conditionals Δ there exists a ranking function κ that models Δ, i.e. $\kappa \models \Delta$, therefore ranking functions can be seen as an implementation of conditional beliefs. On the other hand, every ranking function κ defines a set of conditionals via the following extraction:

$$\Delta_\kappa = \{(B \,|\, A) \in (\mathcal{L} \,|\, \mathcal{L}) \,|\, \kappa(AB) < \kappa(A\overline{B})\}. \qquad (1)$$

This bidirectional (but not bijective) translation between ranking functions and conditional belief bases, enables us to define conditional merging operators for conditional profiles $C = \{\Delta_1, \ldots, \Delta_m\}$ via the combination of sets of admissible ranking functions $E = \{\kappa_1, \ldots, \kappa_m\}$ with Δ_i-admissible ranking functions κ_{Δ_i}, since (1) defines an extraction of sets of conditionals from epistemic states.

Using the extraction of conditional beliefs from ranking functions, we define a semantic characterization of conditional merging operators in general, which we call *coalescent assignment* and transfer the quality criteria defined in Section 3 to the framework of ranking functions.

Definition 5. *Let C be conditional belief profile. A* coalescent assignment *is a mapping $C \mapsto \kappa_C$ that assigns to each conditional belief profile C a ranking function κ_C.*

This definition of coalescent assignments corresponds to the definition of conditional merging operators in Definition 2. Note that our coalescent assignments are inspired by *syncretic assignments* for integrity constraint merging operators defined by Konieczny and Pino Pérez in (2002). We propose the following set of postulates for coalescent assignments:

(CA1) If $\bigcup C$ is consistent, then for each $(B \,|\, A) \in \bigcup C$ it holds that $\kappa_C(AB) < \kappa_C(A\overline{B})$

(CA2) If $C_1 \equiv_c C_2$, then $\kappa_{C_1} = \kappa_{C_2}$

(CA3) If $(B \,|\, A) \in \bigcap C$ then $\kappa_C(AB) < \kappa_C(A\overline{B})$

The following postulates implement different strategies for determining κ_C, one in the style of majority merging and one in the style of arbitration:

(Maj$^\kappa$) There exists $n \in \mathbb{N}$, s.t. for all $(B \,|\, A) \in \Delta$, it holds that $\kappa_{C \sqcup \Delta^n}(AB) < \kappa_{C \sqcup \Delta^n}(A\overline{B})$

(Arb$^\kappa$) For all $n \in \mathbb{N}$, $\kappa_{C \sqcup \Delta}(AB) < \kappa_{C \sqcup \Delta}(A\overline{B})$ iff $\kappa_{C \sqcup \Delta^n}(AB) < \kappa_{C \sqcup \Delta^n}(A\overline{B})$ for $(B \,|\, A) \in C \sqcup \Delta$

The next theorem connects the postulates for conditional merging operators and for coalescent assignments via the lifting from ranking functions to sets of conditionals defined in (1).

Theorem 4. *Let C be a conditional profile and Δ be a conditional belief base. Let κ_C be the result of a coalescent assignment. For*

$$\Gamma(C) = \{(B \,|\, A) \in \bigcup C \mid \kappa_C(AB) < \kappa_C(A\overline{B})\} \qquad (2)$$

it holds that:

$\omega \in \Omega$	κ_{Δ_1}	κ_{Δ_2}	κ_{Δ_3}	K_{Min}	K_{Max}	K_{Σ}
BRS	0	2	3	0	3	5
$BR\overline{S}$	2	1	2	1	2	5
$B\overline{R}S$	2	2	1	1	2	5
$B\overline{R}\overline{S}$	1	2	2	1	2	5
$\overline{B}RS$	1	1	0	0	1	2
$\overline{B}R\overline{S}$	0	1	1	0	1	2
$\overline{B}\overline{R}S$	0	1	0	0	1	1
$\overline{B}\overline{R}\overline{S}$	0	0	0	0	0	0

Table 1: The table shows three admissible ranking functions for Δ_1, Δ_2 and Δ_3 from Example 3 and the result of the combination operation of $E = \{\kappa_{\Delta_1}, \kappa_{\Delta_2}, \kappa_{\Delta_3}\}$ via the operators K_{Min}, K_{Max} and K_{Σ}.

1. *If κ_C satisfies (CA1) then $\Gamma(C)$ satisfies (CM1)*
2. *If κ_C satisfies (CA2) then $\Gamma(C)$ satisfies (CM2)*
3. *If κ_C satisfies (CA3) then $\Gamma(C)$ satisfies (CM3)*
4. *If $\kappa_{C \sqcup \Delta^n}$ satisfies (Maj$^\kappa$) then Γ satisfies (Maj)*
5. *If $\kappa_{C \sqcup \Delta}$ and $\kappa_{C \sqcup \Delta^n}$ satisfy (Arb$^\kappa$) then Γ satisfies (Arb)*

Proof. $\Gamma(C)$ is a conditional merging operator, i.e. the set defined in (2) is always consistent, because κ_C is a coalescent assignment and therefore a ranking function that models $\Gamma(C)$. The statements 1.-3. follow in a straightforward way. Let $n \in \mathbb{N}$, s.t. $\kappa_{C \sqcup \Delta}(AB) < \kappa_{C \sqcup \Delta}(A\overline{B})$ for all $(B \mid A) \in \Delta$. Note that the existence of such an n is guaranteed since $\kappa_{C \sqcup \Delta}$ satisfies (Maj$^\kappa$). Because $(B \mid A) \in \bigcup(C \sqcup \Delta)$ it follows from (2) that $(B \mid A) \in \Gamma(C \sqcup \Delta^n)$, i.e., $\Delta \subseteq \Gamma(C \sqcup \Delta^n)$ and Γ satisfies (Maj). For (Arb), we assume that $(B \mid A) \in \Gamma(C \sqcup \Delta)$. From (2), it follows that $\kappa_{C \sqcup \Delta}(AB) < \kappa_{C \sqcup \Delta}(A\overline{B})$ and therefore $\kappa_{C \sqcup \Delta^n}(AB) < \kappa_{C \sqcup \Delta^n}(A\overline{B})$ due to (Arb$^\kappa$). Thus $(B \mid A) \in \Gamma(C \sqcup \Delta^n)$. The other inclusion follows the same argumentation vice versa and thus, Γ satisfies (Arb). □

Note that, due to the lifting in (2), we are able to define epistemic conditional merging operators via coalescent assignments. Following this strategy, we will define epistemic conditional merging operators in the following subsections via the combination of epistemic states that implement the conditional beliefs of the agents in the merging process. The concatenation of defining sets of ranking functions from the conditional profile and the combination of these representations of conditional beliefs will define a coalescent assignment, i.e. a fruitful approach to epistemic conditional merging.

5.2 Epistemic Conditional Merging via Combination

In this section, we define *combination operators for ranking functions* that map sets of ranking functions $\{\kappa_1, \dots, \kappa_m\}$ to a combined ranking function $K(\{\kappa_1, \dots, \kappa_m\})$. We call a finite set of epistemic states $E = \{\kappa_1, \dots, \kappa_m\}$ a *ranking profile*. We assume in this and the following subsection that the mapping between conditional profiles C and the corresponding ranking profile E is given in a straightforward way by the belief states of the agents participating in the merging process.

Definition 6. *A* combination operator K for ranking functions *is a function from the set of all non-empty ranking profiles E to the set of all ranking functions, i.e. $K : E \mapsto K(E)$ and $K(E)$ is a ranking function satisfying the following basic properties:*

(K0) $K(\{\kappa\})(\omega) = \kappa(\omega)$

(K1) *If $\kappa_i(\omega) \leqslant \kappa_i(\omega')$ for all $i \in \{1, \dots, |E|\}$ then $K(E)(\omega) \leqslant K(E)(\omega')$*

(K2) *If $K(E)(\omega) \leqslant K(E)(\omega')$ then $\kappa_i(\omega) \leqslant \kappa_i(\omega')$ for some $i \in \{1, \dots, |E|\}$*

(K0) requires that trivial combinations with singleton ranking profiles lead to no changes. (K1) states that if all ranking functions in E agree that ω is at least as plausible as ω', then this should also hold for the resulting ranking function. If ω is at least as plausible as ω' after combination, then (K2) expects justification for this via a ranking function κ in which ω is at least as plausible than ω'. Note that (K2) is a restatement of the Pareto's principle in its contrapositive form, which is one of the properties used to establish Arrow's impossibility theorem in social choice theory (Arrow 1963). Note that, the definition of combination operators for ranking functions and the corresponding postulates are inspired by the definition of combination operators for lists of epistemic states from (Meyer 2001). Meyer introduces more postulates that guide the combination operation on more general epistemic states, which become obsolete in the framework of ranking functions. Furthermore, combination operators by Meyer share some similarities with propositional merging operators á la Konieczny and Pino Pérez, but also take on a different perspective due to the usage of full epistemic states which is useful for our purposes. Now, we introduce three combination operators for ranking functions, which are inspired by common combination operations in the framework of Meyer:

Definition 7. *Let $E = \{\kappa_1, \dots, \kappa_m\}$ be a ranking profile and $\omega \in \Omega$:*

- $K_{Min}(E)(\omega) = K_0 + \min_{i=1,\dots,m}\{\kappa_i(\omega)\}$ *with* $K_0 = -\min_{\omega \in \Omega}\{\min_{i=1,\dots,m}\{\kappa_i(\omega)\}\}$
- $K_{Max}(E)(\omega) = K_0 + \max\limits_{i=1,\dots,m}\{\kappa_i(\omega)\}$ *with* $K_0 = -\min_{\omega \in \Omega}\{\max_{i=1,\dots,m}\{\kappa_i(\omega)\}\}$
- $K_{\Sigma}(E) = K_0 + \sum_{i=1}^{m} \kappa_i(\omega)$ *with* $K_0 = -\min_{\omega \in \Omega}\{\sum_{i=1}^{m} \kappa_i(\omega)\}$

K_0 is a normalization constant, s.t. the result of each combination operator is again a ranking function.

The following proposition shows that the operators from Definition 7 are in fact combination operators for ranking functions, i.e. they satisfy (K0) to (K2):

Proposition 1. *Let $E = \{\kappa_i\}_{i=1}^m$ be a ranking profile. $K_{Min}(E), K_{Max}(E)$ and $K_{\Sigma}(E)$ are combination operators on ranking functions.*

Proof. Due to K_0, it follows that $K_{Min}(E), K_{Max}(E)$ and $K_{\Sigma}(E)$ are ranking functions. (K0) follows immediately for $E = \{\kappa\}$. Let $\omega, \omega' \in \Omega$ with $\kappa_i(\omega) \leqslant \kappa_i(\omega')$ for all $i \in$

$\{1, \dots, m\}$, then $K_{Min}(E)(\omega) \leqslant K_{Min}(E)(\omega')$, the same holds for $K_{Max}(E)$ and $K_{\Sigma}(E)$, i.e. (K1) holds. Now, we assume that $K_{Min}(E)(\omega) \leqslant K_{Min}(E)(\omega')$, it follows immediately that there exists some $i \in \{1, \dots, m\}$ s.t. $\kappa_i(\omega) \leqslant \kappa_i(\omega')$. The same holds for K_{Max}. For K_{Σ}, we assume that there exists no κ_i, s.t. $\kappa_i(\omega) \leqslant \kappa_i(\omega')$, then it follows that for all κ_i with $1 \leqslant i \leqslant m$, that $\kappa_i(\omega) > \kappa_i(\omega')$, this contradicts that $K_{\Sigma}(E) = \sum_{i=1}^{m} \kappa_i(\omega) \leqslant \sum_{i=1}^{m} \kappa_i(\omega') = K_{\Sigma}(E)$. We can follow that (K2) holds for all combination operation from Definition 7. □

Now, that we have defined suitable mappings between ranking profiles and combined epistemic states, we will use them as coalescent assignment in order to define conditional merging operators. Note that in order to show, that the combination operators for ranking functions defined in Definition 7 are coalescent assignments that satisfy some desirable properties, we need a mapping from C to E, we assume that this mapping is given via the epistemic states of the agents corresponding to the conditional beliefs in C. We do not make any further restrictions than Δ_i-admissibility.

Theorem 5. *Let $C = \{\Delta_i\}_{i=1}^{m}$ be a conditional profile and $E = \{\kappa_i\}_{i=1}^{m}$ be a ranking profile with Δ_i-admissible ranking functions κ_i. The combination operators K_{Min}, K_{Max} and K_{Σ} define results of coalescent assignments via the mapping $C \mapsto E \mapsto K$, that satisfy (CA2).*

Proof. K_{Min}, K_{Max} and K_{Σ} are ranking functions due to K_0 from Definition 7. They satisfy postulate (CA2) for coalescent assignments, which follows straight from the Definition 1. □

We define the following conditional merging operator via the extraction of conditional information as in (1) from combined ranking functions:

Definition 8. *Let $C = \{\Delta_1, \dots, \Delta_m\}$ be a conditional profile and $E = \{\kappa_1, \dots, \kappa_m\}$ be a ranking profile with Δ_i-admissible ranking functions κ_i, i.e., $\kappa_i \models \Delta_i$. Let K be a combination operator for ranking functions. We define a* conditional merging operator via combination operators *as*

$$\Gamma_K^{co}(C) = \{(B \,|\, A) \in \bigcup C \mid K(E)(AB) < K(E)(A\overline{B})\}.$$

Using Theorem 4 we can follow:

Corollary 1. *$\Gamma_K^{co}(C)$ satisfies (CM*), (CM0) and (CM2) for each combination operation K_{Min}, K_{Max} and K_{Σ}.*

Proof. (CM*) follows immediately from the definition of Γ_K^{co}. $\Gamma_K^{co}(C)$ is defined via combination operators $K(E)$ and $K(E) \models \Gamma_K^{co}(C)$, i.e. (CM0) follows immediately. (CM2) follows from Theorem 5 via Theorem 4. □

The benefit of conditional merging via combination operators is that it enables us to take not only the beliefs of agents concerning conditional rules but also their entire current belief state including information about their preferences into account. This information is used to guide the merging process. We illustrate the influence of epistemic states of the agents on the merging process by continuing the example from Section 2:

Example 3 (Continue Example 2)**.** *Table 1 shows suitable ranking functions $\kappa_{\Delta_1}, \kappa_{\Delta_2}, \kappa_{\Delta_3} \in E$ corresponding to the financial experts' beliefs $\Delta_1, \Delta_2, \Delta_3 \in C$. For example both expert 1 and expert 2 believe that, bubbles lead to crashes, but agent 1 thinks that a bubble is very likely, whereas agent 2 thinks that a bubble is not. The merging via combination operators is also illustrated in Table 1, we use $K_{Min}(E)$, $K_{Max}(E)$ and $K_{\Sigma}(E)$ as combination operators and get the following results for $\Gamma_{Min}^{co}, \Gamma_{Max}^{co}$ and Γ_{Σ}^{co}:*

$$\begin{aligned}\Gamma_{Min}^{co}(C) &= \{(R \,|\, B), (S \,|\, B)\}\\ \Gamma_{Max}^{co}(C) &= \{(\bar{S} \,|\, BR)\}\\ \Gamma_{\Sigma}^{co}(C) &= \emptyset.\end{aligned}$$

In this example, we see that conditional relationships across conditional belief bases are not respected by the combination operators, since $(R \,|\, \bar{S})$ cannot be found in any of the merging results but is part of every maximal consistent set $\mathcal{M}_C$, i.e., $(R \,|\, \bar{S}) \in \Gamma_S^{pm}(C)$ for any selection function S.

In the following, we show which of our conditional merging operators defined via combination operators classify as majority resp. arbitration conditional merging operators:

Theorem 6. *The following statements hold:*

- *The coalescent assignments defined by K_{Min} and K_{Max} satisfy (Arb$^{\kappa}$) and therefore Γ_{Min}^{co} and Γ_{Max}^{co} are arbitration conditional merging operators.*
- *The coalescent assignment defined by K_{Σ} satisfies (Maj$^{\kappa}$) and therefore Γ_{Σ}^{co} is a majority conditional merging operator.*

Proof. For the first statement, we show that (Arb$^{\kappa}$) holds for K_{Min}, K_{Max}. Let $E = \{\kappa_1, \dots, \kappa_m, \kappa_{\Delta}\}$ and $E_n = \{\kappa_1, \dots, \kappa_m, \kappa_{\Delta}, \dots, \kappa_{\Delta}\}$ with $n-$times κ_{Δ}. Then $K_{Min}(E)(\omega) = K_0 + \min_{i=1,\dots,m}\{\kappa_i(\omega), \kappa_{\Delta}(\omega)\} = K_0 + \min_{i=1,\dots,m}\{\kappa_i(\omega), \kappa_{\Delta}(\omega), \dots, \kappa_{\Delta}(\omega)\} = K_{Min}(E_n)$ for all $\omega \in \Omega$ and therefore (Arb$^{\kappa}$) holds for all $n \in \mathbb{N}$. The same argumentation holds for K_{Max}, i.e. K_{Min} and K_{Max} define coalescent assignments that satisfy (Arb$^{\kappa}$) and via Theorem 4 we can follow that Γ_{Min}^{co} and Γ_{Max}^{co} satisfy (Arb). Now, we show that (Maj$^{\kappa}$) holds for K_{Σ}: Pick any k such that $K_{\Sigma}(C \sqcup \Delta^k)(AB) \geqslant K_{\Sigma}(C \sqcup \Delta^k)(A\overline{B})$ and $(B \,|\, A) \in \Delta$. It holds that $\kappa_{\Delta}(AB) < \kappa_{\Delta}(A\overline{B})$. Let $\tilde{n} \geqslant \frac{K_{\Sigma}(C \sqcup \Delta^n)(AB) - K_{\Sigma}(C \sqcup \Delta^n)(A\overline{B})}{\kappa_{\Delta}(A\overline{B}) - \kappa_{\Delta}(AB)}$. Note that by hypothesis $\tilde{n} > 0$. Observe that $K_{\Sigma}(C \sqcup \Delta^{k+\tilde{n}})(AB) < K_{\Sigma}(C \sqcup \Delta^{k+\tilde{n}})(A\overline{B})$. In this way, we can find a $n = k + \tilde{n}$ such that $K_{\Sigma}(C \sqcup \Delta^n)(AB) < K_{\Sigma}(C \sqcup \Delta^n)(A\overline{B})$, which contradicts our hypothesis and therefore (Maj$^{\kappa}$) holds. And via Theorem 4, we can follow that Γ_{Max}^{co} satisfies (Maj). □

The theorem shows that conditional merging operators defined via combination operations are versatile and cover different subclasses of merging operators via different combination operations K in Γ_K^{co}, making them more flexible in the merging process.

The following example shows that (CM1) and (CM3) do not hold for Γ_K^{co} and that it makes sense to refine combination operators so that they take logical dependencies between conditionals more into account:

$\omega \in \Omega$	κ_{Δ_1}	κ_{Δ_2}	K_{Min}	K_{Max}	K_Σ
ABD	0	1	0	1	1
$AB\overline{D}$	1	0	0	1	1
$A\overline{B}D$	1	1	1	1	2
$A\overline{B}\overline{D}$	1	1	1	1	2
$\overline{A}BD$	0	1	0	1	1
$\overline{A}B\overline{D}$	1	0	0	1	1
$\overline{A}\overline{B}D$	0	0	0	0	0
$\overline{A}\overline{B}\overline{D}$	0	0	0	0	0

Table 2: The table shows two admissible ranking functions for Δ_1, Δ_2 from Example 4 and the result of the combination operation of $E = \{\kappa_{\Delta_1}, \kappa_{\Delta_2}\}$ via the operators K_{Min}, K_{Max} and K_Σ. Note that, $\kappa_{\Delta_1}, \kappa_{\Delta_2}$ are the System-Z ranking functions corresponding to Δ_1, Δ_2.

Example 4. *Let* $\Delta_1 = \{(B \,|\, A), (D \,|\, B)\}$, $\Delta_2 = \{(B \,|\, A), (\overline{B} \,|\, D)\}$ *and* $C = \{\Delta_1, \Delta_2\}$. *Note that* C *is consistent and that* $(B \,|\, A) \in \bigcap C$. *As* Δ_i*-admissible ranking functions* $\kappa_{\Delta_1}, \kappa_{\Delta_2}$ *we take the System-Z ranking functions of* Δ_1 *and* Δ_2*, they can be found in Table 2. As combination operation on* $E = \{\kappa_{\Delta_1}, \kappa_{\Delta_2}\}$ *we take* $K_{Min}(E)$, $K_{Max}(E)$ *and* $K_\Sigma(E)$. *Table 2 shows the result of the combination operators and we get the following results for the conditional merging via the corresponding combination operators* $K_{Min}(E)$, $K_{Max}(E)$ *and* $K_\Sigma(E)$

$$\Gamma^{co}_{Min}(C) = \{(B \,|\, A)\}$$
$$\Gamma^{co}_{Max}(C) = \{(\overline{B} \,|\, D)\}$$
$$\Gamma^{co}_{\Sigma}(C) = \{(B \,|\, A), (\overline{B} \,|\, D)\}.$$

None of the conditional merging operators determines a combined ranking function that satisfies all conditionals in $\bigcup C$ *even though* $\bigcup C$ *is consistent. Moreover,* $\bigcap C \not\subseteq \Gamma^{co}_{Max}(C)$ *even though all agents agree that this conditional rule should be accepted. We conclude that combination operators are not sufficient to handle conditional relationships across conditional belief bases in a conditional profile.*

Admissible ranking functions for conditional belief bases Δ_i in C respect conditional dependencies within Δ_i, but, in order to merge the bases in C we also need to monitor conditional dependencies from conditionals across conditional belief bases in C.

5.3 Epistemic Conditional Merging via Combination w.r.t. Conditional Dependencies

In order to monitor conditional interactions while combining ranking functions, we need to take all conditionals in C into consideration. Therefore, we make use of ordered tolerance partitions of conditional belief bases Δ (first introduced by Pearl in (1990)) which are commonly used to define System-Z ranking functions for conditional belief bases Δ. The partition $\Delta = (\Delta_0, \ldots, \Delta_p)$ can only be determined for a consistent Δ and it defines maximal sets Δ_k ($k \in \{0, \ldots, p\}$) of conditionals that tolerate each other and are tolerated by all conditionals in $\bigcup_{j \geqslant k} \Delta_j$. We use the standard definition of tolerance introduced by Adams (Adams 1965), where a conditional is tolerated by a set of conditionals iff there exists a world that verifies the conditional and does not falsify any of the conditionals in the set of conditionals. The notion of tolerance keeps track of conditional relationships, which the sheer combination of admissible ranking functions fails to do for conditionals from different Δ_i's in C. Therefore, we extend the standard combination operators with the ability to respect tolerance relations across conditional bases in C. Since tolerance partition can only be determined for consistent sets of conditionals and $\bigcup C$ is in general not consistent, we define paraconsistent partitions for (inconsistent) sets Δ.

$\omega \in \Omega$	κ_{Δ_1}	κ_{Δ_2}	K^c_{Min}	K^c_{Max}	K^c_Σ
ABD	0	1	1	2	2
$AB\overline{D}$	1	0	2	3	4
$A\overline{B}D$	1	1	3	3	5
$A\overline{B}\overline{D}$	1	1	3	3	5
$\overline{A}BD$	0	1	1	2	5
$\overline{A}B\overline{D}$	1	0	0	1	1
$\overline{A}\overline{B}D$	0	0	0	0	0
$\overline{A}\overline{B}\overline{D}$	0	0	0	0	0

Table 3: For Example 5, the table shows κ_{Δ_1}, κ_{Δ_2} and the combined ranking functions $K_{Min}(E), K_{Max}(E)$ and $K_\Sigma(E)$, the layers of Ω are depicted using different shades of gray, where Ω_0 is white ($\zeta_0 = 0$), Ω_1 is light gray ($\zeta_1 = 2$) and $\overline{\Omega}$ is dark gray ($\overline{\zeta} = 3$). In this example the factors ζ_i and $\overline{\zeta}$ are the same for all combination operations.

Definition 9. *Let* Δ *be a set of conditionals and let* $(\Delta_0, \ldots, \Delta_p, \overline{\Delta})$ *be a partition of* Δ *such that each* $(B \,|\, A) \in \Delta_k$ *is tolerated by the conditionals in* $\bigcup_{k \leqslant j \leqslant p} \Delta_j$ *and*

$$\overline{\Delta} = \{(B \,|\, A) \in \Delta \,|\, (B \,|\, A) \textit{ is not tolerated by } \Delta_p\},$$

we call $\overline{\Delta}$ *the* conflicting subbase of Δ *and* $(\Delta_0, \ldots, \Delta_p, \overline{\Delta})$ *the* paraconsistent partition *of* Δ.

Note that Δ is consistent if and only if $\overline{\Delta} = \emptyset$. Using the paraconsistent partition of $\bigcup C$, we define conditional merging operators via combination operators that respect conditional dependencies. In order to differentiate between conditionals that tolerate each other and conditionals that do not on the level of possible worlds in Ω, we define layers of Ω, s.t. $\Omega = (\Omega_0, \ldots, \Omega_p, \overline{\Omega})$.

Definition 10. *Let* Δ *be a set of conditionals and* $\Delta = (\Delta_0, \ldots, \Delta_p, \overline{\Delta})$ *be its paraconsistent partition. We define the* paraconsistent partition of possible worlds $\Omega = (\Omega_0, \ldots, \Omega_p, \overline{\Omega})$ w.r.t Δ *as follows:*

$$\Omega_k = \Omega \setminus (\Omega^f_{\bigcup_{k \leqslant j \leqslant p} \Delta_j \cup \overline{\Delta}}) - \bigcup_{0 \leqslant l \leqslant k-1} \Omega_l$$

$$\textit{and } \overline{\Omega} = \Omega - \bigcup_{k \geqslant 0} \Omega_k.$$

$\Omega_k \subseteq \Omega = (\Omega_0, \ldots, \Omega_p, \overline{\Omega})$ is the set of worlds that have not been inserted in any layer of Ω before, and that do not falsify any of the conditionals that tolerate all conditionals in $\bigcup_{k \leqslant j \leqslant p} \Delta_j \cup \overline{\Delta}$. Since we determine the partition

$\Delta = (\Delta_0, \dots, \Delta_p, \overline{\Delta})$ via the tolerance test the corresponding layer Ω_k for Δ_k is always non-empty. Note that it is possible that $\Delta = \overline{\Delta}$ and $\Omega = \overline{\Omega}$, in this case the network of conditional dependencies is too dense to identify conditionals that are tolerated by all other conditionals across the conditional belief bases. This is a special case which we will deal with later in the paper. We illustrate the paraconsistent partition of conditional profiles and the corresponding partitions of worlds via continuing Example 4 in the upcoming Example 5.

Using the layers of Ω, we can define new combination operators that also divide the result of the combination operation K into layers and therefore respect conditional relationships:

Definition 11. *Let C be a conditional profile and $(C_0, \dots, C_p, \overline{C})$ be the paraconsistent partition of $\bigcup C$ and $\Omega = (\Omega_0, \dots, \Omega_p, \overline{\Omega})$ be the corresponding partition of Ω. Let $E = \{\kappa_1, \dots, \kappa_m\}$ with Δ_i-admissible ranking functions κ_i, we define* combination operators w.r.t. conditional dependencies $K^c(E)$ *as follows:*

$$K^c(E)(\omega) = K_0 + K(E)(\omega) + \zeta_k$$
$$\textit{with } \zeta_k = \max_{\omega \in \Omega_{k-1}} \{K^c(E)(\omega)\} + 1 \textit{ and } \zeta_0 = 0$$

for each $\omega \in \Omega_k$ and

$$K^c(E)(\omega) = K_0 + K(E)(\omega) + \overline{\zeta}$$
$$\textit{with } \overline{\zeta} = \max_{\omega \in \Omega_k} \{K^c(E)(\omega)\} + 1$$

for $\omega \in \overline{\Omega}$, where K_0 is a normalization constant that assures that K^c is a ranking function. The factors $\zeta_k, \overline{\zeta}$ separate the layers of Ω determined by $(\Omega_0, \dots, \Omega_p, \overline{\Omega})$.

The following proposition holds for all of the above defined operators and shows that $K^c(E)$ separates the plausibility ranks of worlds from different Ω_k's:

Proposition 2. *For a combination operator $K^c(E)$ as defined in Definition 11 and $\omega \in \Omega_k$, it holds that $K^c(E)(\omega) < K^c(E)(\omega')$ for $\omega' \in \Omega_{k+1}$ resp. if $\omega \in \Omega_p$ and $\omega' \in \overline{\Omega}$.*

Proof. Let $\omega \in \Omega_k$ and $\omega' \in \Omega_{k+1}$, $0 \leqslant k \leqslant p - 1$, then $K^c(E)(\omega) < \zeta_{k+1}$, i.e., $K^c(E)(\omega) < K^c(E)(\omega')$. This works in an analogous way for $\omega \in \Omega_p$ and $\omega' \in \overline{\Omega}$. □

Using different combination operations K, we can define different combination operators that respect conditional dependencies. As in the section before, we take the minimum K_{Min}, the maximum K_{Max} and the sum of worlds K_Σ and get the combination operators w.r.t. conditional dependencies K^c_{Min}, K^c_{Max} and K^c_Σ. The following theorem addresses the relationship between standard combination operators as defined in Definition 6 and combination operators that respect conditional dependencies:

Theorem 7. *Let $C = (C_0, \dots, C_p, \overline{C})$ be a conditional profile and $\Omega = (\Omega_0, \dots, \Omega_p, \overline{\Omega})$ be the corresponding partition of Ω. Let $E = \{\kappa_1, \dots, \kappa_m\}$ with Δ_i-admissible ranking functions κ_i and $K^c(E)$ be a combination operator as defined in Definition 11, then $K^c(E)$ with K_{Min}, K_{Max} and K_Σ satisfies (K0) and for $\omega, \omega' \in \Omega_k$ it satisfies (K1) and (K2).*

$\omega \in \Omega$	$\kappa_{\Delta'_1}$	$\kappa_{\Delta'_2}$	K_{Min}	K_{Max}	K_Σ
ABD	0	1	0	0	0
$AB\overline{D}$	2	3	2	3	5
$A\overline{B}D$	2	1	4	6	9
$A\overline{B}\overline{D}$	1	1	4	5	8
$\overline{A}BD$	0	2	5	9	12
$\overline{A}B\overline{D}$	3	0	5	10	13
$\overline{A}\overline{B}D$	2	1	6	9	13
$\overline{A}\overline{B}\overline{D}$	2	1	6	9	13

Table 4: The table shows two Δ'_i-admissible ranking functions $\kappa_{\Delta'_1}, \kappa_{\Delta'_2} \in E'$ from Example 5 and the combined ranking functions $K_{Min}(E'), K_{Max}(E')$ and $K_\Sigma(E')$. The layers of Ω are depicted using different shades of gray, where Ω_0 is white, Ω_1 is light gray and $\overline{\Omega}$ is dark gray. For each layer of Ω and each combination operator K_{Min}, K_{Max} and K_Σ, we get different factors ζ'_j $(j = 0, 1)$ resp. $\overline{\zeta}'$, for all combination operators it holds that $K_0 = 0$.

Proof. That K^c satisfies (K0) follows straightforward from K_0 in Definition 11. For ω, ω' from the same Ω_k the factor ζ_k is the same for both worlds, i.e. we can follow (K1) and (K2) from the fact that (K1) and (K2) hold for the combination operators K that define K^c. □

Since we add factors ζ resp. $\overline{\zeta}$ and therefore assure that K^c respects conditional dependencies within C, we cannot guarantee that (K1) and (K2) hold for world $\omega \in \Omega_k$ and $\omega' \in \Omega_{k'}$ with $k \neq k'$. For a counterexample for (K1) and (K2) from worlds $\omega \in \Omega_k$ and $\omega' \in \Omega_{k'}$ with $k \neq k'$, see Example 5 and Table 4, where $\kappa_{\Delta'_i}(A\overline{B}\overline{C}) < \kappa_{\Delta_i}(AB\overline{C})$ for $i = 1, 2$ but $K^c(E')(AB\overline{C}) < K^c(E')(A\overline{B}\overline{C})$ for each operator. Following the same line as in the previous subsection, we show that K^c defines a coalescent assignment and therefore a conditional merging operator. Again, for the coalescent assignment we assume that the mapping from C to E is given via the belief sets of the agents that participate in the merging process.

Theorem 8. *Let $C = \{\Delta_i\}_{i=1}^m$ be a conditional profile and $E = \{\kappa_i\}_{i=1}^m$ be a ranking profile with Δ_i-admissible ranking functions κ_i. The combination operators K^c_{Min}, K^c_{Max} and K^c_Σ define coalescent assignments via the mapping $C \mapsto E \mapsto K$, that satisfy (CA1) and (CA2).*

Proof. K^c_{Min}, K^c_{Max} and K^c_Σ are ranking functions due to K_0 from Definition 11. For (CA1), we show that $K^c(C)$ with $C = (C_0, \dots, C_p, \overline{C})$ respects logical relationships between conditionals outside $\overline{C}$ and therefore satisfies all conditionals in $C \backslash \overline{C}$. For a conditional in $(B \,|\, A) \in C_k$, it holds that there exists a world ω s.t. ω verifies $(B \,|\, A)$ and does not falsify any conditional from $\bigcup_{k \leqslant j \leqslant p} C_j \cup \overline{C}$, i.e. $\omega \in \Omega_l$ with $l \leqslant k$. For all $\omega' \in Mod(A\overline{B})$ it holds that $\omega' \in \Omega_{l'}$ with $k < l'$, since $\omega' \in \Omega^f_{\bigcup_{0 \leqslant j \leqslant k} C_j \cup \overline{C}}$, i.e., $K^c(E)(\omega) < K^c(E)(\omega')$, since $l < l'$ and the choice of $\zeta_{l'}$. Therefore, we can follow that $K^c(E)(AB) < K^c(E)(A\overline{B})$ for all K^c_{Min}, K^c_{Max} and K^c_{Σ} and $(B \mid A) \in \Gamma^{coc}_{K^c}(C)$. For consistent $\bigcup C$, it holds that $\overline{C} = \emptyset$ and therefore $\Gamma^{coc}_{K^c}(E) \models \bigcup C$. (CA2) follows straightforward from the Definition 1. □

We get the following conditional merging operator via the extraction of conditional information as in (1) from the combined ranking functions that respect conditional dependencies:

Definition 12. *Let $C = \{\Delta_1, \dots, \Delta_m\}$ be a conditional profile and $E = \{\kappa_1, \dots, \kappa_m\}$ be a ranking profile with Δ_i-admissible ranking functions κ_i, i.e., $\kappa_i \models \Delta_i$. Let K^c be a combination operator that respects conditional dependencies. We define a* conditional merging operator via combination operators w.r.t. conditional dependencies *as follows:*

$$\Gamma^{coc}_{K^c}(C) = \{(B \mid A) \in \bigcup C \mid K^c(E)(AB) < K^c(E)(A\overline{B})\}.$$

Using Theorem 8 we can follow:

Corollary 2. $\Gamma^{coc}_{K^c}(C)$ *satisfies (CM*),(CM0), (CM1) and (CM2) for each combination operation that respects conditional dependencies K^c_{Min}, K^c_{Max} and K^c_{Max}.*

Proof. (CM*) follows immediately from the definition of $\Gamma^{coc}_{K^c}$. $\Gamma^{coc}_{K^c}(C)$ is defined via combination operators $K^c(E)$ and $K^c(E) \models \Gamma^{coc}_{K^c}(C)$, i.e. (CM0) follows immediately. (CM1) and (CM2) follow from Theorem 5 via Theorem 8. □

We illustrate paraconsistent partitions and the corresponding partition of Ω and the combination operators from Definition 11 and continue Example 4:

Example 5 (Continue Example 4)**.** *For $C = \{\Delta_1, \Delta_2\}$ from Example 4, we get the paraconsistent partition $C = (\{(\overline{B} \mid D)\}, \{(B \mid A), (D \mid B)\}, \emptyset)$ with $\overline{C} = \emptyset$. The corresponding layers of Ω are depicted in Table 3 using different gray tones. The determination of the layers Ω_k in the partition of Ω in detail works as follows: For Ω_0, we first determine $\Omega^f_{C_0 \cup C_1 \cup \overline{C}} = \Omega^f_{\bigcup C}$*

$$\Omega^f_{\bigcup C} = \{ABD, AB\overline{D}, A\overline{B}D, A\overline{B}\overline{D}, \overline{A}BD, \overline{A}B\overline{D}\}$$
$$\Rightarrow \Omega_0 = \{\overline{A}\overline{B}D, \overline{A}\overline{B}\overline{D}\}.$$

For the next layer, we have to determine the set of worlds that falsify any conditional from $C_1 = \{(B \mid A), (D \mid B)\} \cup \overline{C}$, whereby however $\overline{C} = \emptyset$, i.e.,

$$\Omega^f_{C_1 \cup \overline{C}} = \{AB\overline{D}, A\overline{B}D, A\overline{B}\overline{D}, \overline{A}B\overline{D}\}$$

and $\Omega_1 = \Omega \setminus \Omega^f_{C_1 \cup \overline{C}} - \Omega_0 = \{ABD, \overline{A}BD\}$. The remaining worlds are part of $\overline{\Omega} = \{AB\overline{D}, A\overline{B}D, A\overline{B}\overline{D}, \overline{A}B\overline{D}\}$ and it holds that $\Omega = \Omega_0 \cup \Omega_1 \cup \overline{\Omega}$. We get the following merging result:

$$\Gamma^{co}_{Min}(C) = \Gamma^{co}_{Max}(C) = \Gamma^{co}_{\Sigma}(C) = \bigcup C.$$

This result corresponds to the intuition that all conditionals in $\bigcup C$ should be accepted. Note that as Δ_i-admissible ranking functions in $E = \{\kappa_{\Delta_1}, \kappa_{\Delta_2}\}$ we used the System-Z ranking functions as in Example 4, but this does not contribute to the fact that $\Gamma^{co}_{Min}(C) = \bigcup C$ since Proposition 2 holds for all combinations on general Δ_i-admissible ranking functions. To illustrate conditional combination operators for inconsistent conditional profiles, we modify Δ_1, Δ_2 and take $\Delta'_1 = \{(B \mid A), (D \mid \overline{A}), (A \mid \overline{B})\}$ and $\Delta'_2 = \{(B \mid A), (\overline{D} \mid \overline{A})\}$, so that $C' = \{\Delta'_1, \Delta'_2\}$ is inconsistent and we get the following paraconsistent partition $C' = (C'_1, C'_2, \overline{C'}) = (\{(B \mid A)\}, \{(A \mid \overline{B})\}, \{(D \mid \overline{A}), (\overline{D} \mid \overline{A})\})$. The layers of $\Omega = (\Omega'_0, \Omega'_1, \overline{\Omega}')$ can be found in Table 4 depicted in different gray tones. We get the following merging result:

$$\Gamma^{co}_{Min}(C') = \Gamma^{co}_{Max}(C') = \{(B \mid A), (A \mid \overline{B})\}$$
$$\Gamma^{co}_{\Sigma}(C') = \{(B \mid A), (A \mid \overline{B}), (D \mid \overline{A})\}.$$

Example 5, shows that the combination with respect to logical dependencies among conditionals leads to intuitive merging results. For conditionals in $\overline{C}$, the combination operators yield different results but the least conflicting conditionals outside of $\overline{C}$ are always accepted.

The following theorem classifies Γ^{coc}_{Min} and Γ^{coc}_{Max} as arbitration conditional merging operators:

Theorem 9. *The coalescent assignments defined by K^c_{Min} and K^c_{Max} satisfy (Arb$^\kappa$) and therefore Γ^{coc}_{Min} and Γ^{coc}_{Max} satisfy (Arb).*

Proof. It holds that $C \sqcup \Delta = C' = (C'_0, \dots, C'_p, \overline{C'}) = C \sqcup \Delta^n = C'_n$, i.e., the layers of $\Omega = (\Omega'_0, \dots, \Omega'_p, \overline{\Omega'})$ that correspond to C' and C'_n are the same. Thus, the constants ζ_k that seperate the layers in K^c_{Min} resp. K^c_{Max} are the same and we can follow (Arb$^\kappa$) for K^c_{Min} and K^c_{Max} from (Arb$^\kappa$) for K_{Min} resp. K_{Max} from the proof of Theorem 6 and therefore (Arb) holds for Γ^{coc}_{Min} resp. Γ^{coc}_{Min}. □

Note that, Γ^{coc}_{Σ} also prefers majorities in the merging process but only within the conditionals of $\overline{C}$, the strict separation of layers prevents the sheer overvaluation of majorities in C and therefore Γ^{coc}_{Σ} does not satisfy (Maj) in general.

We also continue the example from Section 2, to illustrate the results of the conditional combination operators:

Example 6 (Continue Example 3)**.** *It holds that $C = (C_0, \overline{C})$ with $C_0 = \{(\bar{S} \mid R)\}$ and $\overline{C} = \{(R \mid B), (S \mid B), (\bar{S} \mid BR)\}$, i.e. $\Omega_0 = \{\overline{B}R\overline{S}, \overline{B}\overline{R}S, \overline{B}\overline{R}\overline{S}\}$ and $\overline{\Omega} = \Omega \setminus \Omega_0$ as depicted in Table 5 alongside the conditional combination operators K^c_{Min}, K^c_{Max} and K^c_{Σ}. We get the following results for $\Gamma^{coc}_{Min}, \Gamma^{coc}_{Max}$ and Γ^{coc}_{Σ}:*

$$\Gamma^{coc}_{Min}(C) = \{(\bar{S} \mid R), (R \mid B), (S \mid B)\}$$
$$\Gamma^{coc}_{Max}(C) = \{(\bar{S} \mid R), (\bar{S} \mid BR)\}$$
$$\Gamma^{coc}_{\Sigma}(C) = \{(\bar{S} \mid R)\}.$$

Note that, we get similar results as for the merging operation with standard combination operators, but $(\bar{S} \mid R)$ is added to each set of $\Gamma^{coc}_{K^c}$, i.e. we respect conditional interactions inbetween C and the overall beliefs of every agent are still taken into account. This is a great advantage of merging

$\omega \in \Omega$	κ_{Δ_1}	κ_{Δ_2}	κ_{Δ_3}	K^c_{Min}	K^c_{Max}	K^c_{Σ}
BRS	0	2	3	1	5	8
$BR\overline{S}$	2	1	2	2	4	8
$B\overline{R}S$	2	2	1	2	4	8
$B\overline{R}\overline{S}$	1	2	2	2	4	8
$\overline{B}RS$	1	1	0	1	3	5
$\overline{B}R\overline{S}$	0	1	1	0	1	2
$\overline{B}\overline{R}S$	0	1	0	0	1	1
$\overline{B}\overline{R}\overline{S}$	0	0	0	0	0	0

Table 5: The table shows three Δ_i-admissible ranking functions $\kappa_{\Delta_1}, \kappa_{\Delta_2}, \kappa_{\Delta_3} \in E$ for Example 6 and the combined ranking functions $K_{Min}(E), K_{Max}(E)$ and $K_{\Sigma}(E)$. The layers of Ω are depicted using different shades of gray, where Ω_0 is white and Ω_1 is light gray. For each layer of Ω and each combination operator $K^c_{Min}(E)$, $K^c_{Max}(E)$ and $K^c_{\Sigma}(E)$, we get different factors ζ_j $(j = 0, 1)$ resp. $\overline{\zeta}$, for all combination operators it holds that $K_0 = 0$.

operators for conditional beliefs determined by conditional combination operators over conditional merging operators via maximal consistent sets.

6 Discussion and Conclusion

Merging conditional belief bases, i.e. synthesizing different, possibly conflicting, inference rules, is a challenging task that occurs in various scenarios, especially in multi-agent systems. For propositional merging operators, we can find a variety of approaches in literature (see (Konieczny and Pérez 2011) for an overview), but, since the logical structure of conditionals differs fundamentally from the one of propositions new challenges emerge. In this paper, we proposed two suitable approaches to conditional merging, the first one is based on the syntax of the agents' beliefs and the second one on the belief states underlying these beliefs, and provided for both ways suitable conditional merging operators respecting logical dependencies between conditionals. The following example illustrates why our epistemic merging operators do not accept all conditionals in the intersection of a conditional profile in general:

Example 7. *Let* $\Delta_1 = \{(B \,|\, A), (A \,|\, \overline{B})\}$, $\Delta_2 = \{(B \,|\, A), (\overline{B} \,|\, \overline{A})\}$ *and* $\Delta_3 = \{(B \,|\, A), (B \,|\, \overline{A})\}$ *and* $C = \{\Delta_1, \Delta_2, \Delta_3\}$. *It holds that no conditional in* $\bigcup C$ *is tolerated by the other conditionals and therefore* $C = (\overline{C})$ *and* $\Omega = \overline{\Omega}$. *The net of logical dependencies between conditionals is too dense and it holds that* $K(E) = K^c(E)$ *with* $E = \{\kappa_{\Delta_i}\}_{i=1,2,3}$ *and therefore we cannot guarantee the acceptance of* $(B \,|\, A) = \bigcap C$ *for all* Δ_i*-admissible ranking functions (including System-Z ranking functions) independent of the choice of combination operators.*

A future challenge will be to gain control of conditional beliefs across a conditional profile that highly interact with each other. Another challenge we want to address in this discussion, which is also related to the network of logical dependencies within C, is to establish conditional merging operators that satisfy a conditional version of Arrow's Pareto principle (Arrow 1963), i.e. two groups should not agree on a conditional that is not previously accepted by the compromise within at least one of the groups. Similar problems with combination operators on epistemic states are addressed in (Meyer 2001). From our point of view this is due to the shift from propositions to more complex conditional belief bases resp. ranking functions.

References

Adams, E. 1965. The logic of conditionals. *Inquiry: An Interdisciplinary Journal of Philosophy* 8(1-4):166–197.

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic* 50(2):510–530.

Arrow, K. 1963. *Social Choice and Individual Values*. Wiley: New York, second edition.

Baral, C.; Kraus, S.; and Minker, J. 1991. Combining multiple knowledge bases. *IEEE Trans. Knowl. Data Eng.* 3(2):208–220.

De Finetti, B. 1975. *Theory of Probability: A critical introductory treatment*. Wiley Series in Probability and Statistics. Wiley.

Halpern, J. Y. 2005. *Reasoning about uncertainty*. MIT Press.

Konieczny, S., and Pérez, R. P. 1998. On the logic of merging. In Cohn, A. G.; Schubert, L. K.; and Shapiro, S. C., eds., *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998*, 488–498. Morgan Kaufmann.

Konieczny, S., and Pérez, R. P. 2002. Merging information under constraints: A logical framework. *J. Log. Comput.* 12(5):773–808.

Konieczny, S., and Pérez, R. P. 2011. Logic based merging. *J. Philos. Log.* 40(2):239–270.

Liberatore, P., and Schaerf, M. 1998. Arbitration (or how to merge knowledge bases). *IEEE Trans. Knowl. Data Eng.* 10(1):76–90.

Meyer, T. A. 2001. On the semantics of combination operations. *J. Appl. Non Class. Logics* 11(1-2):59–84.

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In Parikh, R., ed., *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, USA, March 1990*, 121–135.

Revesz, P. Z. 1997. On the semantics of arbitration. *Int. J. Algebra Comput.* 7(2):133–160.

Spohn, W. 1988. Ordinal conditional functions: A dynamic theory of epistemic states. In Harper, W. L., and Skyrms, B., eds., *Causation in Decision, Belief Change, and Statistics*. Dordrecht: Springer. 105–134.

Spohn, W. 2014. *The Laws of Belief - Ranking Theory and Its Philosophical Applications*. Oxford University Press.

A Reduct-Driven Study of Argumentation Frameworks With Collective Attacks

Wolfgang Dvořák , **Matthias König** , **Markus Ulbricht** and **Stefan Woltran**
TU Wien, Institute of Logic and Computation
{dvorak,mkoenig,mulbricht,woltran}@dbai.tuwien.ac.at

Abstract

In this paper, we investigate structural properties of argumentation frameworks with collective attacks (SETAFs) which generalize Dung-style argumentation frameworks (AFs) to scenarios where some arguments can only be defeated by a set of arguments. We propose a notion of a *reduct* for SETAFs which extends a recent proposal for Dung AFs. We show how recent results regarding the reduct extend to the more general SETAFs providing further evidence that they constitute a natural generalization of classical AFs. These results include basic properties of semantics w.r.t. their reduct, a compatibility requirement (so-called modularization property) as well as a formal tool to trace non-determinism back to even cycle arguments. Moreover, we compare SETAFs to logic programs and show relations between our reduct notion and established ones in the domain of logic programming.

1 Introduction

Abstract argumentation frameworks (AFs) as proposed by Dung (1995) in his seminal paper are nowadays a classical research area in knowledge representation and reasoning. In an AF, arguments are interpreted as abstract entities and thus the focus is solely on the relationship between them, i.e. which arguments are in conflict with each other. Consequently, an AF is simply a directed graph, where the vertices are interpreted as arguments and edges as attacks between them. AFs have been thoroughly investigated over the last decade and various extensions have been proposed in order to extend their expressive power. For example, researchers considered the addition of supports (Cayrol and Lagasquie-Schiex 2005), recursive attacks (Baroni et al. 2011), claims (Dvořák, Rapberger, and Woltran 2020a), or probabilities (Thimm 2012) to mention a few.

In the present paper we consider Argumentation Frameworks with collective attacks (SETAFs), introduced by Nielsen and Parsons (2006). SETAFs generalize Dung-style AFs in the sense that some arguments can only be effectively defeated by a collection of attackers, yielding a natural representation as a directed hypergraph. Many key semantic properties of AFs have been shown to carry over to SETAFs, see e.g. (Nielsen and Parsons 2006; Flouris and Bikakis 2019). Moreover, work has been done on expressiveness (Dvořák, Fandinno, and Woltran 2019), and translations from SETAFs to AFs (Polberg 2017; Flouris and Bikakis 2019). Also the hypergraph structure of SETAFs has recently been subject of investigation (Dvořák, König, and Woltran 2021a; Dvořák, König, and Woltran 2021b). In this work we complement existing research by introducing a reduct notion for SETAFs, that enables us to give alternative characterization of the semantics, show a modularization property, and characterize extensions via explanation schemes. Moreover, we will use the reduct to investigate the correspondence between SETAFs and atomic logic programs.

As AFs and SETAFs (Dvořák and Dunne 2017; Dvořák, Greßler, and Woltran 2018) constitute computationally hard problems, researchers have investigated techniques to divide a given framework in a way that extensions can be computed step-wise. While traditional approaches are using the graph-structure to divide the framework (Baroni et al. 2014; Baroni, Giacomin, and Guida 2005; Baumann 2011) the recently introduced *modularization property* for AFs (Baumann, Brewka, and Ulbricht 2020a) does not make any assumptions on the structure of the graph. The modularization property formalizes a compatibility requirement between the accepted arguments within a given extension and those whose acceptance status is not yet determined. That is, extensions of different sub-frameworks can be merged together in order to find a novel extension of the whole framework. The key underlying concept is the so-called E-reduct of a given AF F w.r.t. an extension E. Intuitively, this reduct is a tool to analyze the behavior of an AF when a certain set of arguments is set to true, similar in spirit to the Gelfond-Lifschitz-Reduct for logic programs. In order to define a modularization property for SETAFs we first introduce a notion of E-reduct for SETAFs that is a proper generalization of the E-reduct for AFs.

Apart from being the main tool formalizing the modularization property, the reduct has also been utilized to propose novel semantics (Baumann, Brewka, and Ulbricht 2020b) as well as formalizing how non-determinism in AFs can be traced back to arguments occurring in even cycles (Baumann and Ulbricht 2021). In this work we pick up the latter and introduce explanation schemes for SETAF that characterize complete extensions by choices on the arguments occurring in even cycles.

As mentioned earlier the introduced E-reduct is in spirit of the Gelfond-Lifschitz reduct for logic programs. Logic programs are a well-established knowledge representation formalism. They are not only well-understood from a theoretical point of view, but also utilized in many application scenarios. A considerable amount of research is devoted to comparing logic programs with AFs, see e.g. (Caminada et al. 2015). We thus consider a correspondence between atomic logic programs and SETAFs and study the E-reduct equivalent on logic programs. This equivalence between the reducts then results in an equivalence between argumentation and logic programming semantics.

The main contribution of this paper is to show that our natural extension of the reduct notion for AFs is well-behaving for SETAFs as well. We show that basic properties are preserved, as well as their implications in terms of the structure of extensions. More specifically the paper is organized as follows.

- After giving necessary preliminaries in Section 2 we introduce the E-reduct SF^E for a SETAF SF and a set E of arguments and investigate its core properties, including the modularization property for SETAFs (Section 3),
- we demonstrate how non-determinism in SETAFs can be traced back to even cycles (Section 4), leading to the notion of explanation schemes for SETAFs,
- in Section 5 we show how our notion of the reduct of a SETAF has a natural correspondence to the reduct of atomic logic programs; thereby, we also show how to translate SETAFs into logic programs,
- finally, we conclude in Section 6.

Notice that some technical details are omitted due to the lack of space.

2 Preliminaries

We briefly recall the definitions of SETAFs and its semantics (see, e.g., (Bikakis et al. 2021)). Throughout the paper, we assume a countably infinite domain $\mathfrak{A}$ of possible arguments.

Definition 2.1. A SETAF is a pair $SF = (A, R)$ where $A \subseteq \mathfrak{A}$ is finite, and $R \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the attack relation. For an attack $(T, h) \in R$ we call T the *tail* and h the *head* of the attack. SETAFs (A, R), where for all $(T, h) \in R$ it holds that $|T| = 1$, amount to (standard Dung) AFs. In that case, we usually write (t, h) to denote the set-attack $(\{t\}, h)$. Moreover, for a SETAF $SF = (A, R)$, we use $A(SF)$ and $R(SF)$ to identify its arguments A and its attack relation R, respectively.

Given a SETAF (A, R), we write $S \mapsto_R a$ if there is a set $T \subseteq S$ with $(T, a) \in R$. Moreover, we write $S' \mapsto_R S$ if $S' \mapsto_R a$ for some $a \in S$. We drop subscript R in $\mapsto_R$ if there is no ambiguity. For $S \subseteq A$, we use S^+_R to denote the set $\{a \mid S \mapsto_R a\}$ and define the *range* of S (w.r.t. R), denoted $S^\oplus_R$, as the set $S \cup S^+_R$.

Example 2.2. Consider the SETAF $SF = (A, R)$ with $A = \{a, b, c, d, e, f\}$ and $R = \{(a, b), (a, d), (\{b, e\}, f), (\{a, b\}, d), (\{d, e\}, a), (f, c), (f, e)\}$ (see Example 3.2(a); the three collective attacks are colored).

We will now identify special 'kinds' of attacks and fix the notions of redundancy-free and self-attack-free SETAFs.

Definition 2.3. Given a SETAF $SF = (A, R)$, an attack $(T, h) \in R$ is *redundant* if there is a $(T', h) \in R$ with $T' \subset T$. A SETAF without redundant attacks is *redundancy-free*. An attack $(T, h) \in R$ is a *self-attack* if $h \in T$. A SETAF without self-attacks attacks is *self-attack-free*.

Redundant attacks can be efficiently detected and then be omitted without changing the standard semantics (Dvořák, Rapberger, and Woltran 2020b; Polberg 2017). In the following we always assume redundancy-freeness for all SETAFs, unless stated otherwise. The well-known notions of conflict and defense from classical Dung-style-AFs naturally generalize to SETAFs.

Definition 2.4. Given a SETAF $SF = (A, R)$, a set $S \subseteq A$ is *conflicting* in SF if $S \mapsto_R a$ for some $a \in S$. A set $S \subseteq A$ is *conflict-free* in SF, if S is not conflicting in SF, i.e. if $T \cup \{h\} \not\subseteq S$ for each $(T, h) \in R$. *cf*(SF) denotes the set of all conflict-free sets in SF.

Definition 2.5. Given a SETAF $SF = (A, R)$, an argument $a \in A$ is *defended* (in SF) by a set $S \subseteq A$ if for each $B \subseteq A$, such that $B \mapsto_R a$, also $S \mapsto_R B$. A set $T \subseteq A$ is defended (in SF) by S if each $a \in T$ is defended by S (in SF).

Moreover, we make use of the *characteristic function* Γ_{SF} of a SETAF $SF = (A, R)$, defined as $\Gamma_{SF}(S) = \{a \in A \mid S \text{ defends } a\}$ for $S \subseteq A$.

The semantics we study in this work are the grounded, admissible, complete, preferred, stable semantics, which we will abbreviate by *grd*, *adm*, *com*, *pref*, *stb*, respectively (Flouris and Bikakis 2019; Nielsen and Parsons 2006).

Definition 2.6. Given a SETAF $SF = (A, R)$ and a conflict-free set $S \in cf(SF)$. Then,

- $S \in adm(SF)$, if S defends itself in SF,
- $S \in com(SF)$, if $S \in adm(SF)$ and $a \in S$ for all $a \in A$ defended by S,
- $S \in grd(SF)$, if $S = \bigcap_{T \in com(SF)} T$,
- $S \in pref(SF)$, if $S \in adm(SF)$ and there is no $T \in adm(SF)$ s.t. $T \supset S$,
- $S \in stb(SF)$, if $S \mapsto a$ for all $a \in A \setminus S$,

The relationship between the semantics has been clarified in (Dvořák, Greßler, and Woltran 2018; Flouris and Bikakis 2019; Nielsen and Parsons 2006) and matches with the relations between the semantics for Dung AFs, i.e. for any SETAF SF:

$$stb(SF) \subseteq pref(SF) \subseteq com(SF) \subseteq adm(SF) \subseteq cf(SF)$$

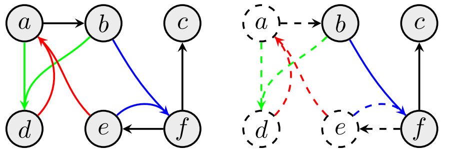
3 Introducing the SETAF-Reduct

In the remainder of this paper, we will investigate the properties of the *reduct* of a SETAF w.r.t. a given set E. Intuitively, the reduct w.r.t. E represents the SETAF that result from "accepting" E and rejecting what cannot be defended now, while not deciding on the remaining arguments.

Definition 3.1. Given a SETAF $SF = (A, R)$ and $E \subseteq A$, the *E-reduct of SF* is the SETAF $SF^E = (A', R')$, with

$$\begin{aligned} A' &= A \setminus E^\oplus_R \\ R' &= \{(T \setminus E, h) \mid (T, h) \in R,\, T \cap E^+_R = \emptyset, \\ &\qquad T \not\subseteq E,\, h \in A'\} \end{aligned}$$

Example 3.2. Consider SF and its reduct $SF^{\{d,e\}}$.



(a) SETAF SF (b) $SF^{\{d,e\}}$

In the reduct SF^E, we only need to consider arguments that are still undecided, i.e., all arguments neither in E nor attacked by E. In contrast to the AF-reduct (Baumann, Brewka, and Ulbricht 2020a), here some attacks that involve deleted arguments are preserved. In particular, if the arguments in the tail of an attack are "accepted" (i.e., in E), the attack can still play a role in attacking or defending. However, if the tail of an attack (T, h) is already attacked by E, we can disregard (T, h).

We start with a technical lemma to settle basic properties of the reduct.

Lemma 3.3. *Given a SETAF $SF = (A, R)$ and two disjoint sets $E, E' \subseteq A$. Let $SF^E = (A', R')$.*

1. *If there is no $S \subseteq A$ s.t. $S \mapsto_R E'$, then the same is true in SF^E.*
2. *Assume E does not attack $E' \in cf(SF)$. Then, E defends E' iff there is no $S' \subseteq A'$ s.t. $S' \mapsto_{R'} E'$.*
3. *Let $E \in cf(SF)$. If $E \cup E'$ does not attack E in SF and $E' \subseteq A'$, with $E' \in cf\left(SF^E\right)$ then $E \cup E' \in cf(SF)$.*
4. *Let $E \cup E' \in cf(SF)$. If $E' \mapsto_{R'} a$, then $E \cup E' \mapsto_R a$.*
5. *If $E \cup E' \in cf(SF)$, then $SF^{E\cup E'} = \left(SF^E\right)^{E'}$.*

We are now ready to present the main result of this section, the *modularization property*, generalizing the respective result from Dung AFs (Baumann, Brewka, and Ulbricht 2020a). In particular, it allows us to build admissible sets and complete extensions iteratively. After finding such a set $E \subseteq A$ we can efficiently compute its reduct SF^E and pause before evaluating the remaining (sub-)framework to find admissible/complete supersets $E' \supset E$. Hence, this first step can be seen as an intermediate result that enables us to reduce the computational effort of finding extensions in SF, as the arguments whose status is already determined by accepting E do not have to be considered again. Instead, we can reason on the reduct SF^E.

Theorem 3.4 (Modularization Property)**.** *Let SF be a SETAF, $\sigma \in \{adm, com\}$ and $E \in \sigma(SF)$.*

1. *If $E' \in \sigma(SF^E)$, then $E \cup E' \in \sigma(SF)$.*
2. *If $E \cap E' = \emptyset$ and $E \cup E' \in \sigma(SF)$, then $E' \in \sigma(SF^E)$.*

Proof. Let $SF^E = (A', R')$. First consider $\sigma = adm$.

1) Since E is admissible and $E' \subseteq A'$, E' does not attack E. By Lemma 3.3, item 3, $E \cup E' \in cf(SF)$. Now assume $S \mapsto_R E \cup E'$. If $S \mapsto_R E$, then $E \mapsto_R S$ by admissibility of E. If $S \mapsto_R E'$, there is $T \subseteq S$ s.t. $(T, e') \in R$ for some $e' \in E'$. In case $E \mapsto_R T$, we are done. Otherwise, $(T \setminus E, e') \in R'$ and by admissibility of E' in SF^E, $E' \mapsto_{R'} T \setminus E$. By Lemma 3.3, item 4, $E \cup E' \mapsto_R T \setminus E$.

2) Now assume $E \cup E' \in adm(SF)$. We see $E' \in cf\left(SF^E\right)$ as follows: If $(T', e') \in R'$ for $T' \subseteq E'$ and $e' \in E'$, then there is some $(T, e') \in R$ with $T' = T \setminus E$. Hence $E \cup E' \mapsto E'$, contradiction. Now assume T' is not admissible in SF^E, i.e. there is $(T', e') \in R'$ with $e' \in E'$ and E' does not counterattack T' in SF^E. Then there is some $(T, e') \in R$ with $T' = T \setminus E$ and $T \cap E_R^+ = \emptyset$. By admissibility of $E \cup E'$, $E \cup E' \mapsto_R T$, say $(T^*, t) \in R$, $T^* \subseteq E \cup E'$ and $t \in T$. Since $E \cup E'$ is conflict-free, $T^* \cap E_R^+ = \emptyset$ and thus we either have a) $T^* \subseteq E$, contradicting $T \cap E_R^+ = \emptyset$, or b) $(T^* \setminus E, t) \in R'$ and $t \in T'$, i.e. E' counterattacks T' in SF^E contradicting the above assumption.

Now consider $\sigma = com$.

1) We have $E \cup E' \in adm(SF)$ by the above considerations. Moreover, E' is complete, i.e. $(SF^E)^{E'}$ does not contain unattacked arguments in the reduct SF^E (see Proposition 3.5). Lemma 3.3, item 5, implies that $SF^{E\cup E'}$ does not contain unattacked arguments, either. Hence $E \cup E' \in com(SF)$.

2) Given $E \cup E' \in com(SF)$ we have $E' \in adm\left(SF^E\right)$ by the above considerations. Regarding completeness, we again use the fact that $SF^{E\cup E'} = (SF^E)^{E'}$ does not contain unattacked arguments. □

Note that the modularization property also holds for *stb* and *pref* semantics. However, the only admissible set in the reduct w.r.t. a stable/preferred extension is the empty set, rendering the property trivial. The exact relation is captured by the following alternative characterizations of the semantics under our consideration.

Proposition 3.5. *Let $SF = (A, R)$ be a SETAF, $E \in cf(SF)$ and $SF^E = (A', R')$.*

1. *$E \in stb(SF)$ iff $SF^E = (\emptyset, \emptyset)$,*
2. *$E \in adm(SF)$ iff $S \rightarrow_R E$ implies $S \setminus E \not\subseteq A'$,*
3. *$E \in pref(SF)$ iff $E \in adm(SF)$ and $\bigcup adm\left(SF^E\right) = \emptyset$,*
4. *$E \in com(SF)$ iff $E \in adm(SF)$ and no argument in SF^E is unattacked.*

Proof. The characterizations for *stb* and *adm* are straightforward and *pref* is due to the modularization property of *adm*. For *com*(SF) we apply Lemma 3.3, item 2, to each singleton E' occurring in SF^E. □

4 Explanation Schemes

Now that we established the modularization property for SETAFs, we will define *explanation schemes* using even length cycles and show their usefulness in computing complete extensions. Cycles on directed hypergraphs can be defined in various ways, we use the notion of the *primal-cycle* (Dvořák, König, and Woltran 2021a). The results of this section also hold for *set-cycles* and *incidence-cycles* (Dvořák, König, and Woltran 2021b), as discussed in Appendix A.

Definition 4.1. Let $SF = (A, R)$ be a SETAF. Then its *primal graph* is defined as *Primal*$(SF) = (A', R')$, where

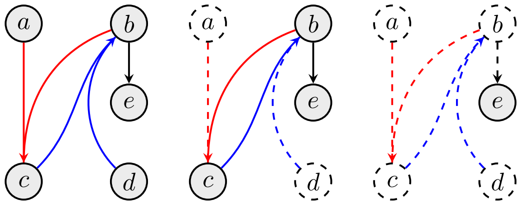
$A' = A$, and $R' = \{(t,h) \mid (T,h) \in R, t \in T\}$. A *cycle* of length n in SF is a sequence $(a_1, a_2, \dots, a_n, a_1)$ in its primal graph *Primal*(SF) such that all a_i are distinct, $(a_n, a_1) \in R'$, and $(a_i, a_{i+1}) \in R'$ for $1 \le i \le n-1$. By *Ev*(SF) we denote the set of arguments occurring in an even length cycle in SF.

On AFs, explanation schemes have been defined in (Baumann and Ulbricht 2021); in what follows we generalize these definitions and results to SETAFs. The idea is to restrict the solution space to the power set of the arguments appearing in even length cycles. After guessing a suitable subset of these arguments, complete extensions can be obtained by propagation (i.e. iteration of the characteristic function). This guessed set serves as an *explanation* for the resulting extension.

Definition 4.2. Given a SETAF $SF = (A, R)$ and a set $X \subseteq A$. The triple (E_0, E_1, E_2) is an *explanation scheme* whenever

- $E_0 \in$ *grd*(SF),
- $E_1 \subseteq$ *Ev*(SF^{E_0}) with $E_1 \in$ *cf*(SF^{E_0}), and
- $E_2 \in$ *grd*$\big((SF^{E_0})^{E_1}\big)$.

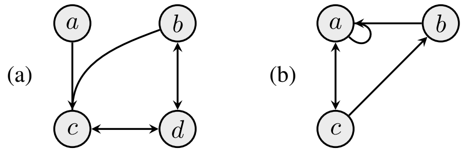
Example 4.3. The explanation scheme $(\{a,d\}, \{c\}, \{e\})$.



(a) SETAF SF (b) SF^{E_0} (c) $(SF^{E_0})^{E_1}$

If $X \subseteq E_0 \cup E_1 \cup E_2$ then we say (E_0, E_1, E_2) is an explanation scheme for X. In the Definition 4.2 we require $E_1 \in$ *cf*(SF^{E_0}), whereas in the corresponding definition for AFs we want $E_1 \in$ *cf*(F). For AFs $F = (A, R)$ we have for any sets $X, Y \subseteq A$ and $X \in$ *cf*(F) that $X' = X \cap A(F^Y) \in$ *cf*(F^Y). Hence, $E_1 \in$ *cf*(F) iff $E_1 \in$ *cf*(F^{E_0}). For SETAFs, this does not hold, as Example 4.4 illustrates.

Example 4.4. In (a) we have $\{b,c\} \in$ *cf*(SF), but $\{b,c\} \notin$ *cf*$(SF^{\{a\}})$.



We are only interested in explanation schemes with certain properties, which we define in the following.

Definition 4.5. An explanation scheme (E_0, E_1, E_2) is *successful* if $E_0 \cup E_1 \cup E_2$ is conflict-free in SF and defends E_1 in SF.

For AFs, every explanation scheme that defends E_1 is also conflict-free. However, this is not the case for SETAFs, as Example 4.4(b) illustrates. The explanation scheme $(\emptyset, \{a\}, \{b\})$ defends $\{a\}$, but is not conflict-free in SF.

Towards the full characterization, first we show that successful explanation schemes capture complete extensions.

Lemma 4.6. *For every $SF = (A, R)$ and every successful explanation scheme (E_0, E_1, E_2) in SF we have $E = E_0 \cup E_1 \cup E_2 \in$ com(SF).*

Proof. E is conflict-free and defends E_1 by definition. By construction also E_0 and E_2 are defended, hence, $E \in$ *adm*(SF). Moreover, as $E_2 \in$ *com*$((SF^{E_0})^{E_1})$ and by Lemma 3.3, item (5), SF^E has no unattacked argument, i.e. $E \in$ *com*(SF). □

The following result generalizes from AFs. It formalizes the intuition that in the reduct of a SETAF SF w.r.t. $E \subseteq A$ an argument defended by a set $X \subseteq A(SF^E)$ is also defended by $E \cup X$ in SF and vice versa.

Lemma 4.7. *Let $SF = (A, R)$ be a SETAF, $E \subseteq A$ and $SF' = SF^E = (A', R')$. Then for any $X \subseteq A(SF')$, $\Gamma_{SF'}(X)$ is the set of arguments in A' which is defended by $E \cup X$ in SF.*

Proof. ($\subseteq$) Let $e \in \Gamma_{SF'}(X)$, i.e. e is defended by X in SF'. That means for any attack $(T, e) \in R$ we either have (i) $T \cap E^+_R \neq \emptyset$, in which case E defends e in SF, or (ii) $T \cap E^+_R = \emptyset$, and since $e \in A(SF')$, $T \not\subseteq E$, i.e. $T' = T \setminus E \neq \emptyset$ with $(T', e) \in R'$. As e is defended by $\Gamma_{SF'}(X)$ in SF', $\Gamma_{SF'}(X) \mapsto_{R'} T'$, and $\Gamma_{SF'}(X) \cup E \mapsto_R T$.

($\supseteq$) Assume $e \in A'$ is defended by $E \cup X$ in SF. Let $(T, e) \in R$ be an arbitrary attack towards e. If $E \mapsto_R T$, the attack does not appear in SF' and e is defended against it in SF'. Otherwise, there is a counterattack (S, t) with $t \in T$, $S \subseteq E \cup X$ and $S \cap X \neq \emptyset$. Hence, there is $(S', t) \in R'$ with $S' \subseteq X$, i.e. e is defended by X in SF'. □

Hinting at the importance of even-cycles for complete extensions, we now establish that complete extensions other than the grounded contain arguments from even-cycle. Ultimately, we will show that these arguments explain *all* complete extensions.

Lemma 4.8. *Let SF be a SETAF with $E \in$ com$(SF) \setminus$ grd(SF). Then there is some $a \in E$ with $a \in$ Ev(SF).*

Proof. Let $E, E' \in$ *com*(SF) and assume none of them contains an argument occurring in an even-cycle. Let SF' be the SETAF after adding the attack $(\{x\}, x)$ for each $x \in$ *Ev*(SF). Still, $E, E' \in$ *com*(SF'). Now let SF'' be the SETAF after deleting each attack (T, h) with $h \in$ *Ev*(SF) and $T \cap$ *Ev*$(SF) \neq \emptyset$ from SF'. As these attacks did not help to defend any arguments in E or E' and moreover cannot be defended by any conflict-free set themselves, we still have $E, E' \in$ *com*(SF''). But as SF'' is even-(primal)-cycle-free, the only complete extension is the grounded extension (Dvořák, König, and Woltran 2021a). □

Lemma 4.9. *Let $SF = (A, R)$ be a SETAF and let $E \in$ com(SF). For any $E' \subseteq E$ we have $E'' = E \setminus E'$ is satisfying (i) $E' \cup E'' = E$, and (ii) $E'' \in$ com$(SF^{E'})$.*

Proof. Let $SF' = SF^{E'} = (A', R')$. (i) is true by definition. For (ii) we first show $E'' \in \mathit{cf}(SF')$: any possible attack $(T', h) \in R'$ with $T' \cup \{h\} \subseteq E''$ that would violate its conflict-freeness corresponds to an attack $(T, h) \in R$ with $T' \subseteq T$. Since $(T', h) \in R'$, we have $T \cap E'^{+}_{R} = \emptyset$. That means $T \setminus T' \subseteq E'$, and consequently $T \cup \{h\} \subseteq E$, which is a contradiction to the assumption that E is conflict-free in SF. Hence, no such attack can exist. Next, we show that E'' contains every argument it defends in SF'. As by Lemma 3.3, item (5), it holds $SF^E = (SF')^{E''}$ and E is complete, there are no unattacked arguments in $(SF')^{E''}$. Finally, it remains to show that E'' defends itself in SF'. As clearly $E'' \subseteq A(SF')$ and $E'' \subseteq A \setminus E'^{\oplus}_{R}$ is the set defended by $E = E' \cup E''$ in SF, Lemma 4.7 applies and we know E'' is defends itself in SF'. □

Finally, we state the decomposition property: complete extensions correspond to a successful explanation scheme, and every successful explanation scheme characterizes a complete extension. In fact, several successful explanation schemes can correspond to the same complete extension. As a successful explanation scheme is uniquely defined by E_1, this means that no two complete extensions coincide on their even-cycle arguments. Hence, exactly these arguments serve well as an explanation for the extension as a whole.

Theorem 4.10. *A set $E \subseteq A$ is in $com(SF)$ iff E can be decomposed into a successful explanation scheme.*

Proof. The $(\Leftarrow)$ direction is covered by Lemma 4.6. For $(\Rightarrow)$ let $E \in com(SF)$, and let

- $E_0 \in grd(SF)$,
- $E_1 = Ev(SF^{E_0}) \cap E$, and
- $E_2 \in grd((SF^{E_0})^{E_1})$.

We will show that (i) E_1 is conflict-free in SF^{E_0} and (ii) $E_0 \cup E_1 \cup E_2 = E$. For (i) assume the opposite is true, i.e. there is an $(T', h) \in R(SF^{E_0})$ with $T' \cup \{h\} \subseteq E_1$. That means there is an attack $(T, h) \in R$ with $T' \subseteq T$. But as $(T', h) \in R(SF^{E_0})$ we know $T \cap E_0^+ = \emptyset$, hence, $T \subseteq E_0 \cup E_1$, but then $E \notin \mathit{cf}(SF)$, a contradiction.

It remains to show (ii): $(\subseteq)$: $E_0 \subseteq E$ is immediate and $E_1 \subseteq E$ follows from the definition of $Ev(.)$. Let $SF' = (A', R') = SF^{E_0 \cup E_1}$ and let $G \in grd(SF')$. Towards contradiction assume there is some $a \in G \setminus E$. Let a be chosen such that $a \in \Gamma^{i+1}_{SF'}(\emptyset)$ and there is no $a' \in \Gamma^{j}_{SF'}(\emptyset)$ with $a' \notin E$ for all $j < i$ (i.e. a is the first such argumentwe encounter when constructing the grounded extension step by step). By Lemma 4.7 a is defended by $E_0 \cup E_1 \cup \Gamma^{i}_{SF'}(\emptyset)$ in SF. From the way we chose i we get $\Gamma^{i}_{SF'}(\emptyset) \subseteq E$, and consequently $E_0 \cup E_1 \cup \Gamma^{i}_{SF'}(\emptyset) \subseteq E$. As $\Gamma(.)$ is monotone, E defends a in SF. Moreover, $E \cup \{a\}$ is conflict-free in SF by Lemma 3.3, item 3. But this is a contradiction to our assumption $E \in com(SF)$.

$(\supseteq)$: By Lemma 4.9 we know $X = E \setminus (E_0 \cup E_1) \in com((SF^{E_0})^{E_1})$. By Lemma 4.8 we know that, as by construction of E_0 and E_1 the set X cannot contain arguments in even cycles, X is indeed the grounded extension of $(SF^{E_0})^{E_1}$, i.e. $X = E_2$. □

5 SETAFs and Logic Programs

The goal of this section is to establish a connection between atomic logic programs and SETAFs w.r.t. the following aspects:

- There shall be a translation between SETAFs and atomic LPs which preserves the semantics in a natural way,
- there shall be compatible notions of reducts for both SETAFs and atomic LPs.

We consider logic programs with default negation not. Such programs consist of rules of the form

$$c \leftarrow a_1, \ldots, a_n, \text{not } b_1, \ldots, \text{not } b_m. \qquad (1)$$

where $0 \leq n, m$ and the a_i, b_i, and c are ordinary atoms. Throughout this section we will consider atomic logic programs (Janhunen 2004), that is, $n = 0$. We let $head(r) = \{c\}$, $neg(r) = \{b_1, \ldots, b_m\}$, and $body(r) = \{a_1, \ldots, a_n, b_1, \ldots, b_m\}$. For $B = \{b_1, \ldots, b_m\}$ we use $c \leftarrow \text{not } B$. as a shorthand for such rules. We call rules with empty bodies *facts*, and write "c." instead of "$c \leftarrow .$". We let $\mathcal{L}(P)$ be the set of all atoms occurring in P.

Definition 5.1. A 3-valued Herbrand Interprtation I of a logic program P is a tuple $I = (T, F)$ with $T \cup F \subseteq \mathcal{L}(P)$ and $T \cap F = \emptyset$. We say $a \in \mathcal{L}(P)$ is true iff $a \in T$, false iff $a \in F$ and undefined otherwise.

In the following, we define the reduct P/I of an atomic program w.r.t. a 3-valued interpretation I. Note that our assumption $n = 0$ renders the definition of this reduct simpler than the usual one given in the literature.

Given an atomic logic program P with interpretation $I = (T, F)$ we define the reduct P/I of P w.r.t. I as follows: Starting from P, i) remove each rule r from P with $T \cap neg(r) \neq \emptyset$, ii) remove "not b" from each remaining rule whenever $b \in F$, and iii) replace each occurrence of "not b" from each remaining rule with u for a fresh atom u. By $\Psi_P(I) = (T_\Psi, F_\Psi)$ we denote the least 3-valued model of P/I, i.e. T_Ψ is minimal and F_Ψ maximal s.t.

- $a \in T_\Psi$ iff there is a fact "a." occurring in P/I.
- $a \in F_\Psi$ iff no rule with head a occurs in P/I.

We are now ready to define:

Definition 5.2. Let $I = (T, F)$ be a 3-valued interpretation of P. Then I is

- P-stable if $I = \Psi_P(I)$;
- T is well-founded if I is P-stable with minimal T,
- T is regular if I is P-stable with maximal T,
- T is stable if I is P-stable and $T \cup F = \mathcal{L}(p)$.

Example 5.3. Consider the following logic program P.

$$\begin{array}{lll} P: & a \leftarrow \text{not } b, \text{not } c. & b \leftarrow \text{not } a, \text{not } c. \\ & c \leftarrow \text{not } a, \text{not } b. & \\ & d \leftarrow \text{not } a. & e \leftarrow \text{not } a. \\ & d \leftarrow \text{not } b, \text{not } c. & e \leftarrow \text{not } d. \end{array}$$

Let us verify that we have four P-stable models $I_1 = (\emptyset, \emptyset)$, $I_2 = (\{a, d\}, \{b, c, e\})$, $I_2 = (\{b, d, e\}, \{a, c\})$, and $I_4 = (\{c, d, e\}, \{a, b\})$. We obtain

$$\begin{array}{llllll} P/I_1: & a \leftarrow u. & b \leftarrow u. & c \leftarrow u. & d \leftarrow u. & e \leftarrow u. \\ P/I_2: & a. & d. & & & \\ P/I_3: & b. & d. & e. & & \\ P/I_4: & c. & d. & e. & & \end{array}$$

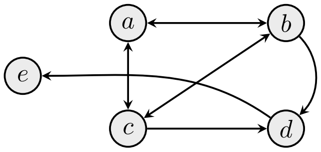
Indeed, we have $\Psi_P(I_i) = I_i$ for $i = 1, 2, 3, 4$. Thereby, I_1 is the well-founded one and I_2, I_3 and I_4 are both regular and stable.

There is a natural, well-known relation between atomic logic programs where each rule head is unique and Dung-style AFs (Caminada et al. 2015). In a nutshell, a rule of the form $c \leftarrow \text{not } b_1, \dots, \text{not } b_m.$ in a program P encodes that $b_1, \dots, b_m$ are the attackers of c. Vice versa, if c is an argument in an AF $F = (A, R)$ with $(b_1, c), \dots, (b_m, c)$ being the tuples in R of the form $(\cdot, c)$, then we get a rule $c \leftarrow \text{not } b_1, \dots, \text{not } b_m.$ in the corresponding logic program. This procedure translates stable models of the given program into stable extensions of the AF and vice versa.

Example 5.4. Consider a simplified version of the above example:

$$\begin{array}{lll} P': & a \leftarrow \text{not } b, \text{not } c. & b \leftarrow \text{not } a, \text{not } c. \\ & c \leftarrow \text{not } a, \text{not } b. & \\ & d \leftarrow \text{not } b, \text{not } c. & e \leftarrow \text{not } d. \end{array}$$

Observe that each rule head is unique. We note that the stable models of P' are $\{\{a, d\}, \{b, e\}, \{c, e\}\}$. Now consider the following AF F obtained by the above described construction:



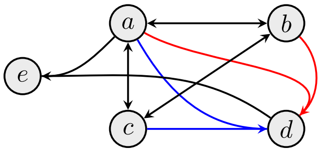
Indeed, $stb(F) = \{\{a, d\}, \{b, e\}, \{c, e\}\}$.

Let us examine why this construction only works for atomic logic programs with unique rule heads. Consider for example the two rules

$$d \leftarrow \text{not } a. \qquad d \leftarrow \text{not } b, \text{not } c.$$

In order to defeat d, it does not suffice to accept b or c anymore, because as long as a is not present, d can be accepted nonetheless. Hence, defeating d requires some collective attacks containing both a and either b or c. This cannot be straightforwardly encoded in usual AFs, but SETAFs naturally possess this expressive power. Let us demonstrate how our initial example program can be expressed as such.

Example 5.5. We observe that P models a choice between a, b and c. Moreover, d is *not* acceptable iff a and b or a and c are; e is *not* acceptable iff a and d is. We hence expect P to correspond to the following SETAF SF:



Indeed, we find

$$com(SF) = \{\emptyset, \{a, d\}, \{b, d, e\}, \{c, d, e\}\}$$

in accordance with the P-stable models of P.

Let us formalize this construction in general. First we require the notion of a hitting set.

Definition 5.6. Let $\mathcal{M}$ be a set of sets. We call $\mathcal{H}$ a *hitting set* of $\mathcal{M}$ if $\mathcal{H} \cap M \neq \emptyset$ for each $M \in \mathcal{M}$. A hitting set $\mathcal{H}$ of $\mathcal{M}$ is a *minimal hitting set* of $\mathcal{M}$ ($\mathcal{H} \in HS_{min}(\mathcal{M})$) if $\mathcal{H}' \subsetneq \mathcal{H}$ implies $\mathcal{H}'$ is not a hitting set of $\mathcal{M}$.

Now we are ready to define the actual translations between SETAFs and atomic logic programs:

Definition 5.7. Let P be an atomic logic program. For $c \in \mathcal{L}(P)$ we let $B_P(c) = \{body(r) \mid head(r) = c\}$. We define the corresponding SETAF $SF_P = (A_P, R_P)$ by letting

$$A_P = \{a \in \mathcal{L}(P) \mid a \in \textstyle\bigcup_{r \in P} head(r)\},$$
$$R_P = \{(T, c) \mid T \in HS_{min}(B_P(c))\}.$$

Definition 5.8. Let $SF = (A, R)$ be a SETAF. For $a \in A$ we let $tails_{SF}(a) = \{T \mid (T, a) \in R\}$. We define the corresponding logic program P_{SF} by letting

$$\begin{aligned} P_{SF} = &\{c \leftarrow \text{not } B. \mid B \in HS_{min}(tails_{SF}(c))\} \\ &\cup \{c. \mid tails_{SF}(c) = \emptyset\}. \end{aligned}$$

In order to infer a natural correspondence between SF_P and P_{SF} we need the notion of *redundancy-free* logic programs, similar in spirit to redundancy-free SETAFs as defined above.

Definition 5.9. A logic program P is *redundancy-free* if i) each $a \in \mathcal{L}(P)$ occurs in the head of some rule, ii) there are no two distinct rules $c \leftarrow \text{not } B_1.$ and $c \leftarrow \text{not } B_2.$ with $B_1 \subseteq B_2$.

For redundancy-free programs and SETAFs we obtain the following relation.

Lemma 5.10. *If SF is a redundancy-free SETAF, then $(SF_P)_{SF} = SF$. If P is a redundancy-free logic program then $(P_{SF})_P = P$.*

Proof. This follows from the following result from (Berge 1989): Let $X = \{X_1, \dots, X_n\}$ be a set of sets with $X_i \not\subseteq X_j$ for $i \neq j$. Then $HS_{min}(HS_{min}(X)) = X$. □

Let us demonstrate the above result using our running example:

Example 5.11. Consider again the above SETAF. We find

$$\begin{array}{ll} tails_{SF}(a) = \{\{b\}, \{c\}\} & tails_{SF}(b) = \{\{a\}, \{c\}\} \\ tails_{SF}(c) = \{\{a\}, \{b\}\} & \\ tails_{SF}(d) = \{\{a, b\}, \{a, c\}\} & tails_{SF}(e) = \{\{a, d\}\}. \end{array}$$

with hitting sets

$$\begin{array}{ll} \{\{b,c\}\} & \{\{a,c\}\} \\ \{\{a,b\}\} & \\ \{\{a\},\{b,c\}\} & \{\{a\},\{d\}\} \end{array}$$

which indeed correspond to the rules in P:

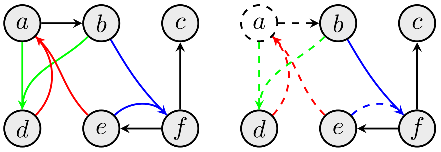
$$\begin{array}{lll} P: & a \leftarrow \text{not } b, \text{not } c. & b \leftarrow \text{not } a, \text{not } c. \\ & c \leftarrow \text{not } a, \text{not } b. & \\ & d \leftarrow \text{not } a. & e \leftarrow \text{not } a. \\ & d \leftarrow \text{not } b, \text{not } c. & e \leftarrow \text{not } d. \end{array}$$

Our next goal is to compare the reduct notions of SETAFs and LPs. In order to obtain two compatible concepts, we need to make some adjustments. Let us start by the SETAF reduct. A somewhat unnatural feature from the point of view of LPs is that the SETAF reduct SF^E removes the arguments in E although they are intuitively set to be true. This prevents a smooth correspondence between SF^E and any natural LP reduct notions, because one would not remove rules of an LP based on their rule heads.

Hence, let us now examine a slightly different version of the reduct for SETAFs in order to streamline the relation. To this end we simply define an E-reduct for SETAFs which preserves the arguments contained in E as follows: For a SETAF $SF = (A, R)$ and $E \subseteq A$ we let $Pre(SF, E) = (A', R')$ where

$$\begin{aligned} A' &= A \setminus E_R^+, \\ R' &= \{(T \setminus E, h) \mid (T, h) \in R, h \in A', \\ &\qquad\qquad T \setminus E \neq \emptyset, T \cap E_R^+ = \emptyset\}. \end{aligned}$$

Example 5.12. Consider the following SETAF SF. Constructing the reduct $Pre(SF, \{d, e\})$ yields removal of a, but d and e are still present.



(a) SETAF SF (b) $Pre(SF, \{d, e\})$

Observe in particular that the attack $(\{b, e\}, f)$ is reduced to (b, f) in the reduct since e is set to be true anyway.

A key strength of SF^E is that we are able to characterize SETAF semantics quite elegantly, as we formalized in Proposition 3.5. The same is however true for $Pre(SF, \cdot)$.

Proposition 5.13. *Let $SF = (A, R)$ be a SETAF and $E \in cf(SF)$.*

1. *$E \in stb(SF)$ iff $Pre(SF, E) = (E, \emptyset)$,*
2. *$E \in adm(SF)$ iff no $e \in E$ is attacked in $Pre(SF, E)$,*
3. *$E \in com(SF)$ iff $E = \Gamma_{Pre(SF,E)}(\emptyset)$,*
4. *$E \in pref(SF)$ iff $E = \Gamma_{Pre(SF,E)}(\emptyset)$ and in addition $com(Pre(SF, E)) = \{E\}$.*

Let us now turn to the reduct for LPs. Here, the main issue is the third step turning each occurrence of "not b" into u for a fresh atom u. This three-valued approach does not compare very well to the SETAF reduct notions. Thus, we will adjust this reduct as well.

Definition 5.14. Given a logic program P and $E \subseteq \mathcal{L}(P)$, we define the E-reduct P^E as follows: Starting from P, i) remove each rule r from P with $E \cap neg(r) \neq \emptyset$, ii) remove "not b" from each remaining rule whenever b does not occur in the head of any rule anymore, and iii) remove redundant rules.

Observe that the reduct P^E is closely related to the first two step when constructing P/E. Next we show that this reduct notion is also capable of characterizing P-stable models, similar in spirit to Proposition 5.13 for SETAFs.

Proposition 5.15. *Let P be a logic program and $I = (T, F)$ a 3-valued interpretation. Then I is a P-stable model iff*

- *$T = \{c \mid c. \in P^T\}$ and*
- *$F = \mathcal{L}(P) \setminus \mathcal{L}\left(P^T\right)$.*

Proof. ($\Rightarrow$) Let $I = (T, F)$ be a P-stable model. If $c \in T$, then $c.$ must be a fact in P/I. If $c \in F$, no rule r with $head(r) = \{c\}$ occurs in P/I. So let us now compute P^T. In the first step, rules r with $T \cap neg(r) \neq \emptyset$ are removed in both cases. Since no other rule is removed when constructing P/I, we must have that $c \in F$ iff c does not occur in the head of any rule anymore after this first step. Thus, in the second step "not b" gets removed whenever $b \in F$ in both P/I and P^T. Finally, in P^T redundant rules are removed; this does not change the facts or occurring atoms in the program. We thus obtain $T = \{c \mid c. \in P^T\}$ and $F = \mathcal{L}(P) \setminus \mathcal{L}\left(P^T\right)$.

($\Leftarrow$) Let $I = (T, F)$ with the two mentioned properties. Our reasoning is as above, yielding that T corresponds to the facts in P/I and F to those atoms which never occur in the head of any rule. So I is P-stable. □

Now we show the desired compatibility result for the reduct notions. It formalizes that moving in between the two frameworks and constructing the reducts can be done in any order. Before stating the actual result we require the following auxiliary lemma about properties of hitting sets.

Lemma 5.16. *Let $\mathcal{M}$ be a set of sets with $E \subseteq M$ for each $M \in \mathcal{M}$. Let $\mathcal{M} \setminus E := \{M \setminus E \mid M \in \mathcal{M}\}$. Then $\mathcal{S} \in HS_{min}(\mathcal{M} \setminus E)$ iff $\mathcal{S} \in HS_{min}(\mathcal{M})$ with $E \cap \mathcal{S} = \emptyset$.*

Equipped with this lemma we are now in a position to infer the desired compatibility result of the reduct notions.

Theorem 5.17.

1. *For a SETAF SF and $E \subseteq A$ we have $P_{Pre(SF,E)} = (P_{SF})^E$.*
2. *For an atomic program P and $E \subseteq \mathcal{L}(P)$ we have $SF_{P^E} = Pre(SF_P, E)$.*

Proof. We prove the first item only, since the other claim can be shown analogously. Let $SF = (A, R)$ be a SETAF. Set $Pre(SF, E) = (A', R')$.

($\subseteq$) Let $r \in P_{Pre(SF,E)}$. Assume r is of the form $r = c \leftarrow \text{not } B.$ for $B \neq \emptyset$. By construction, $c \in A'$ implying $T \setminus E \neq \emptyset$ for each $(T, c) \in R$. Hence by definition $B \in HS_{min}(\mathit{tails}_{Pre(SF,E)}(c))$ where

$$\mathit{tails}_{Pre(SF,E)}(c) = \{T \setminus E \mid (T, c) \in R, T \cap E_R^+ = \emptyset\}.$$

By Lemma 5.16,

$$B \in HS_{min}\left(\{T \mid (T, c) \in R, T \cap E_R^+ = \emptyset\}\right) \qquad (2)$$

satisfying $B \cap E = \emptyset$. By (2), $c \leftarrow \text{not } B.$ is a rule in P_{SF} and since $B \cap E = \emptyset$, it is a rule after the first step of constructing the reduct $(P_{SF})^E$. Moreover, each set $T' \in \mathit{tails}_{SF^E}(c)$ satisfies $T' \subseteq A'$ and thus, B must do so as well by minimality implying $r \in (P_{SF})^E$.

Now assume $r \in P_{SF^E}$ is a fact with head c. Then $\mathit{tails}_{Pre(SF,E)}(c) = \emptyset$, i.e. c is an unattacked argument in $Pre(SF, E)$. If c does not possess any attackers, it occurs in $(P_{SF})^E$ by definition. Hence suppose c possesses attackers in SF. By construction of $Pre(SF, E)$ we have $T \cap E_R^+ \neq \emptyset$ for each $T \in \mathit{tails}_{SF}(c)$. Now consider P_{SF}. Each rule $c \leftarrow \text{not } B.$ in P_{SF} is s.t. B is a minimal hitting set of $\mathit{tails}_{SF}(c)$, thus at least one of them satisfies $B \subseteq E_R^+$. Thus, we found the corresponding fact $c.$ if we can ensure that the atoms occurring in E_R^+ do not occur in the head of any rule in $(P_{SF})^E$. To this end let $e \in E_R^+$ and $e \leftarrow \text{not } B_e.$ a rule in P_{SF}, i.e. B_e is a hitting set of $\mathit{tails}_{SF}(e)$. By definition $e \in E_R^+$ means $T \subseteq E$ for at least one $T \in \mathit{tails}_{SF}(e)$. Hence $B_e \cap T \neq \emptyset$ and thus, $e \leftarrow \text{not } B_e.$ is removed when constructing $(P_{SF})^E$.

($\supseteq$) Consider a rule $c \leftarrow \text{not } B.$ occurring in $(P_{SF})^E$. We show that $c \notin E_R^+$ and hence $c \in A'$. Indeed, suppose $(T, c) \in R$ for some $T \subseteq E$ and consider a rule $c \leftarrow \text{not } B_c.$ in P_{SF}. Since B_c is a hitting set of $\mathit{tails}(c)$, $B_c \cap E \neq \emptyset$ and hence, the rule gets removed when constructing $(P_{SF})^E$. Since our rule was arbitrary, this is a contradiction. Having established $c \in A'$, our rule $c \leftarrow \text{not } B.$ is s.t.

$$B \in HS_{min}\left(\{T \mid (T, c) \in R, T \cap E_R^+ = \emptyset\}\right)$$

satisfying $B \cap E = \emptyset$ and hence by Lemma 5.16, B is a minimal hitting set of

$$\mathit{tails}_{SF^E}(c) = \{T \setminus E \mid (T, c) \in R, T \cap E_R^+ = \emptyset\}.$$

Hence, $c \leftarrow \text{not } B.$ is a rule in P_{SF^E}.

Now let $c.$ be a fact in $(P_{SF})^E$. As above, $c \in A'$. We infer that c is unattacked in SF^E and hence our fact occurs in P_{SF^E} as well. □

Due to the reduct characterizations of SETAFs and LPs as well as their relation, we may characterize complete extensions $E \in \mathit{com}(SF)$ in terms of P_{SF} and vice versa, P-stable models in terms of SF_P.

Proposition 5.18. *Let P be a logic program, $I = (T, F)$ a 3-valued interpretation, SF be a SETAF and $E \subseteq A$.*

1. *I is a P-stable model iff $T \in \mathit{com}(SF_P)$ and $F = T^+_{SF_P}$.*
2. *$E \in \mathit{com}(SF)$ iff there exists a P-stable model (E, F') of P_{SF}.*

Proof. 1) By Proposition 5.15 we have that I is a P-stable model iff $T = \{c \mid c. \in P^T\}$ and $F = \mathcal{L}(P) \setminus \mathcal{L}\left(P^T\right)$. By construction of SF_{P^T} the latter is equivalent to $T = \Gamma_{SF_{P^T}}(\emptyset)$ and $F = L(P) \setminus A(SF_{P^T})$, which by Theorem 5.17 is equivalent to $T = \Gamma_{Pre(SF_P,T)}(\emptyset)$ and $F = L(P) \setminus A(Pre(SF_P, T))$. By Proposition 5.13 and the definition of $Pre(SF_P, T)$ the last statement is equivalent to $T \in \mathit{com}(SF_P)$ and $F = T^+_{SF_P}$.

2) By Proposition 5.13 we have that $E \in \mathit{com}(SF)$ iff $T = \Gamma_{Pre(SF,T)}(\emptyset)$. By construction of $P_{Pre(SF,E)}\}$ the latter is equivalent to $E = \{c \mid c. \in P_{Pre(SF,E)}\}$ which by Theorem 5.17 is equivalent to $E = \{c \mid c. \in (P_{SF})^E\}$. By setting $F' = \mathcal{L}(P) \setminus \mathcal{L}\left(P^T\right)$ and using Proposition 5.15 we obtain that $E = \{c \mid c. \in (P_{SF})^E\}$ iff (T, F) is a P-stable model of P_{SF}. □

Notice that the above correspondence between P-stable and complete semantics directly extends to the other semantics under our considerations.

6 Conclusion

In this paper we introduced an E-reduct for SETAFs which served as a basis for (a) modularization results for SETAF semantics, (b) alternative characterizations of the semantics and (c) explanation schemes for complete extensions. These results demonstrate that our definition of the E-reduct is a proper generalization of the corresponding notion for AFs. Moreover, we investigated the relation between SETAFs and atomic logic programs. Our results relate the E-reduct of SETAFs to a corresponding reduct of logic programs and further provide an equivalence between argumentation and logic programming semantics.

The E-reduct for AFs has successfully been used to define a new family of semantics based on the notion of *weak admissibility* (Baumann, Brewka, and Ulbricht 2020b). These semantics address problems with the standard semantics that have already been pointed out by Dung in his original proposal. Our E-reduct for SETAFs already paves the way to generalize the notion of weak admissibility also to SETAFs. Investigating the corresponding semantics for SETAFs is an interesting direction for future work.

Acknowledgments This research has been supported by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, and by the Austrian Science Fund (FWF) through projects P30168, P32830, and Y698.

References

Baroni, P.; Cerutti, F.; Giacomin, M.; and Guida, G. 2011. AFRA: argumentation framework with recursive attacks. *Int. J. Approx. Reason.* 52(1):19–37.

Baroni, P.; Boella, G.; Cerutti, F.; Giacomin, M.; van der Torre, L. W. N.; and Villata, S. 2014. On the input/output behavior of argumentation frameworks. *Artif. Intell.* 217:144–197.

Baroni, P.; Giacomin, M.; and Guida, G. 2005. SCC-recursiveness: a general schema for argumentation semantics. *Artif. Intell.* 168(1-2):162–210.

Baumann, R., and Ulbricht, M. 2021. Choices and their consequences - explaining acceptable sets in abstract argumentation frameworks. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021*. To appear.

Baumann, R.; Brewka, G.; and Ulbricht, M. 2020a. Comparing weak admissibility semantics to their dung-style counterparts - reduct, modularization, and strong equivalence in abstract argumentation. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, 79–88.

Baumann, R.; Brewka, G.; and Ulbricht, M. 2020b. Revisiting the foundations of abstract argumentation - semantics based on weak admissibility and weak defense. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 2742–2749. AAAI Press.

Baumann, R. 2011. Splitting an argumentation framework. In Delgrande, J. P., and Faber, W., eds., *Logic Programming and Nonmonotonic Reasoning - 11th International Conference, LPNMR 2011. Proceedings*, volume 6645 of *Lecture Notes in Computer Science*, 40–53. Springer.

Berge, C. 1989. *Hypergraphs - combinatorics of finite sets*, volume 45 of *North-Holland mathematical library*. North-Holland.

Bikakis, A.; Cohen, A.; Dvořák, W.; Flouris, G.; and Parsons, S. 2021. Joint attacks and accrual in argumentation frameworks. *FLAP* 8(6):1437–1501.

Caminada, M.; Sá, S.; Alcântara, J.; and Dvořák, W. 2015. On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reasoning* 58:87–111.

Cayrol, C., and Lagasquie-Schiex, M. 2005. On the acceptability of arguments in bipolar argumentation frameworks. In Godo, L., ed., *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU 2005, Proceedings*, volume 3571 of *Lecture Notes in Computer Science*, 378–389. Springer.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.

Dvořák, W.; Fandinno, J.; and Woltran, S. 2019. On the expressive power of collective attacks. *Argument Comput.* 10(2):191–230.

Dvořák, W.; Greßler, A.; and Woltran, S. 2018. Evaluating SETAFs via answer-set programming. In Thimm, M.; Cerutti, F.; and Vallati, M., eds., *Proceedings of the Second International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2018)*, volume 2171 of *CEUR Workshop Proceedings*, 10–21. CEUR-WS.org.

Dvořák, W.; König, M.; and Woltran, S. 2021a. Graph-classes of argumentation frameworks with collective attacks. In Faber, W.; Friedrich, G.; Gebser, M.; and Morak, M., eds., *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, Proceedings*, volume 12678 of *Lecture Notes in Computer Science*, 3–17. Springer.

Dvořák, W.; König, M.; and Woltran, S. 2021b. On the complexity of preferred semantics in argumentation frameworks with bounded cycle length. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021*. To appear.

Dvořák, W.; Rapberger, A.; and Woltran, S. 2020a. Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to SETAFs. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, 341–350.

Dvořák, W.; Rapberger, A.; and Woltran, S. 2020b. On the different types of collective attacks in abstract argumentation: equivalence results for SETAFs. *J. Log. Comput.* 30(5):1063–1107.

Dvořák, W. 2012. *Computational Aspects of Abstract Argumentation*. Ph.D. Dissertation, Vienna University of Technology, Institute of Information Systems.

Dvořák, W., and Dunne, P. E. 2017. Computational problems in formal argumentation and their complexity. *FLAP* 4(8):2557–2622.

Flouris, G., and Bikakis, A. 2019. A comprehensive study of argumentation frameworks with sets of attacking arguments. *Int. J. Approx. Reason.* 109:55–86.

Janhunen, T. 2004. Representing normal programs with clauses. In de Mántaras, R. L., and Saitta, L., eds., *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004*, 358–362. IOS Press.

Nielsen, S. H., and Parsons, S. 2006. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In Maudet, N.; Parsons, S.; and Rahwan, I., eds., *Argumentation in Multi-Agent Systems, Third International Workshop, ArgMAS 2006, Revised Selected and Invited Papers*, volume 4766 of *Lecture Notes in Computer Science*, 54–73. Springer.

Polberg, S. 2017. *Developing the Abstract Dialectical Framework*. Ph.D. Dissertation, Vienna University of Technology, Institute of Information Systems.

Thimm, M. 2012. A probabilistic semantics for abstract argumentation. In Raedt, L. D.; Bessiere, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P. J. F., eds., *Proceedings of the 20th European Conference on Artificial Intelligence, ECAI 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 750–755. IOS Press.

A Incorporating Different Cycle Notions

In Section 4 we introduce explanation schemes via the notion of even-primal-cycles. In (Dvořák, König, and Woltran 2021b) other cycle notions for SETAFs are introduced, namely set-cycles and incidence-cycles. In the following we will show that the same results hold for these cycle notions and choosing incidence-cycles rather than primal-cycles might even give an advantage. As a side product we pinpoint the complexity for reasoning in SETAFs that have no (even-/odd-length) cycles of these kinds.

To this end we assume the reader to have a basic understanding of complexity results in the context of formal argumentation; for a gentle introduction see e.g. (Dvořák and Dunne 2017). In particular, we will refer to the complexity of *credulous and skeptical reasoning*, i.e. given a SETAF $SF = (A, R)$, deciding whether an argument $a \in A$ is in one/all extension(s) $\sigma(SF)$ for some semantics σ. For the semantics under our consideration, these tasks are in general on hard for up to the second level of the polynomial hierarchy, yet for odd-primal-cycle-free SETAFs the complexity drops to the first level of the polynomial hierarchy and for even-primal-cycle-free SETAFs they become tractable.

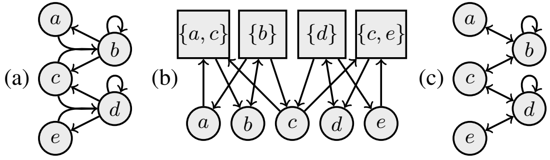
In the incidence graph, the SETAF (A, R) is represented as a bipartite directed graph, with the arguments A as one part, and the tails of the attacks in R as the other. We add an edge from every argument to the tails it appears in, and edges from every tail to arguments *attacked by* it.

Definition A.1. For a SETAF $SF = (A, R)$ let $\mathit{tails}(SF) = \{T \mid (T, h) \in R\}$.Then $\mathit{Inc}(SF) = (V, E)$ with $V = A \cup \mathit{tails}(SF)$ and $E = \{(t, T), (T, h) \mid (T, h) \in R, t \in T\}$ is its *incidence graph*.

Definition A.2. Let $SF = (A, R)$ be a SETAF. A *cycle* C of length $|C| = n$ is a directed cycle $C = (T_1, a_1, T_2, a_2, \dots, a_n, T_1)$ in $\mathit{Inc}(SF)$. We say C is (i) an *incidence-cycle* if all a_i and all T_i are distinct; (ii) a *primal-cycle* if all a_i are distinct; and (iii) a *set-cycle* if all T_i are distinct.

As the name suggests, a primal-cycle corresponds to a cycle in the *primal graph* (Dvořák, König, and Woltran 2021a), a representation of a SETAF as a directed graph. For a SETAF $SF = (A, R)$, its primal graph $\mathit{Primal}(SF)$ is the directed graph with A as its vertices, and an edge between two vertices a and b iff a is part of an attack towards b in R (see Example A.3). Every incidence-cycle is a primal-cycle and a set-cycle. Note that on AFs all of these cycle notions coincide with 'classical' directed, non-repeating cycles.

Example A.3. (a) SF, (b) $\mathit{Inc}(SF)$, and (c) $\mathit{Primal}(SF)$.



The following statement will be the basis for our results on (even-/odd-)(incidence-/primal-/set-)cycle-free SETAFs.

Lemma A.4. *Let SF be a SETAF. Every cycle C of length k in SF can be divided into m incidence-cycles $C_1, C_2, \dots, C_m$ such that $\sum_{i=1}^{m} |C_i| = |C|$.*

Note that in this "division" we do not form new cycles by using "shortcuts", i.e. edges that are not yet used in C. Instead we just split the cycle such that no argument and no tail appears more than once in each sub-cycle, hence not changing the overall length in the incidence graph. From this and that every incidence-cycle is a primal-cycle and set-cycle, the following immediately follows:

Proposition A.5. *A SETAF SF is incidence-cycle-free iff SF is primal-cycle-free iff SF is set-cycle-free.*

Corollary A.6. *The complexity results for primal-cycle-free SETAFs*[1] *apply for incidence-cycle-free SETAFs and set-cycle-free SETAFs.*

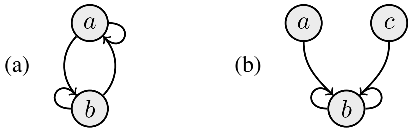
Also every odd-primal-cycle and every odd-set-cycle contains an odd-incidence-cycle, which is in turn an odd-primal-cycle and an odd-set-cycle. Hence, the next results follows:

Proposition A.7. *A SETAF SF is odd-incidence-cycle-free iff SF is odd-primal-cycle-free iff SF is odd-set-cycle-free.*

Corollary A.8. *The complexity results for odd-primal-cycle-free SETAFs*[1] *apply for odd-incidence-cycle-free SETAFs and odd-set-cycle-free SETAFs.*

For even-cycle-free SETAFs the equivalent of Proposition A.5 does not hold. As every incidence-cycle is a primal-cycle and a set-cycle, even-incidence-cycle-freeness implies even-primal-cycle-freeness and even-set-cycle-freeness, but no other direction of these implications holds, as the following counter-examples show:

Example A.9. (a) is even-set-cycle-free, even-incidence-cycle-free, but *not* even-primal-cycle- free; (b) is even-primal-cycle-free, even-incidence-cycle-free, but *not* even-set-cycle-free.



However, for our purposes each of our cycle-notions is sufficient, as the next lemma illustrates. This is a generalization of the corresponding result on AFs from (Dvořák 2012) and was already stated in this form in (Dvořák, König, and Woltran 2021b), and for primal-cycles in (Dvořák, König, and Woltran 2021a).

Lemma A.10. *Every SETAF SF with $|\mathit{com}(SF)| \geq 2$ has an even-incidence-cycle (and, hence, an even-primal-cycle and an even-set-cycle).*

Again, we use the fact that every incidence-cycle is a primal-cycle and a set-cycle.

Corollary A.11. *The complexity results for even-primal-cycle-free SETAFs*[1] *apply for even-incidence-cycle-free SETAFs and even-set-cycle-free SETAFs.*

Finally, note that our choice of primal-cycles for the definition of explanation schemes in Section 4 was arbitrary: all results leading up and including Theorem 4.10 also hold if the definition of $\mathit{Ev}(\cdot)$ is changed to Ev^s and Ev^i, that we define to yield the arguments in even length set- and incidence-cycles, respectively. With this in mind, we define (successful) incidence-explanation schemes and (successful) set-explanation schemes analogous to Definition 4.2 and Definition 4.5. In fact, as $\mathit{Ev}^i(SF) \subseteq \mathit{Ev}(SF)$ and $\mathit{Ev}^i(SF) \subseteq \mathit{Ev}^s(SF)$, incidence-explanation schemes require us to guess potentially less arguments than the primal-cycle and set-cycle counterparts. Then we can obtain the following result in the same manner as Theorem 4.10:

Theorem A.12. *A set $E \subseteq A$ is in $\mathit{com}(SF)$ iff E can be decomposed into a successful incidence-/set-explanation scheme.*

[1] See (Dvořák, König, and Woltran 2021b).

Computational Complexity of Strong Admissibility for Abstract Dialectical Frameworks

Atefeh Keshavarzi Zafarghandi[1] , **Wolfgang Dvořák**[2] , **Rineke Verbrugge**[1] , **Bart Verheij**[1]

[1]Department of Artificial Intelligence, Bernoulli Institute,
University of Groningen, The Netherlands
[2]Institute of Logic and Computation, TU Wien, Austria

Abstract

Abstract dialectical frameworks (ADFs) have been introduced as a formalism for modeling and evaluating argumentation allowing general logical satisfaction conditions. Different criteria used to settle the acceptance of arguments are called semantics. Semantics of ADFs have so far mainly been defined based on the concept of admissibility. Recently, the notion of strong admissibility has been introduced for ADFs. In the current work we study the computational complexity of the following reasoning tasks under strong admissibility semantics. We address 1. the credulous/skeptical decision problem; 2. the verification problem; 3. the strong justification problem; and 4. the problem of finding a smallest witness of strong justification of a queried argument.

1 Introduction

Interest and attention in argumentation theory has been increasing among artificial intelligence researchers (Bench-Capon and Dunne 2007). Applications of argumentation theory are based on a variety of argumentation formalisms and methods of evaluating arguments (Atkinson et al. 2017; Baroni et al. 2018; van Eemeren et al. 2014). Dung's abstract argumentation frameworks (Dung 1995) (AFs for short) have received notable attention, also thanks to their simple syntax that can model and evaluate a number of non-monotonic reasoning tasks. Semantics of AFs single out coherent subsets of arguments that fit together, according to specific criteria (Baroni, Caminada, and Giacomin 2011).

AFs model individual attack relations among arguments. Abstract dialectical frameworks (ADFs) are expressive generalizations of AFs in which the logical relations among arguments can be represented. ADFs were first introduced in (Brewka and Woltran 2010), and were further refined in (Brewka et al. 2013; Brewka et al. 2017; Brewka et al. 2018).

Often a new semantics is a refinement of an already existing one by introducing further restrictions on the set of accepted arguments or possible attackers. One of the main types of semantics of AFs is the grounded semantics. Its characteristics include that 1. each AF has a unique grounded extension; 2. the grounded extension collects all the arguments about which no one doubts their acceptance; 3. the grounded extension is often a subset of the set of extensions of other types of AF semantics. Thus, it is important to investigate whether an argument belongs to the grounded extension of a given AF. The notion of strong admissibility is introduced for AFs to answer the query 'Why does an argument belong to the grounded extension?'.

While the grounded extension collects all the arguments of a given AF that can be accepted without any doubt, a strongly admissible extension provides a (minimal) justification why specific arguments can be accepted without any doubt, i.e. belong to the grounded extension. Thus, the strong admissibility semantics can be the basis for an algorithm that can be used not only for answering the credulous decision problem but also for human-machine interaction that requires an explainable outcome (cf. (Caminada and Uebis 2020; Booth, Caminada, and Marshall 2018)).

In AFs, the concept of strong admissibility semantics has first been defined in the work of Baroni and Giacomin (2007), and later in (Caminada 2014). Furthermore, in (2019), Caminada and Dunne presented a labelling account of strong admissibility to answer the decision problems of AFs under grounded semantics. Moreover, Caminada showed in (2018; 2014) that strong admissibility plays a role in discussion games for AFs under grounded semantics. In addition, the computational complexity of strong admissibility of AFs has been analyzed (Caminada and Dunne 2020; Dvořák and Wallner 2020).

Because of the specific structure of ADFs, the definition of strong admissibility semantics of AFs cannot be directly reused in ADFs. Thus the concept of strong admissibility for ADFs has been introduced (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021a). This concept fulfils properties that are related to those of the strong admissibility semantics for AFs, as follows:

1. Strong admissibility is defined in terms of strongly justified arguments. 2. Strongly justified arguments are recursively reconstructed from their strongly justified parents. 3. Each ADF has at least one strongly admissible interpretation. 4. The set of strongly admissible interpretations of ADFs forms a lattice with as least element the trivial interpretation and as maximum element the grounded interpretation. 5. The strong admissibility semantics can be used to answer whether an argument is justifiable under grounded semantics. 6. The strong admissibility semantics of ADFs is different from the admissible, conflict-free, complete and grounded semantics of ADFs. 7. The strong admissibility

semantics for ADFs is a proper generalization of the strong admissibility semantics for AFs.

Whereas several fundamental properties of strong admissibility semantics for ADFs have been established, the computational complexity under strong admissibility semantics has not been studied. This work closes this gap by studying the complexity of the central reasoning tasks under the strong admissibility semantics of ADFs, as follows. 1. The credulous decision problem, i.e., whether there exists a strongly admissible interpretation that satisfies the queried argument, is coNP-complete. 2. The skeptical decision problem, i.e., whether all strongly admissible interpretations satisfy a queried argument, is trivial. 3. The verification problem, i.e., whether a given interpretation is a strongly admissible interpretation of an ADF, is coNP-complete. 4. The strong justification problem for an argument in an interpretation, i.e., whether an argument is strongly justified in an interpretation, is coNP-complete. 5. The problem of finding a small witness of strong justification of an argument, i.e, whether there exists a strongly admissible interpretation that satisfies a queried argument and is smaller than a given bound, is Σ_2^{P}-complete.

2 Formal Background

We recall the basics of AFs (Dung 1995) and ADFs (Brewka et al. 2018). Also we recall the definition of strong admissibility for ADFs and an associated algorithm, presented in (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b).

2.1 Abstract Argumentation Frameworks

We start the preliminaries to our work by recalling the basic notion of Dung's abstract argumentation frameworks (AFs). Subsequently, we present the extension form of strong admissibility semantics of AFs (Baroni and Giacomin 2007).

Definition 1. (Dung 1995) An abstract argumentation framework (AF) is a pair (A, R) in which A is a set of arguments and $R \subseteq A \times A$ is a binary relation representing attacks among arguments.

Let $F = (A, R)$ be a an AF. For each $a, b \in A$, the relation $(a, b) \in R$ is used to represent that a is an argument attacking the argument b. An argument $a \in A$ is, on the other hand, defended by a set $S \subseteq A$ of arguments (alternatively, the argument is acceptable with respect to S) (in F) if for each argument $c \in A$, it holds that if $(c, a) \in R$ then there is an $s \in S$ such that $(s, c) \in R$ (s is called a defender of a).

Different semantics of AFs present which sets of arguments in an AF can be jointly accepted (see the overview (Baroni, Caminada, and Giacomin 2011)). Set $S \subseteq A$ is called a *conflict-free* extension (in F) if there are no $a, b \in S$ s.t. $(a, b) \in R$. The *characteristic function* $F : 2^A \mapsto 2^A$ is defined as $F(S) = \{a \mid a \text{ is defended by } S\}$. Set $S \subseteq A$ is called an *admissible* extension (in F) if $S \subseteq F(S)$. Further, set $S \subseteq A$ is a *grounded* extension of an AF if S is the $\subseteq$-least fixed point of F.

Definition 2. (Baroni and Giacomin 2007) Given an argumentation framework $F = (A, R)$, $a \in A$ and $S \subseteq A$, it is said that a is *strongly defended* by S if and only if each attacker $c \in A$ of a is attacked by some $s \in S \setminus \{a\}$ such that s is strongly defended by $S \setminus \{a\}$.

Example 1. Let $F = (\{a, b, c\}, \{(a, b), (b, c)\})$ be an AF. Argument a is strongly defended by $S = \emptyset$, since a is not attacked by any argument. Also, argument c is strongly defended by set $S = \{a, c\}$, since the attacker of c, namely b is attacked by $a \in S \setminus \{c\}$ and a itself is strongly defended.

Definition 3. Given an AF (A, R) and set $S \subseteq A$, it is said that S is a *strongly admissible* extension of S if every $s \in S$ is strongly defended by S.

In Example 1, sets $S_1 = \emptyset$, $S_2 = \{a\}$, and $S_3 = \{a, c\}$ are strongly admissible extensions of F; all of them are subsets of the grounded extension of F. However, set $S' = \{c\}$ is not a strongly admissible extension of F, since $c \in S'$ is not strongly defended by $S' \setminus \{c\}$. Because argument c is attacked by b, however, no argument in $S' \setminus \{c\}$ attacks b.

2.2 Abstract Dialectical Frameworks

We summarize key concepts of abstract dialectical frameworks (Brewka and Woltran 2010; Brewka et al. 2018).

Definition 4. An abstract dialectical framework (ADF) is a tuple $D = (A, L, C)$ where:

- A is a finite set of arguments (statements, positions);
- $L \subseteq A \times A$ is a set of links among arguments;
- $C = \{\varphi_a\}_{a \in A}$ is a collection of propositional formulas over arguments, called acceptance conditions.

An ADF can be represented by a graph in which nodes indicate arguments and links show the relation among arguments. Each argument a in an ADF is labelled by a propositional formula, called acceptance condition, φ_a over $par(a)$ such that, $par(a) = \{b \mid (b, a) \in L\}$. The acceptance condition of each argument clarifies under which condition the argument can be accepted. An argument a is called an *initial argument* if $par(a) = \{\}$.

A *three-valued interpretation* v (for D) is a function $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, that maps arguments to one of the three truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$). Interpretation v is called *trivial*, and v is denoted by $v_{\mathbf{u}}$, if $v(a) = \mathbf{u}$ for each $a \in A$. Further, v is called a two-valued interpretation if for each $a \in A$ either $v(a) = \mathbf{t}$ or $v(a) = \mathbf{f}$.

Truth values can be ordered via the information ordering relation $<_i$ given by $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair of truth values are related by $<_i$. Relation $\leq_i$ is the reflexive closure of $<_i$. The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ is a complete meet-semilattice with the meet operator $\sqcap_i$, such that $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and returns $\mathbf{u}$ otherwise. The meet of two interpretations v and w is then defined as $(v \sqcap_i w)(a) = v(a) \sqcap_i w(a)$ for all $a \in A$.

It is said that an interpretation v is an *extension* of another interpretation w, if $w(a) \leq_i v(a)$ for each $a \in A$, denoted by $w \leq_i v$. Further, if $v \leq_i w$ and $w \leq_i v$, then v and w are equivalent, denoted by $v \sim_i w$.

For reasons of brevity, we will shorten the notion of three-valued interpretation $v = \{a_1 \mapsto t_1, \ldots, a_m \mapsto t_m\}$ with arguments $a_1, \ldots, a_m$ and truth values $t_1, \ldots, t_m$ as follows: $v = \{a_i \mid v(a_i) = \mathbf{t}\} \cup \{\neg a_i \mid v(a_i) = \mathbf{f}\}$. For instance, $v = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}\} = \{\neg a, b\}$.

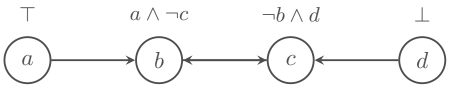


Figure 1: ADF of Examples 2 and 3

Given an interpretation v (for D), the partial valuation of φ_a by v is $v(\varphi_a) = \varphi_a^v = \varphi_a[b/\top : v(b) = \mathbf{t}][b/\bot : v(b) = \mathbf{f}]$, for $b \in par(a)$. Semantics for ADFs can be defined via the *characteristic operator* Γ_D, presented in Definition 5.

Definition 5. Let D be an ADF and let v be an interpretation of D. Applying Γ_D on v leads to v' such that for each $a \in A$, v' is as follows:

$$v'(a) = \begin{cases} \mathbf{t} & \text{if } \varphi_a^v \text{ is irrefutable (i.e., } \varphi_a^v \text{ is a tautology)} , \\ \mathbf{f} & \text{if } \varphi_a^v \text{ is unsatisfiable} , \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

Most types of semantics for ADFs are based on the concept of admissibility. An interpretation v for a given ADF F is called *admissible* iff $v \leq_i \Gamma_F(v)$; it is *preferred* iff v is $\leq_i$-maximal admissible; it the *grounded* interpretation of D iff v is the least fixed point of Γ_D. The set of all σ interpretations for an ADF D is denoted by $\sigma(D)$, where $\sigma \in \{adm, grd, prf\}$ abbreviates the different semantics in the obvious manner.

Example 2. An example of an ADF $D = (S, L, C)$ is shown in Figure 1. To each argument a propositional formula is associated, namely, the acceptance condition of the argument. For instance, the acceptance condition of c, namely $\varphi_c : \neg b \wedge d$, states that c can be accepted in an interpretation in which b is denied and d is accepted.

The interpretation $v_1 = \{a, \neg c, \neg d\}$ is an admissible interpretation, since $\Gamma_D(v_1) = \{a, b, \neg c, \neg d\}$ and $v_1 \leq_i \Gamma_D(v_1)$. Furthermore, $v_2 = \{a, b, \neg c, \neg d\}$ is a unique grounded interpretation and a preferred interpretation in D.

The notions of an argument being acceptable or deniable in an interpretation are defined as follows.

Definition 6. Let $D = (A, L, C)$ be an ADF and let v be an interpretation of D.

- An argument $a \in A$ is called *acceptable* with respect to v if φ_a^v is irrefutable.
- An argument $a \in A$ is called *deniable* with respect to v if φ_a^v is unsatisfiable.

We say that an argument is justified with respect to v if it is either acceptable or deniable with respect to v.

We redefine two decision problems of ADFs in Definition 7.

Definition 7. Let $D = (A, L, C)$ be an ADF, let σ be semantics of ADFs, i.e., $\sigma \in \{adm, prf, grd, cf\}$, and let a be an argument of A.

- a is *credulously acceptable (deniable)* under σ if there exists an interpretation v with $v \in \sigma(D)$ in which $v(a) = \mathbf{t}$ ($v(a) = \mathbf{f}$, respectively), denoted by $Cred_\sigma$.
- a is *skeptically acceptable (deniable)* under σ if for each v with $v \in \sigma(D)$ it holds that $v(a) = \mathbf{t}$ ($v(a) = \mathbf{f}$, respectively).

2.3 The Strong Admissibility Semantics for ADFs

In this section, we rephrase the concept of strong admissibility semantics for ADFs from (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021a), which is defined based on the notion of strongly justifiable arguments (i.e., strongly acceptable/deniable arguments). Below, the interpretation $v_{|P}$ is equal to $v(p)$ for any $p \in P$, and assigns all arguments that do not belong to P to $\mathbf{u}$, i.e., $v_{|P} = v_{\mathbf{u}}|^{p \in P}_{v(p)}$.

Definition 8. Let $D = (A, L, C)$ be an ADF and let v be an interpretation of D. Argument a is a *strongly justified* argument in interpretation v with respect to set E if one of the following conditions holds:

- $v(a) = \mathbf{t}$ and there exists a subset of parents of a excluding E, namely $P \subseteq par(a) \setminus E$ such that (a) a is acceptable with respect to $v_{|P}$ and (b) all $p \in P$ are strongly justified in v with respect to set $E \cup \{p\}$.
- $v(a) = \mathbf{f}$ and there exists a subset of parents of a excluding E, namely $P \subseteq par(a) \setminus E$ such that (a) a is deniable with respect to $v_{|P}$ and (b) all $p \in P$ are strongly justified in v with respect to set $E \cup \{p\}$.

An argument a is *strongly acceptable*, resp. *strongly deniable*, in v if $v(a) = \mathbf{t}$, resp. $v(a) = \mathbf{f}$, and a is strongly justified in v with respect to set $\{a\}$. We further say that an argument is *strongly justified* in v if it is either strongly acceptable or deniable in v.

Note that in Definition 8, the set of parents of a can be the empty set, i.e., $P = \emptyset$. If the set of parents of an argument, is empty, then $v_{|P} = v_{\mathbf{u}}$. In this case, a is strongly acceptable/deniable in v if $\varphi_a^{v_{\mathbf{u}}}$ is irrefutable/unsatisfiable, respectively. We say that *a is not strongly justified in an interpretation v* if there is no such a set of parents of a that satisfies the conditions of Definition 8 for a. The notion of strongly justified arguments in a given interpretation is presented in Example 3.

Example 3. Let $D = (\{a, b, c, d\}, \{\varphi_a : \top, \varphi_b : a \wedge \neg c, \varphi_c : \neg b \wedge d, \varphi_d : \bot\})$ be the ADF depicted in Figure 1. Let $v = \{b, \neg c, \neg d\}$. We show that c and d are strongly justified in v and b is not strongly justified in v. Since $v(c) = v(d) = \mathbf{f}$, we show that c and d are strongly deniable in v. First, since $\varphi_d^{v_{\mathbf{u}}} \equiv \bot$, it holds that d is strongly deniable in v.

We show that c is strongly deniable in v with respect to $E = \{c\}$. we choose the subset of parents of c excluding c equal to $P = \{d\}$. It is easy to check that $\varphi_c^{v_{|P}}$ is unsatisfiable, i.e., $\varphi_c^{v_{|P}} \equiv \varphi_c^{v_{|d}} \equiv \bot$. That is, c is deniable w.r.t. $v_{|d}$. Then, since $d \in P$, $v(d) = \mathbf{f}$ and d is strongly justified in v with respect to $E = \{c, d\}$, c is strongly deniable in v.

To show that b is not strongly justified in v, since $v(b) = \mathbf{t}$, we show that b is not strongly acceptable in v. Toward a contradiction, assume that b is strongly acceptable in v. Thus, we have to choose a set of parents of b, namely P that satisfies $\varphi_b^{v_{|P}} \equiv \top$. Let $P = par(b)$. Since $\varphi_b^{v_{|P}} \not\equiv \top$, there is not subset of $par(b)$ that satisfies the conditions of Definition 8 for b. Thus, b is not strongly acceptable in v.

In Example 3, if we choose a set of parents of c equal to $\{b\}$, then we cannot show that c is strongly deniable in interpretation v. The reason is that b is not strongly justified in v,

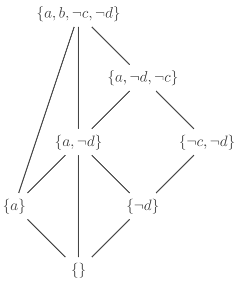


Figure 2: Complete lattice of the strongly admissible interpretations of the ADF of Example 3

as is presented in Example 3. This shows the importance of choosing a right set of parents that satisfies the conditions of Definition 8 for a queried argument. However, there exists an alternative method for checking whether an argument is strongly justified, presented in (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b), in which there is no need of indicating a set of parents of a queried argument.

Definition 9. Let $D = (A, L, C)$ be an ADF and let v be an interpretation of D. An interpretation v is a *strongly admissible* interpretation if for each a such that $v(a) = \mathbf{t}/\mathbf{f}$, it holds that a is a strongly justified argument in v.

To clarify the notion of strongly admissible interpretations of ADFs, we continue Example 3 in Example 4.

Example 4. Consider again the ADF of Example 3, i.e., $D = (\{a, b, c, d\}, \{\varphi_a : \top, \varphi_b : a \wedge \neg c, \varphi_c : \neg b \wedge d, \varphi_d : \bot\})$, depicted in Figure 1. Let $v = \{b, \neg c, \neg d\}$. As shown in Example 3, c and d are strongly justified in v. However, b is not strongly justified in v. Thus, v is not a strongly admissible interpretation of D. However, for instance, $v_1 = \{a\}$, $v_2 = \{\neg c, \neg d\}$ and $v_3 = \{a, b, \neg c, \neg d\}$ are strongly admissible interpretations of D. We show that b is strongly acceptable in v_3. To this end, let $P = \{a, c\}$ be a set of parents of b. First, it holds that $\varphi_b^{v_3|_P} \equiv \top$. Thus, the first condition is satisfied for b. We also have to check whether each parent of b is strongly justified in v_3. To this end, we show that a is strongly acceptable in v_3 and c is strongly deniable in v_3. The latter is obvious by the same method that was presented in Example 3 to show that c is strongly deniable in v. In addition, $\varphi_a^{v_{\mathbf{u}}} \equiv \top$, thus, a is strongly acceptable in v_3. Hence, b and a are strongly justified in v_3. Furthermore, v_3 is a unique grounded interpretation of D.

It is shown in (2021b) that the strongly admissible interpretations of D form a lattice with respect to the $\leq_i$-ordering, with the least element being $v_{\mathbf{u}}$ and the maximum element being the grounded interpretation of D. The set of strongly admissible interpretations of ADF D given in Example 3 form a lattice, depicted in Figure 2.

2.4 Algorithm for Strongly Admissible Interpretations of ADFs

In this section we review an existing method, presented in Section 5 of (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b), to answer the verification problem under strong admissibility semantics. To this end, we introduce $\Gamma_{D,v}$, a variant of the characteristic operator restricted to a given interpretation v.

Definition 10. Let D be an ADF and let v, w be interpretations of D. Let $\Gamma_{D,v}(w) = \Gamma_D(w) \sqcap_i v$, where $\Gamma^n_{D,v}(w) = \Gamma_{D,v}(\Gamma^{n-1}_{D,v}(w))$ for n with $n \geq 1$, and $\Gamma^0_{D,v}(w) = w$.

We next use the $\Gamma_{D,v}$ operator to recall observations on the sequence of interpretations generated by a least fixed-point iteration on $\Gamma_{D,v}$.

Lemma 1 ((Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b))**.** *Let $D = (A, L, C)$ be a given ADF and let v be an interpretation of D. Let $\Gamma^n_{D,v}(v_{\mathbf{u}})$ be the set of interpretations constructed based on v, as in Definition 10. For each i it holds that;*

- $\Gamma^i_{D,v}(v_{\mathbf{u}}) \leq_i \Gamma^{i+1}_{D,v}(v_{\mathbf{u}})$*;*
- $\Gamma^i_{D,v}(v_{\mathbf{u}})$ *is a strongly admissible interpretation of D;*
- *if $\Gamma^i_{D,v}(v_{\mathbf{u}})(a) = \mathbf{t}/\mathbf{f}$, then a is strongly justifiable in $\Gamma^i_{D,v}(v_{\mathbf{u}})$.*

The sequence of interpretations $\Gamma^i_{D,v}(v_{\mathbf{u}})$ as defined in Definition 10 is named the sequence of strongly admissible interpretations constructed based on v in D.

Based on the above observations, one can characterise strongly admissible interpretations v as least fixed point of the corresponding operator $\Gamma_{D,v}$. That is, we can verify an interpretation by computing this sequence of strongly admissible interpretations.

Theorem 1 ((Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b))**.** Let D be an ADF and let v be an interpretation of D. Let $\Gamma^i_{D,v}(v_{\mathbf{u}})$ (for $i \geq 0$) be the sequence of strongly admissible interpretations constructed based on v in D. The following conditions hold:

- there is an m with $m \geq 0$ s.t. $\Gamma^m_{D,v}(v_{\mathbf{u}}) \sim_i \Gamma^{m+1}_{D,v}(v_{\mathbf{u}})$;
- $v \in sadm(D)$ iff there exists an m s.t. $v \sim_i \Gamma^m_{D,v}(v_{\mathbf{u}})$.

Example 5 illustrates the role of Theorem 1 in the verification problem under the strong admissibility semantics.

Example 5. Consider again the ADF given in Example 3, i.e., $D = (\{a, b, c, d\}, \{\varphi_a : \top, \varphi_b : a \wedge \neg c, \varphi_c : \neg b \wedge d, \varphi_d : \bot\})$. Let $v = \{a, \neg c, \neg d\}$. We check whether $v \in sadm(D)$ based on the method presented in Theorem 1. The sequence of strongly admissible interpretations constructed based on v is as follows.
$v_1 = \Gamma_{D,v}(v_{\mathbf{u}}) = \{a, \neg d\} \sqcap_i \{a, \neg c, \neg d\} = \{a, \neg d\}$;
$v_2 = \Gamma^2_{D,v}(v_{\mathbf{u}}) = \{a, \neg c, \neg d\} \sqcap_i \{a, \neg c, \neg d\} = \{a, \neg c, \neg d\}$.
Since $v \sim_i \Gamma^2_{D,v}(v_{\mathbf{u}})$, it holds that $v \in sadm(D)$.

On the other hand, let $v' = \{a, b\}$. We show that $v' \notin sadm(D)$. The sequence of interpretations constructed based on v' is as follows:
$v_1 = \Gamma_D(v_{\mathbf{u}}) \sqcap_i v' = \{a, \neg d\} \sqcap_i \{a, b\} = \{a\}$;
$v_2 = \Gamma_D(v_1) \sqcap_i v' = \{a, \neg d\} \sqcap_i \{a, b\} = \{a\}$.
Thus, the sequence of interpretations constructed based on v' leads to $v_2 = \{a\}$, which is not equal to v', i.e., $v' \not\sim_i v_2$. Hence, v' is not a strongly admissible interpretation of D.

Based on the above results (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b) provides algorithms that decide (a) verification of a given strongly admissible interpretation and (b) whether an argument is strongly acceptable/deniable within a given interpretation that are based on an iterative fixed-point computation of an operator $\Gamma_{D,v}$. However, because testing whether an argument is acceptable in Γ_D is already NP/coNP-hard (Dvořák and Dunne 2018), these procedures are in P^{NP} and as we will show, both problems allow for algorithms of significantly lower complexity.

3 Computational Complexity

We analyse the complexity under strong admissibility semantics for (a) the standard reasoning tasks of ADFs (Dvořák and Dunne 2018) and (b) two problems specific to strong admissibility semantics, i.e., the small witness problem introduced for AFs in (Dvořák and Wallner 2020; Caminada and Dunne 2020) and the strong justification problem.

For a given ADF D we consider the following problems:

1. *The credulous decision problem*: whether an argument a is credulously justifiable with respect to the strong admissibility semantics of D. That is, if there exists a strongly admissible interpretation of D in which a is strongly justified. This reasoning task is denoted as $\textit{Cred}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D)$ and is presented formally as follows:

$$\textit{Cred}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D) = \begin{cases} \text{yes} & \text{if } \exists v \in \textit{sadm}(D) \text{ s.t.} \\ & v(a) = \mathbf{t}/\mathbf{f}, \\ \text{no} & \text{otherwise} \end{cases}$$

2. *The skeptical decision problem*: whether an argument a is skeptically justified with respect to the strong admissibility semantics of D. That is, if a is strongly justified in all strongly admissible interpretations of D, denoted as $\textit{Skept}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D)$, which is presented formally as follows:

$$\textit{Skept}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D) = \begin{cases} \text{yes} & \text{if } \forall v \in \textit{sadm}(D) : \\ & v(a) = \mathbf{t}/\mathbf{f} \text{ holds,} \\ \text{no} & \text{otherwise} \end{cases}$$

3. *The verification problem*: whether a given interpretation v is a strongly admissible interpretation of D, denoted by $\textit{Ver}_{\textit{sadm}}(v, D)$, which is presented formally as follows:

$$\textit{Ver}_{\textit{sadm}}(v, D) = \begin{cases} \text{yes} & \text{if } v \in \textit{sadm}(D), \\ \text{no} & \text{otherwise} \end{cases}$$

4. *The strong justification problem:* The problem whether a given argument a is strongly justified in a given interpretation v is denoted as $\textit{StrJust}(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$, which is presented formally as follows:

$$\textit{StrJust}(a \mapsto \mathbf{t}/\mathbf{f}, v, D) = \begin{cases} \text{yes} & \text{if } a \text{ is strongly} \\ & \text{justified in } v, \\ \text{no} & \text{otherwise} \end{cases}$$

5. *The small witness problem:* We are interested in computing a strongly admissible interpretation that has the least information of the ancestors of a given argument, namely a, where $v(a) = \mathbf{t}/\mathbf{f}$. The decision version of this problem is the k-Witness problem, denoted by $k\text{-}\textit{Witness}_{\textit{sadm}}$, indicating whether a given argument is strongly justified in at least one v such that $v \in \textit{sadm}(D)$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| \leq k$. Note that k is part of the input of this problem. This decision problem is presented formally as follows:

$$k\text{-}\textit{Witness}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D) = \begin{cases} \text{yes} & \text{if } \exists v \in \textit{sadm}(D) \\ & \text{s.t. } v(a) = \mathbf{t}/\mathbf{f} \\ & \& |v^{\mathbf{t}} \cup v^{\mathbf{f}}| \leq k, \\ \text{no} & \text{otherwise} \end{cases}$$

3.1 The Credulous/Skeptical Decision Problems

In this section we study the credulous/skeptical problem under the strong admissibility semantics for ADFs. That is, we show the complexity of deciding whether an argument in question is credulously/skeptically justifiable in at least one/all strongly admissible interpretation(s) of a given ADF.

We show that $\textit{Cred}_{\textit{sadm}}$ is coNP-complete and $\textit{Skept}_{\textit{sadm}}$ is trivial. To this end, we use the fact that the set of strongly admissible interpretations of a given ADF D forms a lattice with respect to the $\leq_i$-ordering, with the maximum element being $\textit{grd}(D)$. Thus, any strongly admissible interpretation of D has at most an amount of information equal to $\textit{grd}(D)$. Thus, answering the credulous decision problem under the strong admissibility semantics coincides with answering the credulous decision problem under the grounded semantics.

Theorem 2. *$\textit{Cred}_{\textit{sadm}}$ is* coNP*-complete.*

Proof. We have that $\textit{Cred}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D) = \textit{Cred}_{\textit{grd}}(a \mapsto \mathbf{t}/\mathbf{f}, D)$ and the latter has been shown to be coNP-complete in (Wallner 2014, Proposition 4.1.3.). □

Concerning skeptical acceptance, notice that the trivial interpretation is the least strongly admissible interpretation in each ADF. Thus, $\textit{Skept}_{\textit{sadm}}(a \mapsto \mathbf{t}/\mathbf{f}, D)$ is trivially *no*.

Theorem 3. *$\textit{Skept}_{\textit{sadm}}$ is a trivial problem.*

3.2 The Verification Problem

In this section, we settle the complexity of $\textit{Ver}_{\textit{sadm}}(v, D)$, i.e., of deciding whether a given interpretation v is a strongly admissible interpretation of an ADF D. We have seen in Section 2.4 that this problem can be solved in P^{NP}.

We first sketch a simple translation-based approach that reduces the verification problem of strongly admissible semantics to the verification problem of grounded semantics. In order to reduce $\textit{Ver}_{\textit{sadm}}(v, D)$ to $\textit{Ver}_{\textit{grd}}(v, D')$, we modify the acceptance conditions φ_a of D to $\varphi'_a = \neg a$ if $v(a) = \mathbf{u}$ and $\varphi'_a = \varphi_a$ otherwise. We then have that $v \in \textit{sadm}(D)$ iff $v \in \textit{grd}(D)$, so that we can use the DP procedure for $\textit{Ver}_{\textit{grd}}(v, D')$ (Wallner 2014, Theorem 4.1.4). This gives a DP procedure. However, as we will discuss next, $\textit{Ver}_{\textit{sadm}}(v, D)$ can be solved within coNP.

Intuitively, since the grounded interpretation is the maximum element of the lattice of strongly admissible interpretations and the credulous decision problem under grounded semantics is coNP-complete, it seems that the verification problem under the strong admissibility semantics has to be coNP-complete. However, having the positive answer for $\mathit{Cred}_{\mathit{grd}}(a \mapsto \mathbf{t}/\mathbf{f}, D)$ for each a with $v(a) = \mathbf{t}/\mathbf{f}$ does not lead to the positive answer of $\mathit{Ver}_{\mathit{sadm}}(v, D)$. This is because $v \leq_i \mathit{grd}(D)$ does not imply that v is a strongly admissible interpretation of D (see Example 6 below).

Example 6. Let $D = (\{a, b\}, \{\varphi_a : \top, \varphi_b : a \vee b\})$. The grounded interpretation of D is $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}\}$. Furthermore, the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{t}\}$ is an admissible interpretation of D such that $v \leq_i \mathit{grd}(D)$. However, v is not a strongly admissible interpretation of D. As we know, the answer of $\mathit{Cred}_{\mathit{grd}}(b \mapsto \mathbf{t}, D)$ is *yes*, but b is not strongly acceptable in v. Thus, v is not a strongly admissible interpretation of D, i.e., the answer to $\mathit{Ver}_{\mathit{sadm}}(v, D)$ is *no*.

To show that $\mathit{Ver}_{\mathit{sadm}}$ is coNP-complete, we modify and combine both the fixed-point iteration from Section 2.4 and the grounded algorithm from (Wallner 2014). To this end, we need some auxiliary results that are shown in Lemmas 2 and 3.

Lemma 2. *Given an ADF D with n arguments, the following statements are equivalent:*

1. *v is a strongly admissible interpretation of D;*
2. *$v = \Gamma^n_{D,v}(v_\mathbf{u})$;*
3. *for each $w \leq_i v$, it holds that $v = \Gamma^n_{D,v}(w)$.*

Proof. • $1 \leftrightarrow 2$: by Theorem 1.

- $2 \mapsto 3$: Assume that $v = \Gamma^n_{D,v}(v_\mathbf{u})$ and that $w \leq_i v$. We show that $v = \Gamma^n_{D,v}(w)$. Since $v_\mathbf{u} \leq_i w \leq_i v$, and Γ_D is monotonic and thus also $\Gamma_{D,v}$ monotonic, we have $\Gamma^n_{D,v}(v_\mathbf{u}) \leq_i \Gamma^n_{D,v}(w) \leq_i \Gamma^n_{D,v}(v)$. Now using that $v = \Gamma^n_{D,v}(v_\mathbf{u})$, we obtain $v \leq_i \Gamma^n_{D,v}(w) \leq_i \Gamma^{2n}_{D,v}(v_\mathbf{u})$. Because $\Gamma_{D,v}$ is a monotonic operator, the fixed-point is reached after at most n iterations and thus $\Gamma^{2n}_{D,v}(v_\mathbf{u}) = \Gamma^n_{D,v}(v_\mathbf{u}) = v$. Hence, $\Gamma^n_{D,v}(w) = v$.
- $3 \mapsto 2$: Assume that for each $w \leq_i v$ it holds that $v \sim_i \Gamma^n_{D,v}(w)$. Thus, since $v_\mathbf{u} \leq_i v$, it holds that $v \sim_i \Gamma^n_{D,v}(v_\mathbf{u})$.

□

In the following, let $v^* = v^\mathbf{t} \cup v^\mathbf{f}$. The notions of completion of an interpretation and model are presented in Definition 11, used in Lemma 3.

Definition 11. Let w be an interpretation. We define the *completion* of w as the set of all two-valued extensions of w, denoted by $[w]_2$ where: $[w]_2 = \{u \mid w \leq_i u \text{ and } u \text{ is a two-valued interpretation}\}$.

Furthermore, a two-valued interpretation u is said to be a *model* of formula φ, if $u(\varphi) = \mathbf{t}$, denoted by $u \models \varphi$.

Lemma 3. *Let D be an ADF and let v be an interpretation of D. $v \notin \mathit{sadm}(D)$ if and only if there exists an interpretation w of D that satisfies all the following conditions:*

1. *$w <_i v$;*
2. *For each $a \in w^\mathbf{u} \cap v^\mathbf{t}$ there exists $u_a \in [w]_2$ s.t. $u_a \not\models \varphi_a$;*
3. *For each $a \in w^\mathbf{u} \cap v^\mathbf{f}$ there exists $u_a \in [w]_2$ s.t. $u_a \models \varphi_a$.*

Proof. $\Leftarrow$: Assume that v and w are interpretations of D that satisfy all of the items 1, 2, 3 presented in the lemma. We show that $v \notin \mathit{sadm}(D)$. Toward a contradiction assume that $v \in \mathit{sadm}(v)$. Let a be an argument such that $a \in w^\mathbf{u} \cap v^\mathbf{t}$, thus, since w satisfies the conditions of the lemma, it holds that there exists $u_a \in [w]_2$ such that $u_a \not\models \varphi_a$, i.e., $u_a(a) = \mathbf{f}$. Furthermore, since $v(a) = \mathbf{t}$ and $v \in \mathit{sadm}(D)$, for any $j \in [v]_2$ it holds that $j \models \varphi_a$. Since $w <_i v$, it holds that $j \in [w]_2$, i.e., $\Gamma_D(w)(a) = \mathbf{u}$. The proof method for the case that $a \in w^\mathbf{u} \cap v^\mathbf{f}$ is similar, i.e., if $a \in w^\mathbf{u} \cap (v^\mathbf{t} \cup v^\mathbf{f})$, then $\Gamma_D(w)(a) = \mathbf{u}$. Thus, for $a \in w^\mathbf{u} \cap v^*$ we have $\Gamma_{D,v}(w)(a) = (\Gamma_D(w) \sqcap v)(a) = \mathbf{u}$. In other words, $\Gamma_{D,v}(w)(a) \leq_i w$ and thus, by the monotonicity of $\Gamma_{D,v}(w)$ also $\Gamma^n_{D,v}(w)(a) \leq_i w <_i v$. Thus, since $\Gamma^n_{D,v}(w) \not\sim_i v$ the third item of Lemma 2 does not hold for w with $w <_i v$. Thus, $v \notin \mathit{sadm}(D)$.
$\Rightarrow$: Assume that $v \notin \mathit{sadm}(D)$. That is, for the fixed point $w = \Gamma^n_{D,v}(v_\mathbf{u})$ we have $w <_i v$. Consider $a \in w^\mathbf{u} \cap v^\mathbf{t}$. Because w is a fixed point, we have that $\Gamma_{D,v}(w)(a) \neq \mathbf{t}$ and thus $\Gamma_D(w) \neq \mathbf{t}$. That is, there is a $u_a \in [w]_2$ such that $u_a \not\models \varphi_a$. Similar reasoning applies to $a \in w^\mathbf{u} \cap v^\mathbf{f}$. □

Lemma 4 shows that the verification problem is a coNP-problem, and Lemma 5 shows the hardness of this problem.

Lemma 4. *$\mathit{Ver}_{\mathit{sadm}}$ is a* coNP*-problem for ADFs.*

Proof. Let D be an ADF and let v be an interpretation of D. For membership, consider the co-problem. By Lemma 3, if there exists an interpretation of w that satisfies the condition of Lemma 3, then v is not a strongly admissible interpretation of D. Thus, guess an interpretation w, together with an interpretation $u_a \in [w]_2$ for each $a \in v^*$, and check whether they satisfy the conditions of Lemma 3. Note that since $w <_i v$ we have to check the second and the third items of Lemma 3 a total of $|v^* \setminus w^\mathbf{u}|$ number of times. That is, this checking has to be done at most $|v^*|$ number of times, when w is the trivial interpretation. Thus, this checking step is linear in the size of v^*. Therefore, the procedure of guessing of w and checking if it satisfies 1, 2, 3 of Lemma 3 is an NP-problem. Thus, if a w satisfies the items of Lemma 3, then the answer to $\mathit{Ver}_{\mathit{sadm}}(v, D)$ is *no*. Otherwise, if we check all interpretations w such that $w <_i v$ and none of them satisfies the conditions of Lemma 3, then the answer to $\mathit{Ver}_{\mathit{sadm}}(v, D)$ is *yes*. Thus, $\mathit{Ver}_{\mathit{sadm}}(v, D)$ is a coNP-problem. □

Lemma 5. *$\mathit{Ver}_{\mathit{sadm}}$ is* coNP*-hard for ADFs.*

Proof. For hardness of $\mathit{Ver}_{\mathit{sadm}}$, we consider the standard propositional logic problem of VALIDITY. Let ψ be an arbitrary Boolean formula and let $X = \mathit{atom}(\psi)$ be the set of atoms in ψ. Let a be a new atom, i.e., $a \notin X$. Construct ADF $D = (\{X \cup \{a\}\}, L, C)$ where $\varphi_x : x$ for each $x \in X$ and $\varphi_a : \psi$. We show that ψ is valid if and only if $v = v_\mathbf{u}|^a_\mathbf{t}$ is a strongly admissible interpretation of D. An illustration of the reduction for the formula $\psi = \neg b \vee b$ to the ADF $D = (\{a, b\}, L, \varphi_a : \psi, \varphi_b : b)$ is shown in Figure 3.

Assume that ψ is a valid formula. We show that v is the grounded interpretation of D. By the acceptance condition of each x, for $x \in X$ it is clear that x is assigned to $\mathbf{u}$ in the grounded interpretation of D. Further, since ψ is a valid

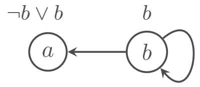


Figure 3: Reduction used in Lemma 5 and 9, for $\psi = \neg b \vee b$.

formula, it holds that ${\varphi_a}^{v_{\mathbf{u}}} \equiv \top$. Thus, the interpretation $v = v_{\mathbf{u}}|^a_{\mathbf{t}}$ is the grounded interpretation of D. Hence, $v \in$ *sadm*(D).

On the other hand, assume that ψ is not valid. Then there exists a two-valued interpretation v of $atom(\psi)$ such that $v \not\models \psi$. This implies that $a \mapsto \mathbf{t}$ does not belong to the grounded interpretation of D. Since the grounded interpretation of D is the maximum element of the lattice of strongly admissible interpretations, it holds that a is not strongly acceptable in any strongly admissible interpretation of D, that is, $v \notin$ *sadm*(D). □

Theorem 4 is a direct result of Lemmas 4–5.

Theorem 4. *Ver*$_{sadm}$ is coNP-complete for ADFs.

3.3 Strong Justification of an Argument

Note that it is possible that an interpretation v contains some strongly justified arguments but v is not strongly admissible itself. Example 7 presents such an interpretation. Thus, the problem *StrJust*$(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$ of deciding whether an argument is strongly justified in a given interpretation of an ADF is different from the previously discussed decision problems. We show that *StrJust* is coNP-complete.

Example 7. Let $D = (\{a, b, c, d\}, \{\varphi_a : \bot, \varphi_b : \neg a \wedge c, \varphi_c : d, \varphi_d : \top\})$ be an ADF. Let $v = \{b, c, d\}$ be an interpretation of D. It is easy to check that c and d are strongly acceptable in v. However, b is not strongly acceptable in D. Thus, v is not a strongly admissible interpretation of D. However, there exists a strongly admissible interpretation of D in which c and d are strongly acceptable and that has less information than v, namely, $v' = \{c, d\}$.

As discussed in Section 2.4, (Keshavarzi Zafarghandi, Verbrugge, and Verheij 2021b) presents a straightforward method of deciding whether a is strongly justified in a given interpretation v. That is, a is strongly acceptable/deniable in v if it is acceptable/deniable by the least fixed point of the operator $\Gamma_{D,v}$ (which is equal to $\Gamma^n_{D,v}(v_{\mathbf{u}})$ for sufficiently large n).

However, the repeated evaluation of Γ_D is a costly part of this algorithm and results in a P^{NP} algorithm. We will next discuss a more efficient method to answer this reasoning task. To this end, we translate a given ADF D to ADF D', presented in Definition 12, such that the queried argument is strongly justifiable in a given interpretation of D if and only if it is credulously justifiable in the grounded interpretation of D'. As shown in Proposition 4.1.3 in (Wallner 2014), the credulous decision problem for ADFs under grounded semantics is a coNP-problem. Thus, verifying whether a given argument is strongly justified in an interpretation is a coNP-problem, since the translation can be done in polynomial time with respect to the size of D.

Definition 12. Let $D = (A, L, C)$ be an ADF and let v be an interpretation of D. The translation of D under v is $D' = (A', L', C')$ such that $A' = A \cup \{x, y\}$ where $x, y \notin A$. Furthermore, for each $a \in A'$ we define the acceptance condition of a in D', namely φ'_a as follows:

- $\varphi'_x : x$;
- $\varphi'_y : y$;
- if $v(a) = \mathbf{u}$, then $\varphi'_a : \neg a$;
- if $v(a) = \mathbf{t}$, then $\varphi'_a = \varphi_a \vee x$;
- if $v(a) = \mathbf{f}$, then $\varphi'_a = \varphi_a \wedge y$.

Notice that our reduction ensures that arguments with $v(a) = \mathbf{u}$ will always be $\mathbf{u}$ in D', arguments with $v(a) = \mathbf{t}$ will be assigned to either $\mathbf{t}$ or $\mathbf{u}$ during the least fixed-point computation and arguments with $v(a) = \mathbf{f}$ will be assigned to either $\mathbf{f}$ or $\mathbf{u}$. That is we introduced arguments x, y to ensure that arguments in v^* are not assigned to the opposite truth value during the iteration of $\Gamma_{D'}$ that leads to the grounded interpretation of D'.

Lemmas 6 and 7 show the correctness of the reduction.

Lemma 6. *Let D be an ADF, let v be an interpretation of D, and let D' be the translation of D, via Definition 12. It holds that if StrJust$(a \mapsto \mathbf{t}/\mathbf{f}, v, D) = yes$, then Cred$_{grd}(a \mapsto \mathbf{t}/\mathbf{f}, D') = yes$.*

Proof. We assume that *StrJust*$(a \mapsto \mathbf{t}, v, D)$ = yes, and we show that *Cred*$_{grd}(a \mapsto \mathbf{t}, D') =$ yes. The proof for the case that *StrJust*$(a \mapsto \mathbf{f}, v, D)$ = yes is similar.

Assume that $v_{\mathbf{u}}$ is the trivial interpretation of D and $v'_{\mathbf{u}}$ is the trivial interpretation of D', i.e., $v'_{\mathbf{u}} = v_{\mathbf{u}} \cup \{x \mapsto \mathbf{u}, y \mapsto \mathbf{u}\}$. Assume that $\Gamma^i_{D,v}(v_{\mathbf{u}})$ is a sequence of strongly admissible interpretations constructed based on v in D, as in Definition 10. Let w be the limit of the sequence of $\Gamma^i_{D,v}(v_{\mathbf{u}})$.

StrJust$(a \mapsto \mathbf{t}, v, D) = yes$ implies that $w(a) = \mathbf{t}$. Since w is a strongly admissible interpretation of D, it holds that $a \mapsto \mathbf{t}$ in the grounded interpretation of D, i.e., there exists a natural number n such that $\Gamma^n_D(v_{\mathbf{u}})(a) = \mathbf{t}$. By induction on n, it is easy to show that $\Gamma^n_{D'}(v'_{\mathbf{u}})(a) = \mathbf{t}$. That is, a is assigned to $\mathbf{t}$ in the grounded interpretation of D'. Thus, *Cred*$_{grd}(a \mapsto \mathbf{t}, D') =$ yes. □

Lemma 7. *Let D be an ADF, let v be an interpretation of D, and let D' be the translation of D via Definition 12. It holds that if Cred$_{grd}(a \mapsto \mathbf{t}/\mathbf{f}, D) = yes$, then StrJust$(a \mapsto \mathbf{t}/\mathbf{f}, v, D) = yes$.*

Proof. Assume that a is justified in the grounded interpretation of D', namely w. Thus, there exists a j such that $w = \Gamma^j_{D'}(w_{\mathbf{u}})$ for $j \geq 0$, where $w_{\mathbf{u}}$ is the trivial interpretation of D'. By induction we prove the claim that for all i, if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^i_{D'}(w_{\mathbf{u}})$, then a is strongly justified in v.

Base case: Assume that $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^1_{D'}(w_{\mathbf{u}})$. By the acceptance conditions of x and y in D', both of them are assigned to $\mathbf{u}$ in w. Then it has to be the case that either $\varphi'_a = \varphi_a \vee x$ or $\varphi'_a = \varphi_a \wedge y$ in D'. Thus, $a \mapsto \mathbf{t}/\mathbf{f} \in$

$\Gamma^1_{D'}(w_\mathbf{u})$ implies that ${\varphi'_a}^{w_\mathbf{u}} \equiv \top/\bot$. Thus, $w(x/y) = \mathbf{u}$, $\varphi'_a = \varphi_a \vee x/\varphi_a \wedge y$ and ${\varphi'_a}^{w_\mathbf{u}} \equiv \top/\bot$ together imply that ${\varphi_a}^{w_\mathbf{u}} \equiv \top/\bot$. Hence, $\varphi_a^{v_\mathbf{u}} \equiv \top/\bot$ where $v_\mathbf{u}$ is the trivial interpretation of D. That is, a is strongly justified in v.

Induction hypothesis: Assume that for all j with $1 \leq j \leq i$, if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^j_{D'}(w_\mathbf{u})$, then a is strongly justified in v.

Inductive step: We show that if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^{i+1}_{D'}(w_\mathbf{u})$, then a is strongly justified in v. Because $x/y \mapsto \mathbf{u} \in w$, we have that $\varphi_a^w \equiv \top/\bot$ implies that $\varphi_a^v \equiv \top/\bot$. Further, $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^{i+1}_{D'}(w_\mathbf{u})$ says that there exists a set of parents of a, namely P, where $P \subseteq w^\mathbf{t} \cup w^\mathbf{f}$, such that, ${\varphi_a}^{w_{|P}} \equiv \top/\bot$. Thus, ${\varphi_a}^{v_{|P}} \equiv \top/\bot$. By induction hypothesis, each $p \in P$ is strongly justified in v. Thus, a is strongly justified in v. □

Theorem 5 is a direct result of Lemmas 6 and 7.

Theorem 5. Let D be an ADF, let v be an interpretation of D, and let D' be the translation of D, via Definition 12. It holds that *Cred*$_{grd}(a \mapsto \mathbf{t}/\mathbf{f}, D) = yes$ iff *StrJust*$(a \mapsto \mathbf{t}/\mathbf{f}, v, D) = yes$.

We use the auxiliary Theorem 5 to present the main result of this section, i.e., to show that *StrJust* is coNP-complete.

Lemma 8. *Let D be an ADF, let a be an argument, and let v be an interpretation of D. Deciding whether a is strongly justified in v, i.e., whether StrJust$(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$, is a* coNP*-problem.*

Proof. It is shown in (Wallner 2014, Proposition 4.1.3) that the credulous decision problem under grounded semantics, i.e., *Cred*$_{grd}$, is a coNP-problem. Further, the translation of a given ADF D to D' via Definition 12 can be done in polynomial time. By Theorem 5, it holds that *Cred*$_{grd}(a \mapsto \mathbf{t}/\mathbf{f}, D) = yes$ iff *StrJust*$(a \mapsto \mathbf{t}/\mathbf{f}, v, D) = yes$. Thus, deciding whether a given argument is strongly justified in interpretation v, i.e., *StrJust*$(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$ is a coNP-problem. □

Lemma 9. *Let D be an ADF, let a be an argument, and let v be an interpretation of D. Deciding whether a is strongly justified in v, i.e., StrJust$(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$, is* coNP*-hard.*

Proof. Let ψ be any Boolean formula and let $X = atom(\psi)$ be the set of atoms in ψ. Let a be a new variable that does not appear in X. Construct $D = (\{X \cup \{a\}\}, L, C)$, such that $\varphi_x : x$ for each $x \in X$ and $\varphi_a : \psi$. ADF D can be constructed in polynomial time with respect to the size of ψ. We show that a is strongly acceptable in any v where $v(a) = \mathbf{t}$ if and only if ψ is a valid formula. An illustration of the reduction for a formula $\psi = \neg b \vee b$ to the ADF $D = (\{a, b\}, L, \varphi_a : \psi, \varphi_b : b)$ is depicted in Figure 3.

Assume that a is strongly acceptable in v, thus by Definition 8, there exists a set of parents of a, namely P, such that ${\varphi_a}^{v_{|P}} \equiv \top$ and for each $p \in P$ it holds that p is strongly justified in v. By the definition of D the acceptance condition of each parent of a, namely p is $\varphi_p : p$, thus, by the acceptance condition of p, it is not strongly justifiable in v. Thus, the only case in which a is strongly acceptable in v is that $P = \emptyset$, i.e., $\varphi_a^{v_\mathbf{u}} \equiv \top$. Hence, for any two-valued interpretation u of $X \cup \{a\}$ it holds that $u \models \psi$. Moreover since the atom a does not appear in ψ we obtain that for any two-valued interpretation u of X it holds that $u \models \psi$. Hence, ψ is a valid formula and it is a *yes* instance of the VALIDITY problem of classical logic.

On the other hand, assume that ψ is a valid formula. Then it is clear that the interpretation v that assigns a to $\mathbf{t}$ and x to $\mathbf{u}$, for each $x \in X$, is the grounded interpretation of D. Thus, the answer to the strong acceptance problem of a in any v with $v(a) = \mathbf{t}$ is *yes*.

For credulous denial of a, it is enough to present the acceptance condition of a equal to the negation of ψ in D, i.e., $\varphi_a : \neg\psi$, and follow a similar method. That is, a is strongly deniable in v, where $v(a) = \mathbf{f}$, if and only if ψ is a valid formula. □

Theorem 6 is a direct result of Lemmas 8 and 9.

Theorem 6. Let D be an ADF, let a be an argument, and let v be an interpretation of D. Deciding whether a is strongly justified in v, i.e., *StrJust*$(a \mapsto \mathbf{t}/\mathbf{f}, v, D)$ is coNP-complete.

3.4 Smallest Witness of Strong Justification

Assume that an argument a, its truth value, and a natural number k are given. We are eager to know whether there exists a strongly admissible interpretation v that satisfies the truth value of a and $|v^\mathbf{t} \cup v^\mathbf{f}| < k$. This reasoning task is denoted by *k-Witness*$_{sadm}(a \mapsto \mathbf{t}/\mathbf{f}, D)$. We show that *k-Witness*$_{sadm}$ is Σ_2^P-complete. Lemma 10 shows that this problem is a Σ_2^P-problem and Lemma 11 indicates the hardness of this reasoning task.

Lemma 10. *Let $D = (A, L, C)$ be an ADF, let a be an argument, let $x \in \{\mathbf{t}, \mathbf{f}\}$, and let k be a natural number. Deciding whether there exists a strongly admissible interpretation v of D where $v(a) = x$ and $|v^\mathbf{t} \cup v^\mathbf{f}| < k$ is a Σ_2^P-problem, i.e., k-Witness$_{sadm}$ is a Σ_2^P-problem.*

Proof. For membership, non-deterministically guess an interpretation v and verify whether this interpretation satisfies the following items:

1. $v \in sadm(D)$;
2. $v(a) = x$;
3. $|v^\mathbf{t} \cup v^\mathbf{f}| < k$.

If v satisfies all the items, then the answer to the decision problem is *yes*, i.e., *k-Witness*$_{sadm}(a \mapsto \mathbf{t}/\mathbf{f}, D) = yes$. The complexity of each of the above items is as follows.

1. Verifying strong admissibility of v is coNP-complete, as is presented in Section 3.2.
2. Verifying if v contains the claim, i.e., if $v(a) = x$, can clearly be done in polynomial time.
3. Collecting $v^\mathbf{t} \cup v^\mathbf{f}$ and checking whether $|v^\mathbf{t} \cup v^\mathbf{f}| < k$ takes only polynomial time.

That is, the algorithm first non-deterministically guesses an interpretation v and then performs checks that are in coNP to verify that v satisfies the requirements of the decision problem. Thus, this gives an $\mathsf{NP}^{\mathsf{coNP}} = \Sigma_2^\mathsf{P}$ procedure. □

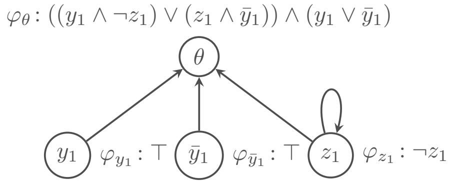

$\varphi_\theta: ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \bar{y}_1)) \wedge (y_1 \vee \bar{y}_1)$
θ
y_1
$\varphi_{y_1}: \top$
$\bar{y}_1$
$\varphi_{\bar{y}_1}: \top$
z_1
$\varphi_{z_1}: \neg z_1$

Figure 4: Illustration of the reduction from the proof of Lemma 11 for $\Theta = \exists y_1 \forall z_1((y_1 \wedge \neg z_1) \vee (z_1 \wedge \neg y_1)) \wedge (y_1 \vee \neg y_1)$.

Lemma 11. *Let $D = (A, L, C)$ be an ADF, let a be an argument, let $x \in \{\mathbf{t}, \mathbf{f}\}$, and let k be a natural number. Deciding whether there exists a strongly admissible interpretation v of D where $v(a) = x$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| < k$ is Σ_2^{P}-hard, i.e., k-Witness$_{sadm}$ is Σ_2^{P}-hard.*

Proof. Consider the following well-known problem on quantified Boolean formulas. Given a formula $\Theta = \exists Y \forall Z\, \theta(Y, Z)$ with atoms $X = Y \cup Z$ (and $Y \cap Z = \emptyset$) and propositional formula θ. Deciding whether Θ is valid is Σ_2^{P}-complete (see e.g. (Arora and Barak 2009)). We can assume that θ is of the form $\psi \wedge \bigwedge_{y \in Y}(y \vee \neg y)$, where ψ is an arbitrary propositional formula over atoms X, and that θ is satisfiable. Moreover, we can assume that the formula θ only uses $\wedge, \vee, \neg$ operations and negations only appear in literals. Let $\bar{Y} = \{\bar{y} : y \in Y\}$, i.e., for each $y \in Y$ we introduce a new argument $\bar{y}$.

We construct an ADF $D_\Theta = (A, L, C)$ with

$$\begin{aligned} A =& Y \cup \bar{Y} \cup Z \cup \{\theta\} \\ C =& \{\varphi_y : \top \mid y \in Y\} \cup \{\varphi_{\bar{y}} : \top \mid y \in Y\} \\ &\cup \{\varphi_z : \neg z \mid z \in Z\} \cup \{\varphi_\theta : \theta[\neg y / \bar{y}]\} \end{aligned}$$

It is easy to verify that the grounded interpretation g of D_Θ sets all arguments $Y \cup \bar{Y}$ to $\mathbf{t}$ and all arguments Z to $\mathbf{u}$. Moreover, $g(\theta) \in \{\mathbf{t}, \mathbf{u}\}$. An illustration of the reduction for a formula $\theta = ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \neg y_1)) \wedge (y_1 \vee \neg y_1)$ to the ADF $D = (A, L, C)$ is shown in Figure 4, where: $A = \{y_1, \bar{y_1}, z_1, \theta\}$, $\varphi_{y_1} : \top, \varphi_{\bar{y}_1} : \top, \varphi_{z_1} : \neg z$ and $\varphi_\theta : ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \bar{y}_1)) \wedge (y_1 \vee \bar{y}_1)$. We show that there is a strongly admissible interpretation v with $v(\theta) = \mathbf{t}$ and $|S| = |Y| + 1$ where $S = v^{\mathbf{t}} \cup v^{\mathbf{f}}$ iff Θ is a valid formula.

- Assume that Θ is a valid formula. We show that there exists a strongly admissible interpretation v with $|S| = |Y| + 1$. Since Θ is a valid formula, there exists an interpretation I_Y of Y such that for any interpretation I_Z of Z, it holds that $I_Y \cup I_Z \models \theta(Y, Z)$, i.e., θ is true. Specifically, it holds that $I_Y \models \theta(Y, Z)$.
  We define a three-valued interpretation v of A such that $v(y) = \mathbf{t}$ if $I_Y(y) = \mathbf{t}$, $v(\bar{y}) = \mathbf{t}$ if $I_Y(y) = \mathbf{f}$, $v(\theta) = \mathbf{t}$, and $v(x) = \mathbf{u}$ otherwise. It is easy to check that v is a strongly admissible interpretation of D where $|S| = |Y| + 1$. Thus, θ is strongly acceptable in a strongly admissible interpretation v where $|S| = |Y| + 1$.
- Let v be the strongly admissible interpretation with $v(\theta) = \mathbf{t}$ and $|S| \leq |Y| + 1$. Let g be the unique grounded interpretation of D. It holds that $v \leq_i g$. For each $z \in Z$, since $c_z : \neg z$, it is clear that $v(z) = \mathbf{u}$ in any strongly admissible interpretation v of D. Moreover, because θ is of the form $\psi \wedge \bigwedge_{y \in Y}(y \vee \neg y)[\neg y / \bar{y}]$, we have that for each $y \in Y$ either $v(y) = \mathbf{t}$ or $v(\bar{y}) = \mathbf{t}$ and thus $|S| = |Y| + 1$. Because of this, we also have that not both $v(y) = \mathbf{t}$ or $v(\bar{y}) = \mathbf{t}$ can be simultaneously true. We can thus define the following interpretation I_Y of Y such that $I_Y(y) = \mathbf{t}$ if $v(y) = \mathbf{t}$ and $I_Y(y) = \mathbf{f}$ if $v(\bar{y}) = \mathbf{t}$. Since θ is strongly accepted with respect to v, we have that for each interpretation I_Z of Z, the formula θ is satisfied by $I_Y \cup I_Z$. That is, the QBF Θ is valid.

□

	Cred$_{sadm}$	*Skept$_{sadm}$*	*Ver$_{sadm}$*	*StrJust*	*k-Witness$_{sadm}$*
AFs	P	trivial	P	n.a.	NP-c
ADFs	coNP-c	trivial	coNP-c	coNP-c	Σ_2^{P}-c

Table 1: Complexity under the strong admissibility semantics of AFs and ADFs ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$)

Theorem 7 is a direct result of Lemmas 10 and 11.

Theorem 7. *k-Witness$_{sadm}$* is Σ_2^{P}-complete.

In Table 1, we summarize our results on the complexity of strong admissibility semantics in ADFs and compare them with the corresponding results for AFs (Caminada and Dunne 2020; Dvořák and Wallner 2020).

4 Conclusion

We studied the computational properties of the strong admissibility semantics of ADFs. When compared to AFs, computational complexity for ADFs increases by one step in the polynomial hierarchy (Stockmeyer 1976) for nearly all reasoning tasks (Strass and Wallner 2015; Dvořák and Dunne 2018). We have shown that, similarly, ADFs have higher computational complexity under the strong admissibility semantics when compared to AFs (Table 1).

From a theoretical perspective we observe that: 1. The credulous decision problem under the strong admissibility semantics of ADFs is coNP-complete, while this decision problem is tractable in AFs. 2. Since the trivial interpretation is the least strongly admissible interpretation for each ADF, the skeptical decision problem is trivial, which is similar for AFs. 3. The verification problem for ADFs is coNP-complete, while it is tractable for AFs. 4. Since an argument can be strongly justified in an interpretation that is not a strongly admissible interpretation, we defined a new reasoning task in Section 3.3, called the strong justification problem. The complexity of this decision problem, which investigates whether a queried argument is strongly justified in a given interpretation, is coNP-complete. 5. The problem of finding a smallest witness of strong justification of an argument investigates whether there exists a strongly admissible interpretation that assigns a minimum number of arguments to $\mathbf{t}/\mathbf{f}$ and satisfies a queried argument is Σ_2^{P}-complete, while this reasoning task is NP-complete for AFs.

We next highlight an interesting difference in the complexity landscapes of AFs and ADFs. When relating the complexity of grounded and strong admissibility semantics, we have that for AFs the verification problems can be (log-space) reduced to each other, while for ADFs there is a gap between the coNP-complete Ver_{sadm} problem and the DP-complete Ver_{grd} problem. That is, on the ADF level the step of proving arguments to be $\mathbf{u}$ in the grounded interpretation adds an NP part to the complexity; a similar effect can be observed for admissible and complete semantics.

As future work, it would be interesting to analyse the computational complexity of the current reasoning tasks for strong admissibility semantics over subclasses of ADFs, in particular bipolar ADFs (Brewka and Woltran 2010) and acyclic ADFs (Diller et al. 2020).

References

Arora, S., and Barak, B. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI Magazine* 38(3):25–36.

Baroni, P., and Giacomin, M. 2007. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence* 171(10-15):675–700.

Baroni, P.; Gabbay, D. M.; Giacomin, M.; and van der Torre, L. 2018. *Handbook of Formal Argumentation*. College Publications, London.

Baroni, P.; Caminada, M.; and Giacomin, M. 2011. An introduction to argumentation semantics. *Knowledge Engineering Review* 26(4):365–410.

Bench-Capon, T. J. M., and Dunne, P. E. 2007. Argumentation in artificial intelligence. *Artificial Intelligence* 171(10-15):619–641.

Booth, R.; Caminada, M.; and Marshall, B. 2018. DISCO: A web-based implementation of discussion games for grounded and preferred semantics. In Modgil, S.; Budzynska, K.; and Lawrence, J., eds., *Proceedings of Computational Models of Argument COMMA*, 453–454. IOS Press, Amsterdam.

Brewka, G., and Woltran, S. 2010. Abstract dialectical frameworks. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, 102–111.

Brewka, G.; Strass, H.; Ellmauthaler, S.; Wallner, J. P.; and Woltran, S. 2013. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 803–809.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2017. Abstract dialectical frameworks. An overview. *IFCoLog Journal of Logics and their Applications (FLAP)* 4(8).

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2018. Abstract dialectical frameworks: An overview. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications, London. 237–285.

Caminada, M., and Dunne, P. E. 2019. Strong admissibility revisited: Theory and applications. *Argument & Computation* 10(3):277–300.

Caminada, M., and Dunne, P. E. 2020. Minimal strong admissibility: A complexity analysis. In *Proceedings of Computational Models of Argument COMMA*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, 135–146. IOS Press, Amsterdam.

Caminada, M., and Uebis, S. 2020. An implementation of argument-based discussion using ASPIC-. In *COMMA*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, 455–456. IOS Press, Amsterdam.

Caminada, M. 2014. Strong admissibility revisited. In *Proceedings of Computational Models of Argument COMMA*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, 197–208. IOS Press, Amsterdam.

Caminada, M. 2018. Argumentation semantics as formal discussion. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications, London. 487–518.

Diller, M.; Zafarghandi, A. K.; Linsbichler, T.; and Woltran, S. 2020. Investigating subclasses of abstract dialectical frameworks. *Argument & Computation* 11(1):191–219.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Dvořák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications, London. 631–687.

Dvořák, W., and Wallner, J. P. 2020. Computing strongly admissible sets. In *Proceedings of Computational Models of Argument COMMA 2020*, 179–190. IOS Press, Amsterdam.

Keshavarzi Zafarghandi, A.; Verbrugge, R.; and Verheij, B. 2021a. Strong admissibility for abstract dialectical frameworks. In *The 36th ACM/SIGAPP Symposium on Applied Computing SAC '21*.

Keshavarzi Zafarghandi, A.; Verbrugge, R.; and Verheij, B. 2021b. Strong admissibility for abstract dialectical frameworks. *Argument & Computation* (Under revision).

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theoretical Computer Science* 3(1):1–22.

Strass, H., and Wallner, J. P. 2015. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence* 226:34–74.

van Eemeren, F. H.; Garssen, B.; Krabbe, E. C. W.; Henkemans, A. F. S.; Verheij, B.; and Wagemans, J. H. M., eds. 2014. *Handbook of Argumentation Theory*. Springer, Berlin.

Wallner, J. P. 2014. *Complexity Results and Algorithms for Argumentation: Dung's Frameworks and Beyond*. Ph.D. Dissertation, Vienna University of Technology, Institute of Information Systems.

Interlinking Logic Programs and Argumentation Frameworks

Chiaki Sakama[1] , **Tran Cao Son**[2]

[1]Wakayama University, 930 Sakaedani, Wakayama 640-8510, Japan
[2]New Mexico State University, Las Cruces, NM 88003, USA
sakama@wakayama-u.ac.jp, tson@cs.nmsu.edu

Abstract

Logic programs (LPs) and *argumentation frameworks* (AFs) are two declarative KR formalisms used for different reasoning tasks. The purpose of this study is interlinking two different reasoning components. To this end, we introduce two frameworks: *LPAF* and *AFLP*. The former enables to use the result of argumentation in AF for reasoning in LP, while the latter enables to use the result of reasoning in LP for arguing in AF. These frameworks are extended to *bidirectional frameworks* in which AF and LP can exchange information with each other. We also investigate their connection to several general KR frameworks from the literature. The proposed framework shares a view similar to the *multi-context system* while its semantics is different from an equilibrium semantics.

1 Introduction

A *logic program* (LP) represents declarative knowledge as a set of rules and realizes commonsense reasoning as logical inference. An *argumentation framework* (AF), on the other hand, represents arguments and an attack relation over them, and defines which arguments are accepted or not under various semantics. Two frameworks specify different types of knowledge and realize different types of reasoning.

In our daily life, however, we often use two modes of reasoning interchangeably. For instance, consider a logic program representing knowledge:

$$\begin{aligned} \mathit{get\_vaccine} &\leftarrow \mathit{safe} \wedge \mathit{effective} \\ \neg \mathit{get\_vaccine} &\leftarrow \mathit{not\ safe} \end{aligned}$$

where we get a vaccine if it is safe and effective. To see whether a vaccine is safe and effective, we refer to an expert opinion. It is often the case, however, that multiple experts have different opinions. In this case, we observe argumentation among experts and take it into account to make a decision. For another example, consider a debate on whether global warming is occurring. Scientists and politicians make different claims based on evidences and scientific knowledge. An argumentation framework is used for representing the debate, while arguments appearing in the argumentation graph are generated as results of reasoning from the background knowledge of participants.

In these examples, we can encode reasoners' private knowledge as logic programs and argumentation in the public space as argumentation frameworks. It is natural to distinguish two different types of knowledge and interlink them with each other. In the first example, an agent has a private knowledge base that refers to a public argumentation framework. In the second example, on the other hand, agents participating in a debate have their private knowledge bases supporting their individual claims.

Logic programs and argumentation are mutually transformed with each other. Dung (1995) provides a transformation from LPs to AFs and shows that stable models (Gelfond and Lifschitz 1988) (resp. the well-founded model (Van Gelder, *et al*. 1991)) of a logic program correspond to stable extensions (resp. the grounded extension) of a transformed argumentation framework. He also introduces a converse transformation from AFs to LPs, and shows that the semantic correspondences still hold. The results are extended to equivalences of LPs and AFs under different semantics (e.g. (Caminada, *et al*. 2015)). Using such transformational approaches, an LP and an AF are combined and one could perform both argumentative reasoning and commonsense reasoning in a single framework. One of the limitations of this approach is that in order to combine an LP and an AF in a single framework, two frameworks must have the corresponding semantics. For instance, suppose that an agent has a knowledge base *LP* and refers to an *AF*. If the agent uses the stable model semantics of *LP*, then *AF* must use the stable extension semantics to combine them into a single framework. Argumentation can have an internal structure in *structured argumentation*. In *assumption based argumentation* (ABA) (Dung *et al*. 2009), for instance, an argument for a claim c is supported by a set of assumptions S if c is deduced from S using a set of LP rules ($S \vdash c$). A structured argumentation has a knowledge base inside an argument and provides reasons that support particular claims. An argument is represented as a tree and an attack relation is introduced between trees. However, merging argumentation and knowledge bases into a single framework would produce a huge argumentation structure that is complicated and hard to manage.

In this paper, we introduce new frameworks, called *LPAF* and *AFLP*, for interlinking LPs and AFs. Each framework is defined as a collection of logic programs and argumentation frameworks. The *LPAF* uses the result of argumentation in AFs for reasoning in LPs. In contrast, the *AFLP* uses the result of reasoning in LPs for arguing in AFs. These frameworks are extended to *bidirectional frameworks* in which AFs and LPs can exchange information with each other. We address applications of the proposed framework and investigate connections to several KR frameworks. The rest of this paper is organized as follows. Section 2 reviews basic notions of logic programming and argumentation frameworks. Section 3 introduces several frameworks for interlinking LPs and AFs. Section 4 presents applications of the proposed frameworks. Section 5 discusses related issues and Section 6 summarizes the paper.

2 Preliminaries

We consider a language that contains a finite set $\mathscr{L}$ of propositional variables.

Definition 2.1 (logic program) A *(disjunctive) logic program* (LP) is a finite set of *rules* of the form:

$$p_1 \vee \cdots \vee p_l \leftarrow \; q_1, \ldots, q_m, \, not\, q_{m+1}, \ldots, not\, q_n$$

$(l,m,n \geq 0)$ where p_i and q_j are propositional variables (or ground atoms) in $\mathscr{L}$ and *not* is *negation as failure*.

The left-hand side of $\leftarrow$ is the *head* and the right-hand side is the *body*. For each rule r of the above form, $head(r)$, $body^+(r)$, and $body^-(r)$ respectively denote the sets of atoms $\{p_1,\ldots,p_l\}$, $\{q_1,\ldots,q_m\}$, and $\{q_{m+1},\ldots,q_n\}$, and $body(r) = body^+(r) \cup body^-(r)$. A *(disjunctive) fact* is a rule r such that $body(r) = \varnothing$. A fact is called a *non-disjunctive fact* if $l = 1$. If an LP contains no disjunctive rule (i.e., $l \leq 1$), it is called a *normal logic program*. Given a program LP, put $Head(LP) = \bigcup_{r \in LP} head(r)$ and $Body(LP) = \bigcup_{r \in LP} body(r)$. Throughout the paper, a program means a propositional/ground program.

Let $\mathscr{B}_{LP}$ be the set of ground atoms appearing in a program LP (called the *Herbrand base*). An interpretation $I \subseteq \mathscr{B}_{LP}$ *satisfies* a rule r if $body^+(r) \subseteq I$ and $body^-(r) \cap I = \varnothing$ imply $head(r) \cap I \neq \varnothing$. An interpretation satisfying every rule in a program is a *model* of the program. A model M of a program LP is *minimal* if there is no model N of LP such that $N \subset M$. The semantics of LP is defined as the set of designated models. Given a program LP, an interpretation I is a *stable model* of LP if it coincides with a minimal model of the program: $LP^I = \{\, p_1 \vee \cdots \vee p_l \leftarrow q_1,\ldots,q_m \mid (p_1 \vee \cdots \vee p_l \leftarrow q_1,\ldots,q_m, not\, q_{m+1},\ldots, not\, q_n) \in LP$ and $\{q_{m+1},\ldots,q_n\} \cap I = \varnothing \,\}$. A program may have no, one, or multiple stable models in general. The *stable model semantics* is defined as the set of stable models (Gelfond and Lifschitz 1988; Przymusinski 1990).

Generally, a logic program LP under the μ semantics is denoted by LP_μ. The semantics of LP_μ is defined as the set $\mathscr{M}^\mu_{LP}$ (or simply $\mathscr{M}^\mu$) of μ models of LP. If a ground atom p is included in every μ model of LP, we write $LP_\mu \models p$. LP_μ is simply written as LP if the semantics is clear in the context. A logic programming semantics μ is *universal* if every LP has a μ model. The stable model semantics is not universal, while the *well-founded semantics* of normal logic programs is universal.[1] A logic program LP under the stable model semantics (resp. well-founded semantics) is written as LP_{stb} (resp. LP_{wf}).

Definition 2.2 (argumentation framework) An *argumentation framework* (AF) is a pair (A,R) where $A \subseteq \mathscr{L}$ is a finite set of *arguments* and $R \subseteq A \times A$ is an *attack relation*.

For an AF (A,R), we say that an argument a *attacks* an argument b if $(a,b) \in R$. We write $a \rightarrow b$ iff $(a,b) \in R$. A set S of arguments *attacks* an argument a iff there is an argument $b \in S$ that attacks a. A set S of arguments is *conflict-free* if there are no arguments $a,b \in S$ such that a attacks b. A set S of arguments *defends* an argument a if S attacks every argument that attacks a. We write $D(S) = \{\, a \mid S \text{ defends } a \,\}$.

The semantics of AF is defined as the set of designated *extensions*. The following four extensions are introduced in (Dung 1995). Given $AF = (A,R)$, a conflict-free set of arguments $S \subseteq A$ is:

- a *complete extension* iff $S = D(S)$;
- a *stable extension* iff S attacks each argument in $A \setminus S$;
- a *preferred extension* iff S is a maximal complete extension of AF (wrt $\subseteq$);
- a *grounded extension* iff S is the minimal complete extension of AF (wrt $\subseteq$).

An argumentation framework AF under the ω semantics is denoted by AF_ω. The semantics of AF_ω is defined as the set $\mathscr{E}^\omega_{AF}$ (or simply $\mathscr{E}^\omega$) of ω extensions of AF. We abbreviate the above four semantics of AF as AF_{com}, AF_{stb}, AF_{prf} and AF_{grd}, respectively. AF_ω is simply written as AF if the semantics is clear in the context. Among the four semantics, the following relations hold: for any AF,

$$\mathscr{E}^{stb}_{AF} \subseteq \mathscr{E}^{prf}_{AF} \subseteq \mathscr{E}^{com}_{AF} \qquad \text{and} \qquad \mathscr{E}^{grd}_{AF} \subseteq \mathscr{E}^{com}_{AF}.$$

$\mathscr{E}^{stb}_{AF}$ is possibly empty, while others are not. In particular, $\mathscr{E}^{grd}_{AF}$ is a singleton set. An argumentation semantics ω is *universal* if every AF has an ω extension. The stable semantics is not universal, while the other three semantics presented above are universal.

3 Linking LP and AF

Throughout this section, LP is a logic program and $AF = (A,R)$ is an argumentation framework.

3.1 From AF to LP

We first introduce a framework that can use the result of argumentation in AFs for reasoning in LPs. In this subsection, we assume that $Head(LP) \cap A = \varnothing$, that is, no rule in LP has an argument in its head.

Definition 3.1 (refer) Given $AF = (A,R)$, LP is partitioned into $LP = R^{+A} \cup R^{-A}$ where

$$R^{+A} = \{\, r \in LP \mid body(r) \cap A \neq \varnothing \},$$
$$R^{-A} = \{\, r \in LP \mid body(r) \cap A = \varnothing \}.$$

[1] We refer the reader to (Van Gelder, *et al.* 1991) for the definition of the well-founded semantics.

We say that each rule in R^{+A} *refers to* arguments, and each rule in R^{-A} is *free from* arguments. An argument $a \in A$ is *referred by LP* if a appears in LP. Define $LP\,|_A = \{\, a \in A \mid a \text{ is referred by } LP \,\}$.

Definition 3.2 (μ model extended by $\mathscr{A}$) Let $AF = (A, R)$ and $\mathscr{A} \subseteq 2^A$. Then a μ *model of LP extended by* $\mathscr{A}$ is a μ model of $LP \cup \{a \leftarrow \mid a \in E \cap LP\,|_A\}$ for any $E \in \mathscr{A}$ if $\mathscr{A} \neq \varnothing$; otherwise, it is a μ model of $R^{-A} \subseteq LP$.

Definition 3.3 (simple LPAF) A *simple LPAF framework* is defined as a pair $\langle LP_\mu, AF_\omega \rangle$, where LP_μ is a logic program under the μ semantics and AF_ω is an argumentation framework under the ω semantics.

Definition 3.4 (LPAF model) Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be a simple LPAF framework. Suppose that AF has the set of ω extensions: $\mathscr{E}^\omega = \{E_1, \ldots, E_k\}$ $(k \geq 0)$. Then an *LPAF model* of φ is defined as a μ model of LP_μ extended by $\mathscr{E}^\omega$. The set of LPAF models of φ is denoted as $\mathbf{M}_\varphi$.

By definition, an LPAF model is defined as a μ model of the program LP by introducing referred arguments acceptable under the ω semantics from the AF part. If the AF part has no ω extension ($\mathscr{E}^\omega = \varnothing$), on the other hand, the AF part provides no justification for arguments referred by LP. In this case, we do not take the consequences that are derived using arguments in AF. Then an LPAF model is constructed by rules that are free from arguments in AF.

Example 3.1 Consider $\varphi_1 = \langle LP_{stb},\, AF_{stb} \rangle$ where

- $LP_{stb} = \{\, p \leftarrow a, \quad q \leftarrow not\, a \,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$.

As AF_{stb} has two stable extensions $\{a\}$ and $\{b\}$, φ_1 has two LPAF models $\{p, a\}$ and $\{q\}$. On the other hand, if we use $\omega = grounded$ then AF_{grd} has the single extension $\varnothing$. Then $\langle LP_{stb},\, AF_{grd} \rangle$ has the single LPAF model $\{q\}$.[2] Next, consider $\varphi_2 = \langle LP_{stb},\, AF_{stb} \rangle$ where

- $LP_{stb} = \{\, p \leftarrow not\, a, \quad q \leftarrow not\, p \,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (a, a)\})$.

As AF_{stb} has no stable extension and the second rule in LP_{stb} is free from arguments, φ_2 has the single LPAF model $\{q\}$. Note that if we keep the first rule then a different conclusion p is obtained from LP_{stb}. We do not consider the conclusion justified because the AF part provides no information on whether the argument a is acceptable or not.

Proposition 3.1 *Let* $\varphi_1 = \langle LP_\mu, AF^1_{\omega_1} \rangle$ *and* $\varphi_2 = \langle LP_\mu, AF^2_{\omega_2} \rangle$ *be two LPAFs such that* $\mathscr{E}^{\omega_1}_{AF^1} \neq \varnothing$. *If* $\mathscr{E}^{\omega_1}_{AF^1} \subseteq \mathscr{E}^{\omega_2}_{AF^2}$, *then* $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$.

Proof: When $\mathscr{E}^{\omega_1}_{AF^1} \neq \varnothing$, an LPAF model M of φ_1 is a μ model of $LP \cup \{a \leftarrow \mid a \in E \cap LP_\mu\,|_A\}$ for any $E \in \mathscr{E}^{\omega_1}_{AF^1}$. By $\mathscr{E}^{\omega_1}_{AF^1} \subseteq \mathscr{E}^{\omega_2}_{AF^2}$, $E \in \mathscr{E}^{\omega_2}_{AF^2}$. Then M is also an LPAF model of φ_2. □

Proposition 3.1 implies the inclusion relations with the same AF under different semantics: $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$ holds for $\varphi_1 = \langle LP_\mu, AF_{prf} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{com} \rangle$; $\varphi_1 = \langle LP_\mu, AF_{stb} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{prf} \rangle$; or $\varphi_1 = \langle LP_\mu, AF_{grd} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{com} \rangle$.

Two programs LP^1_μ and LP^2_μ are *uniformly equivalent relative to A* (denoted $LP^1_\mu \equiv^A_u LP^2_\mu$) if for any set of non-disjunctive facts $F \subseteq A$, the programs $LP^1_\mu \cup F$ and $LP^2_\mu \cup F$ have the same set of μ models (Eiter, *et al.* 2007).

Proposition 3.2 *Let* $\varphi_1 = \langle LP^1_\mu, AF_\omega \rangle$ *and* $\varphi_2 = \langle LP^2_\mu, AF_\omega \rangle$ *be two LPAFs such that* $\mathscr{E}^\omega \neq \varnothing$. *Then,* $\mathbf{M}_{\varphi_1} = \mathbf{M}_{\varphi_2}$ *if* $LP^1_\mu\,|_A = LP^2_\mu\,|_A$ *and* $LP^1_\mu \equiv^A_u LP^2_\mu$ *where* $AF_\omega = (A, R)$.

Proof: By $LP^1_\mu\,|_A = LP^2_\mu\,|_A$, $a \in A$ is referred by LP^1_μ iff a is referred by LP^2_μ. Then, for any $E \in \mathscr{E}^\omega$, $LP^1_\mu \cup \{a \leftarrow \mid a \in E \cap LP^1_\mu\,|_A\}$ and $LP^2_\mu \cup \{b \leftarrow \mid b \in E \cap LP^2_\mu\,|_A\}$ have the same set of μ models if $LP^1_\mu \equiv^A_u LP^2_\mu$. □

A simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$ is *consistent* if φ has an LPAF model. The consistency of φ depends on the chosen semantics μ. In particular, a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$ is consistent if μ is universal. $\varphi = \langle LP_\mu, AF_\omega \rangle$ may have an LPAF model even if $\mathscr{M}^\mu_{LP} = \mathscr{E}^\omega_{AF} = \varnothing$.

Example 3.2 Consider $\varphi = \langle LP_{stb}, AF_{stb} \rangle$ where

- $LP_{stb} = \{\, p \leftarrow not\, a, not\, p, \quad q \leftarrow \,\}$;
- $AF_{stb} = (\{a\}, \{(a, a)\})$.

Then $\mathscr{M}^{stb}_{LP} = \mathscr{E}^{stb}_{AF} = \varnothing$, but φ has the LPAF model $\{q\}$.

A simple LPAF consists of a single LP and an AF, which is generalized to a framework that consists of multiple LPs and AFs.

Definition 3.5 (general LPAF) A *general LPAF framework* is defined as a tuple

$$\langle LP^1_{\mu_1}, \ldots, LP^m_{\mu_m},\, AF^1_{\omega_1}, \ldots, AF^n_{\omega_n} \rangle$$

where each $LP^i_{\mu_i}$ $(1 \leq i \leq m)$ is a logic program LP^i under the μ_i semantics and each $AF^j_{\omega_j}$ $(1 \leq j \leq n)$ is an argumentation framework AF^j under the ω_j semantics.

A general LPAF framework is used in a situation such that multiple agents have individual LPs as their private knowledge bases and each agent possibly refers to open AFs. The semantics of a general LPAF is defined as an extension of a simple LPAF framework.

Definition 3.6 (LPAF state) Let $\varphi = \langle LP^1_{\mu_1}, \ldots, LP^m_{\mu_m},\, AF^1_{\omega_1}, \ldots, AF^n_{\omega_n} \rangle$ be a general LPAF framework. Then the *LPAF state* of φ is defined as a tuple $(\Sigma_1, \ldots, \Sigma_m)$ where $\Sigma_i = (\mathbf{M}^i_1, \ldots, \mathbf{M}^i_n)$ $(1 \leq i \leq m)$ and $\mathbf{M}^i_j$ $(1 \leq j \leq n)$ is the set of LPAF models of $\langle LP^i_{\mu_i}, AF^j_{\omega_j} \rangle$.

By definition, an LPAF state consists of a collection of LPAF models such that each model is obtained by combining a program $LP^i_{\mu_i}$ and an argumentation framework $AF^j_{\omega_j}$.

[2] Note that an AF extension represents whether an argument is accepted or not. If an argument a is not in an extension E, a is not accepted in E. Then $not\, a$ in LP becomes true by negation as failure.

Example 3.3 Consider $\varphi = \langle LP_{stb}, LP_{wf}, AF_{stb}, AF_{grd} \rangle$ where[3]

- $LP_{stb} = LP_{wf} = \{\, p \leftarrow a, not\, q,\;\; q \leftarrow a, not\, p \,\}$;
- $AF_{stb} = AF_{grd} = (\{a,b\}, \{(a,b),(b,a)\})$.

In this case,

- $\langle LP_{stb},\, AF_{stb} \rangle$ has three LPAF models: $\{p,a\}$, $\{q,a\}$ and $\varnothing$.
- $\langle LP_{stb},\, AF_{grd} \rangle$ has the single LPAF model: $\varnothing$.
- $\langle LP_{wf},\, AF_{stb} \rangle$ has two LPAF models: $\{a\}$ and $\varnothing$.
- $\langle LP_{wf},\, AF_{grd} \rangle$ has the single LPAF model: $\varnothing$.

Then φ has the LPAF state (Σ_1, Σ_2) where $\Sigma_1 = (\{\{p,a\},\{q,a\},\varnothing\},\{\varnothing\})$ and $\Sigma_2 = (\{\{a\},\varnothing\},\{\varnothing\})$.

The above example shows that a general LPAF is used for comparing the results of combination between LP and AF under different semantics.

Given tuples $(S_1,\ldots,S_k)$ and $(T_1,\ldots,T_l)$, define

$$(S_1,\ldots,S_k) \oplus (T_1,\ldots,T_l) = (S_1,\ldots,S_k,T_1,\ldots,T_l).$$

By definition, the following result holds.

Proposition 3.3 *Let $\varphi = \langle LP^1_{\mu_1},\ldots,LP^m_{\mu_m},\, AF^1_{\omega_1},\ldots,AF^n_{\omega_n} \rangle$ be a general LPAF framework. Then the LPAF state $(\Sigma_1,\ldots,\Sigma_m)$ of φ is obtained by $(\Sigma_1,\ldots,\Sigma_k) \oplus (\Sigma_{k+1},\ldots,\Sigma_m)$ $(1 \le k \le m-1)$ where $(\Sigma_1,\ldots,\Sigma_k)$ is the LPAF state of $\varphi_1 = \langle LP^1_{\mu_1},\ldots,LP^k_{\mu_k},\, AF^1_{\omega_1},\ldots,AF^n_{\omega_n} \rangle$ and $(\Sigma_{k+1},\ldots,\Sigma_m)$ is the LPAF state of $\varphi_2 = \langle LP^{k+1}_{\mu_{k+1}},\ldots,LP^m_{\mu_m},\, AF^1_{\omega_1},\ldots,AF^n_{\omega_n} \rangle$.*

Proposition 3.3 presents that a general LPAF has the modularity property; φ is partitioned into smaller φ_1 and φ_2, and the introduction of new LPs to φ is done incrementally.

3.2 From LP to AF

We next introduce a framework that can use the result of reasoning in LPs for arguing in AFs. In this subsection, we assume that $Body(LP) \cap A = \varnothing$, that is, no rule in LP has an argument in its body.

Definition 3.7 (AF with support) Let $AF = (A,R)$ and $M \subseteq \mathscr{B}_{LP}$. Then *AF with support M* is defined as $AF^M = (A,R')$ where $R' = R \setminus \{\, (x,a) \mid x \in A \text{ and } a \in A \cap M \,\}$.

By definition, AF^M is an argumentation framework in which every tuple attacking $a \in M$ is removed from R. As a result, every argument included in M is accepted in AF^M.

Definition 3.8 (ω extension supported by $\mathscr{M}$) Let $AF = (A,R)$ and $\mathscr{M} \subseteq 2^{\mathscr{B}_{LP}}$. Then an *$\omega$ extension of AF supported by $\mathscr{M}$* is an ω extension of AF^M for any $M \in \mathscr{M}$ if $\mathscr{M} \neq \varnothing$; otherwise, it is an ω extension of (A',R') where $A' = A \setminus \mathscr{B}_{LP}$ and $R' = R \cap (A' \times A')$.

Definition 3.9 (simple AFLP) A *simple AFLP framework* is defined as a pair $\langle AF_\omega, LP_\mu \rangle$ where AF_ω is an argumentation framework under the ω semantics and LP_μ is a logic program under the μ semantics.

Definition 3.10 (AFLP extension) Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be a simple AFLP framework. Suppose that LP has the set of μ models $\mathscr{M}^\mu$. Then an *AFLP extension* of ψ is defined as an ω extension of AF_ω supported by $\mathscr{M}^\mu$. The set of AFLP extensions of ψ is denoted as $\mathbf{E}_\psi$.

By definition, an AFLP extension is defined as an ω extension of AF^M_ω that takes into account support information in a μ model M from the LP part. If the LP part has no μ model ($\mathscr{M}^\mu = \varnothing$), on the other hand, the LP part provides no ground for arguments in $A \cap \mathscr{B}_{LP}$. In this case, we do not use those arguments that rely on LP. Then an AFLP extension is constructed using arguments that do not appear in LP.

Example 3.4 Consider $\psi_1 = \langle AF_{stb}, LP_{stb} \rangle$ where

- $AF_{stb} = (\{a,b\}, \{(a,b),(b,a)\})$;
- $LP_{stb} = \{\, a \leftarrow p,\;\; p \leftarrow not\, q,\;\; q \leftarrow not\, p \,\}$.

LP_{stb} has two stable models $M_1 = \{a,p\}$ and $M_2 = \{q\}$, then $AF^{M_1}_\omega = (\{a,b\},\{(a,b)\})$ and $AF^{M_2}_\omega = AF_\omega$. As a result, ψ_1 has two AFLP extensions $\{a\}$ and $\{b\}$. On the other hand, if we use $\omega = grounded$, then $\langle AF_{grd}, LP_{stb} \rangle$ has two AFLP extensions $\{a\}$ and $\varnothing$. Next, consider $\psi_2 = \langle AF_{grd}, LP_{stb} \rangle$ where

- $AF_{grd} = (\{a,b,c\}, \{(a,b),(b,c)\})$;
- $LP_{stb} = \{\, a \leftarrow p,\;\; p \leftarrow not\, p \,\}$.

As LP_{stb} has no stable model, ψ_2 has the AFLP extension $\{b\}$ as the grounded extension of $(\{b,c\},\{(b,c)\})$.

Proposition 3.4 *Let $\psi_1 = \langle AF_\omega, LP^1_{\mu_1} \rangle$ and $\psi_2 = \langle AF_\omega, LP^2_{\mu_2} \rangle$ be two AFLPs such that $\mathscr{M}^{\mu_1}_{LP^1} \neq \varnothing$. If $\mathscr{M}^{\mu_1}_{LP^1} \subseteq \mathscr{M}^{\mu_2}_{LP^2}$, then $\mathbf{E}_{\psi_1} \subseteq \mathbf{E}_{\psi_2}$.*

Proof: When $\mathscr{M}^{\mu_1}_{LP^1} \neq \varnothing$, an AFLP extension E of ψ_1 is an ω extension of AF^M_ω for any $M \in \mathscr{M}^{\mu_1}_{LP^1}$. By $\mathscr{M}^{\mu_1}_{LP^1} \subseteq \mathscr{M}^{\mu_2}_{LP^2}$, $M \in \mathscr{M}^{\mu_2}_{LP^2}$. Then E is also an AFLP extension of ψ_2. □

Baumann (2014) introduces equivalence relations of AFs with respect to deletion of arguments and attacks. For two $AF^1_\omega = (A_1,R_1)$ and $AF^2_\omega = (A_2,R_2)$,

- AF^1_ω and AF^2_ω are *normal deletion equivalent* (denoted $AF^1_\omega \equiv_{nd} AF^2_\omega$) if for any set A of arguments $(A'_1, R_1 \cap (A'_1 \times A'_1))$ and $(A'_2, R_2 \cap (A'_2 \times A'_2))$ have the same set of ω extensions where $A'_1 = A_1 \setminus A$ and $A'_2 = A_2 \setminus A$.
- AF^1_ω and AF^2_ω are *local deletion equivalent* (denoted $AF^1_\omega \equiv_{ld} AF^2_\omega$) if for any set R of attacks $(A_1, R_1 \setminus R)$ and $(A_2, R_2 \setminus R)$ have the same set of ω extensions.

By definition, we have the next result.

Proposition 3.5 *Let $\psi_1 = \langle AF^1_\omega, LP_\mu \rangle$ and $\psi_2 = \langle AF^2_\omega, LP_\mu \rangle$ be two AFLPs. Then*

- *When $\mathscr{M}^\mu = \varnothing$, $\mathbf{E}_{\psi_1} = \mathbf{E}_{\psi_2}$ if $AF^1_\omega \equiv_{nd} AF^2_\omega$.*
- *When $\mathscr{M}^\mu \neq \varnothing$, $\mathbf{E}_{\psi_1} = \mathbf{E}_{\psi_2}$ if $AF^1_\omega \equiv_{ld} AF^2_\omega$.*

Baumann (2014) shows that $AF^1_\omega \equiv_{ld} AF^2_\omega$ if and only if $AF^1_\omega = AF^2_\omega$ for any $\omega = \{com, stb, prf, grd\}$. On the other hand, necessary or sufficient conditions for $AF^1_\omega \equiv_{nd} AF^2_\omega$

[3] We consider the well-founded model as the set of true atoms under the well-founded semantics.

are given by the structure of argumentation graphs and they differ from the chosen semantics in general.

A simple AFLP framework $\psi = \langle AF_{\omega}, LP_{\mu} \rangle$ is *consistent* if ψ has an AFLP extension. By definition, a simple AFLP framework $\psi = \langle AF_{\omega}, LP_{\mu} \rangle$ is consistent if ω is universal.

A simple AFLP consists of a single AF and an LP, which is generalized to a framework that consists of multiple AFs and LPs.

Definition 3.11 (general AFLP) A *general AFLP framework* is defined as a tuple

$$\langle AF^1_{\omega_1}, \ldots, AF^n_{\omega_n},\; LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$$

where each $AF^j_{\omega_j}$ $(1 \leq j \leq n)$ is an argumentation framework AF^j under the ω_j semantics and each $LP^i_{\mu_i}$ $(1 \leq i \leq m)$ is a logic program LP^i under the μ_i semantics.

A general AFLP framework is used in a situation such that argumentative dialogues consult LPs as information sources. The semantics of a general AFLP is defined as an extension of a simple AFLP framework.

Definition 3.12 (AFLP state) Let $\psi = \langle AF^1_{\omega_1}, \ldots, AF^n_{\omega_n},\; LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$ be a general AFLP framework. Then an *AFLP state* of ψ is defined as a tuple $(\Gamma_1, \ldots, \Gamma_n)$ where $\Gamma_j = (\mathbf{E}^j_1, \ldots, \mathbf{E}^j_m)$ $(1 \leq j \leq n)$ and $\mathbf{E}^j_i$ $(1 \leq i \leq m)$ is the set of AFLP extensions of $\langle AF^j_{\omega_j}, LP^i_{\mu_i} \rangle$.

By definition, an AFLP state consists of a collection of AFLP extensions such that each extension is obtained by combining $AF^j_{\omega_j}$ and $LP^i_{\mu_i}$.

Example 3.5 Consider $\psi = \langle AF_{grd},\, LP^1_{stb},\, LP^2_{stb} \rangle$ where

- $AF_{grd} = (\{a,b\}, \{(a,b)\})$;
- $LP^1_{stb} = \{\, a \leftarrow p,\;\; p \leftarrow \}$;
- $LP^2_{stb} = \{\, b \leftarrow q,\;\; q \leftarrow \}$.

Then, $\langle AF_{grd}, LP^1_{stb} \rangle$ has the AFLP extension $\{a\}$, while $\langle AF_{grd}, LP^2_{stb} \rangle$ has the AFLP extension $\{a,b\}$. Then the AFLP state of ψ is $((\{a\}, \{a,b\}))$.

A general AFLP has the modularity property.

Proposition 3.6 *Let* $\psi = \langle AF^1_{\omega_1}, \ldots, AF^n_{\omega_n},\; LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$ *be a general AFLP framework. Then the AFLP state* $(\Gamma_1, \ldots, \Gamma_n)$ *of* ψ *is obtained by* $(\Gamma_1, \ldots, \Gamma_k) \oplus (\Gamma_{k+1}, \ldots, \Gamma_n)$ $(1 \leq k \leq n-1)$ *where* $(\Gamma_1, \ldots, \Gamma_k)$ *is the AFLP state of* $\psi_1 = \langle AF^1_{\omega_1}, \ldots, AF^k_{\omega_k},\; LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$ *and* $(\Gamma_{k+1}, \ldots, \Gamma_n)$ *is the AFLP state of* $\psi_2 = \langle AF^{k+1}_{\omega_{k+1}}, \ldots, AF^n_{\omega_n},\; LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$*.*

3.3 Bidirectional Framework

In Sections 3.1 and 3.2 we provided frameworks in which given LPs and AFs one refers the other in one direction. This subsection provides a framework such that LPs and AFs interact with each other. Such a situation happens in social media, for instance, where a person posts his/her opinion to an Internet forum, which arises public discussion on the topic, then the person revises his/her belief by the result of discussion. In this subsection, we assume that any rule in LP could contain arguments in its head or body.

Definition 3.13 (simple bidirectional LPAF)
A *simple bidirectional LPAF framework* is defined as a pair $\langle\langle LP_{\mu}, AF_{\omega} \rangle\rangle$.

Definition 3.14 (BDLPAF model) Let $\zeta = \langle\langle LP_{\mu}, AF_{\omega} \rangle\rangle$ be a simple bidirectional LPAF framework. Suppose that a simple AFLP framework $\psi = \langle AF_{\omega}, LP_{\mu} \rangle$ has the set of AFLP extensions $\mathbf{E}_{\psi}$. Then a *BDLPAF model* of ζ is defined as a μ model of LP_{μ} extended by $\mathbf{E}_{\psi}$.

BDLPAF models reduce to LPAF models if $\mathbf{E}_{\psi}$ coincides with $\mathscr{E}^{\omega}_{AF}$. In the bidirectional framework, an LP can refer arguments in AF and AF can get a support from the LP.

Example 3.6 Consider $\zeta = \langle\langle LP_{stb},\, AF_{stb} \rangle\rangle$ where

- $LP_{stb} = \{\, a \leftarrow not\, p,\;\; q \leftarrow c \,\}$;
- $AF_{stb} = (\{a,b,c\}, \{(a,b), (b,a), (b,c)\})$.

First, the simple AFLP framework $\langle AF_{stb}, LP_{stb} \rangle$ has the single AFLP extension $E = \{a,c\}$. Then, the BDLPAF model of ζ becomes $\{a,c,q\}$.

Similarly, we can make a simple AFLP bidirectional.

Definition 3.15 (simple bidirectional AFLP)
A *simple bidirectional AFLP framework* is defined as a pair $\langle\langle AF_{\omega}, LP_{\mu} \rangle\rangle$.

Definition 3.16 (BDAFLP extension) Let $\eta = \langle\langle AF_{\omega}, LP_{\mu} \rangle\rangle$ be a simple bidirectional AFLP framework. Suppose that a simple LPAF framework $\varphi = \langle LP_{\mu}, AF_{\omega} \rangle$ has the set of LPAF models $\mathbf{M}_{\varphi}$. Then a *BDAFLP extension* of η is defined as an ω extension of AF_{ω} supported by $\mathbf{M}_{\varphi}$.

Example 3.7 Consider $\eta = \langle\langle AF_{grd},\, LP_{stb} \rangle\rangle$ where

- $AF_{grd} = (\{a,b\}, \{(a,b), (b,a)\})$;
- $LP_{stb} = \{\, p \leftarrow a,\;\; q \leftarrow not\, a,\;\; b \leftarrow q \,\}$.

First, the simple LPAF framework $\langle LP_{stb}, AF_{grd} \rangle$ has the single LPAF model $M = \{b,q\}$. Then, the BDAFLP extension of η becomes $\{b\}$.

Given AF_{ω} and LP_{μ}, a series of BDLPAF models or BDAFLP extensions can be build by repeatedly referring with each other. Starting with the AFLP extensions $\mathbf{E}^0_{\psi}$, the BDLPAF models $\mathbf{M}^1_{\varphi}$ extended by $\mathbf{E}^0_{\psi}$ are produced, then the BDAFLP extensions $\mathbf{E}^1_{\psi}$ supported by $\mathbf{M}^1_{\varphi}$ are produced, which in turn produce the BDLPAF models $\mathbf{M}^2_{\varphi}$ extended by $\mathbf{E}^1_{\psi}$, and so on. Likewise, starting with the LPAF models $\mathbf{M}^0_{\varphi}$, the sets $\mathbf{E}^1_{\psi}$, $\mathbf{M}^1_{\varphi}$, $\mathbf{E}^2_{\psi}$, …, are produced. We write the sequences of BDLPAF models and BDAFLP extensions as $[\mathbf{M}^1_{\varphi}, \mathbf{M}^2_{\varphi}, \ldots]$ and $[\mathbf{E}^1_{\psi}, \mathbf{E}^2_{\psi}, \ldots]$, respectively.

Proposition 3.7 *Let* $[\mathbf{M}^1_{\varphi}, \mathbf{M}^2_{\varphi}, \ldots]$ *and* $[\mathbf{E}^1_{\psi}, \mathbf{E}^2_{\psi}, \ldots]$ *be sequences defined as above. Then,* $\mathbf{M}^i_{\varphi} = \mathbf{M}^{i+1}_{\varphi}$ *and* $\mathbf{E}^j_{\psi} = \mathbf{E}^{j+1}_{\psi}$ *for some* $i, j \geq 1$*.*

Proof: If $\mathbf{M}^i_{\varphi} = \varnothing$ for every i or $\mathbf{E}^j_{\psi} = \varnothing$ for every j, the results hold by definition. Suppose that $\mathbf{E}^i_{\psi} \neq \varnothing$ and $\mathbf{M}^i_{\varphi} \neq \varnothing$ for some i. If a BDAFLP extension $E^i \in \mathbf{E}^i_{\psi}$ is supported by $M^i \in \mathbf{M}^i_{\varphi}$, then $M^i \cap A \subseteq E^i$. A BDLPAF model $M^{i+1} \in \mathbf{M}^{i+1}_{\varphi}$ is then constructed as a μ model of

$LP_\mu \cup \{a \leftarrow \mid a \in E^i \cap LP_\mu\,|_A\}$. Since $(E^i \cap LP_\mu\,|_A) \subseteq M^{i+1}$, it holds that $M^i \cap A \subseteq M^{i+1} \cap A$. As such, arguments in BDLPAF models increase monotonically. Since A is finite, $M^k \cap A = M^{k+1} \cap A$ for some $k (\geq 1)$. Then, arguments imported from AF do not change, and any BDLPAF model $M^k \in \mathbf{M}^k_\varphi$ is also a BDLPAF model $M^{k+1} \in \mathbf{M}^{k+1}_\varphi$. Hence, $\mathbf{M}^k_\varphi = \mathbf{M}^{k+1}_\varphi$. The result $\mathbf{E}^j_\psi = \mathbf{E}^{j+1}_\psi$ is shown in a similar way. □

4 Applications

This section provides applications of LPAF/AFLP to several KR frameworks.

4.1 Abduction

Abductive logic programming (ALP) (Kakas, *et al.* 1992) is a framework for realizing abduction in LP. An *abductive logic program* is a pair $\langle P, A \rangle$ where P is a logic program and A is a set of hypotheses called *abducibles*. In ALP abducibles are usually given as a set of ground atoms or also given as a set of rules (Inoue 2014). An LPAF framework $\langle LP_\mu, AF_\omega \rangle$ is considered as an extension of ALP in which potential consistent combinations of abductive hypotheses are given by argumentation frameworks.

Definition 4.1 (abductive LPAF) Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be a simple LPAF framework. Given a ground atom o as an *observation*, o is *explained* in φ if φ has an LPAF model M such that $o \in M$. In this case, the set $M \cap A$ is called a *(credulous) explanation* of o in φ where $AF_\omega = (A, R)$.

Scientific theories have been built through argumentation over hypotheses. An LPAF framework is used for characterizing the process by representing background knowledge LP_μ and scientific debates AF_ω.

Example 4.1 Suppose an LPAF $\varphi = \langle LP_{stb}, AF_{stb} \rangle$ in which AF_{stb} represents a debate between Geocentrism (Earth-centered) versus Heliocentrism (Sun-centered). It is represented by $AF_{stb} = (\{g,h\}, \{(g,h),(h,g)\})$ in its most condensed form where g represents Geocentrism and h represents Heliocentrism. Scientists believe that Venus shows phases like Moon only if it goes around the Sun. The knowledge is represented in LP_{stb} as

$$v \leftarrow h$$

where v represents “Venus shows phases”. In the 17th Century, Galileo Galilei found that Venus went through phases and concluded that Venus must travel around the Sun. The observation is represented as $o = v$ and φ has the LPAF model $\{v, h\}$ in which o is true. As a result, the observation is explained by the hypothesis Heliocentrism.

Abduction in LPAF is extended to general LPAF when there are multiple reasoners and multiple sources of hypotheses.

4.2 Deductive Argumentation

A *structured argumentation* is a framework such that there is an internal structure to an argument. In structured argumentation, knowledge is represented using a formal language and each argument is constructed from that knowledge. Given a logical language $\mathscr{L}$ and a consequence relation $\vdash$ in $\mathscr{L}$, a *deductive argument* (Besnard and Hunter 2014) is defined as a pair $\langle \mathscr{F}, c \rangle$ where $\mathscr{F}$ is a set of formulas in $\mathscr{L}$ and c is a (ground) atom such that $\mathscr{F} \vdash c$. $\mathscr{F}$ is called the *support* of the argument and c is the *claim*. A *counterargument* is an argument that attacks another argument. It is defined in terms of logical contradiction between the claim of a counterargument and the premises of the claim of an attacked argument.

An AFLP framework is captured as a kind of deductive arguments in the sense that LP can support an argument a appearing in AF. There is an important difference, however. In an AFLP, argumentative reasoning in AF and deductive reasoning in LP are separated. The AF part is kept at the abstract level and the LP part represents reasons for supporting particular arguments. In this sense, an AFLP provides a middle ground between abstract argumentation and structured argumentation. Such a separation keeps the whole structure compact and makes it easy to update AF or LP without changing the other part. Thus, AFLP/LPAF supports an elaboration tolerant development of knowledge bases.

With such a difference in mind, we characterize deductive argumentation in AFLP.

Definition 4.2 (support/rebut/undercut) Let $\psi = \langle AF^1_{\omega_1}, \ldots, AF^n_{\omega_n},\ LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$ be a general AFLP framework such that $AF^i_{\omega_i} = (A^i, R^i)$ $(1 \leq i \leq n)$.

- An argument $a \in A^i$ is *supported* in $LP^j_{\mu_j}$ for some $1 \leq j \leq m$ (written $(LP^j_{\mu_j}, a)$) if $LP^j_{\mu_j} \models a$.
- $(LP^j_{\mu_j}, a)$ and $(LP^k_{\mu_k}, b)$ *rebut* each other if $\{(a,b), (b,a)\} \subseteq R^i$ for some i.
- $(LP^j_{\mu_j}, a)$ *undercuts* $(LP^k_{\mu_k}, b)$ if $LP^k_{\mu_k} \cup \{a\} \not\models b$.

Example 4.2 (Besnard and Hunter 2014) (a) There is an argument that the government should cut spending because of a budget deficit. On the other hand, there is a counterargument that the government should not cut spending because the economy is weak. These arguments are respectively represented using deductive arguments as

$$A1 = \langle \{\mathit{deficit},\ \mathit{deficit} \rightarrow \mathit{cut}\},\ \mathit{cut} \rangle,$$
$$A2 = \langle \{\mathit{weak},\ \mathit{weak} \rightarrow \neg \mathit{cut}\},\ \neg \mathit{cut} \rangle$$

where $A1$ and $A2$ rebut each other. The situation is represented using the AFLP $\langle AF_{stb}, LP^1_{stb}, LP^2_{stb} \rangle$ such that

- $AF_{stb} = (\{\mathit{cut}, \mathit{no\text{-}cut}\},\ \{(\mathit{cut}, \mathit{no\text{-}cut}),\ (\mathit{no\text{-}cut}, \mathit{cut})\})$;
- $LP^1_{stb} = \{\mathit{cut} \leftarrow \mathit{deficit},\ \ \mathit{deficit} \leftarrow\}$;
- $LP^2_{stb} = \{\mathit{no\text{-}cut} \leftarrow \mathit{weak},\ \ \mathit{weak} \leftarrow\}$.

Then $(LP^1_{stb}, \mathit{cut})$ and $(LP^2_{stb}, \mathit{no\text{-}cut})$ rebut each other.

(b) There is an argument that the metro is an efficient form of transport, so one can use it. On the other hand, there is a counterargument that the metro is inefficient because of a strike. These arguments are respectively represented using deductive arguments as

$$A1 = \langle \{\mathit{efficient},\ \mathit{efficient} \rightarrow \mathit{use}\},\ \mathit{use} \rangle,$$
$$A2 = \langle \{\mathit{strike},\ \mathit{strike} \rightarrow \neg \mathit{efficient}\},\ \neg \mathit{efficient} \rangle$$

where $A2$ undercuts $A1$.

The situation is represented using an AFLP $\langle AF_{stb}, LP^1_{stb}, LP^2_{stb}\rangle$ such that

- $AF_{stb} = (\{\mathit{efficient}, \mathit{inefficient}\},$ $\{(\mathit{efficient}, \mathit{inefficient}), (\mathit{inefficient}, \mathit{efficient})\})$;
- $LP^1_{stb} = \{\, \mathit{use} \leftarrow \mathit{efficient},\ \ \mathit{efficient} \leftarrow \mathit{not\ inefficient} \,\}$;
- $LP^2_{stb} = \{\, \mathit{inefficient} \leftarrow \mathit{strike},\ \ \mathit{strike} \leftarrow \,\}$.

Then $(LP^2_{stb}, \mathit{inefficient})$ undercuts $(LP^1_{stb}, \mathit{use})$.

4.3 Answer Set Programming

Answer set programming (ASP) (Brewka *et al.* 2007) is a paradigm of declarative problem solving under the stable model semantics of logic programs. In principle, problem solving in ASP consists of two steps: firstly generate potential solutions then check whether they are in fact solutions. The first step requires combinatorial computation that is generally exponential, while the second step is usually polynomial. Then it is natural to seek the possibility of rewriting an ASP program into two components, the generation part and the verification part, and separating computational processes. Here we provide an example of realizing this using LPAF.

Example 4.3 The 3-coloring of a graph is a labelling of its vertexes with at most 3 colors such that no two vertexes sharing the same edge have the same color. This is a combinatorial search problem and it is represented in ASP as follows.

$$\begin{aligned}
&color(V,1) \lor color(V,2) \lor color(V,3) \leftarrow vertex(V)\\
&\leftarrow color(V,C), color(V,D), C \neq D\\
&\leftarrow color(V,C), color(W,C), edge(V,W)
\end{aligned}$$

together with a set of facts of *vertex* and *edge*. The first rule is the generation part and the second and the third rules are the verification part. The number of possible combinations exponentially grows by the increase of vertexes. For simplicity, we assume that there are 4 vertexes

$$vertex(a) \quad vertex(b) \quad vertex(c) \quad vertex(d)$$

and 5 edges

$$edge(a,b) \quad edge(b,c) \quad edge(c,d) \quad edge(d,a) \quad edge(b,d).$$

In this case, one of the solutions is given as:

$$color(a,1) \quad color(b,2) \quad color(c,1) \quad color(d,3).$$

The generation part is represented as the $AF_{stb} = (A, R)$ such that

$$\begin{aligned}
A &= \{\, a_i, b_i, c_i, d_i \mid i = 1,2,3 \,\},\\
R &= \{\, a_i \leftrightarrow a_j,\ b_i \leftrightarrow b_j,\ c_i \leftrightarrow c_j,\ d_i \leftrightarrow d_j \mid\\
&\qquad\qquad\qquad 1 \leq i, j \leq 3\ (i \neq j) \,\}
\end{aligned}$$

where a_i, b_i, c_i, d_i represent $color(a,i)$, $color(b,i)$, $color(c,i)$, $color(d,i)$, respectively. Then the 3-coloring problem is represented in LPAF as (LP_{stb}, AF_{stb}) where

$$LP_{stb} = \{\, \leftarrow x_i, y_i, edge(x_i, y_i) \mid x, y \in \{a,b,c,d\} \text{ and } x \neq y \,\}.$$

Generally, if a disjunctive logic program is *head-cycle-free*[4] it is transformed to a semantically equivalent normal logic program (Ben-Eliyahu and Dechter 1994). Then we can use existing techniques of encoding a normal logic program under the stable model semantics into an argumentation framework under the stable extension semantics (Dung 1995; Caminada, *et al.* 2015). Given a head-cycle-free disjunctive logic program LP, it is split into $LP = P \cup Q$ where

$$P = \{\, r \mid head(r) \neq \varnothing \,\} \text{ and } Q = \{\, r \mid head(r) = \varnothing \,\}.$$

P is then transformed to a semantically equivalent normal logic program $n(P)$. Let $AF^{n(P)}_{stb}$ be an AF encoding of $n(P)$ under the stable model/extension semantics.

Proposition 4.1 *Let $LP = P \cup Q$ be a head-cycle-free disjunctive logic program. Then S is an answer set of LP iff S is an LPAF model of the LPAF framework $\varphi = (Q_{stb}, AF^{n(P)}_{stb})$.*

4.4 Argument Aggregation

Argument aggregation or *collective argumentation* (Bodanza, *et al.* 2017) considers a situation in which multiple agents may have different arguments and/or opinions. The problems are then what and how to aggregate arguments. In abstract argumentation, the problem is formulated as follows. Given several AFs having different arguments and attacks, find acceptable arguments among those AFs. In the *argument-wise aggregation* individually supported arguments are aggregated by some voting mechanism.

Example 4.4 (Bodanza, *et al.* 2017) Suppose three agents deciding which among three arguments *a*, *b*, and *c*, are collectively acceptable. Each agent has a subjective evaluation of the interaction among those arguments, leading to three different individual AFs:

$$\begin{aligned}
AF_1 &= (\{a,b,c\}, \{(a,b), (b,c)\}),\\
AF_2 &= (\{a,b,c\}, \{(a,b)\}),\\
AF_3 &= (\{a,b,c\}, \{(b,c)\}).
\end{aligned}$$

Three AFs have the grounded extensions $\{a,c\}$, $\{a,c\}$, and $\{a,b\}$, respectively. By majority voting, $\{a,c\}$ is obtained as the collective extension.

In this example, however, how an agent performs a subjective evaluation is left as a blackbox. The situation is represented using a general AFLP as follows. Consider a general AFLP $\psi = \langle AF_{grd}, LP^1_{stb}, LP^2_{stb}, LP^3_{stb}\rangle$ with

$$\begin{aligned}
AF_{grd} &= (\{a,b,c\}, \{(a,b), (b,c)\}),\\
LP^1_{stb} &= \{\, p \leftarrow not\, q \,\},\\
LP^2_{stb} &= \{\, c \leftarrow p,\quad p \leftarrow \,\},\\
LP^3_{stb} &= \{\, b \leftarrow not\, q \,\}.
\end{aligned}$$

Then (AF_{grd}, LP^1_{stb}) has the AFLP extension $\{a,c\}$; (AF_{grd}, LP^2_{stb}) has the AFLP extension $\{a,c\}$; (AF_{grd}, LP^3_{stb})

[4] A disjunctive logic program LP is head-cycle-free if the (positive) dependency graph of LP contains no directed cycle that goes through two different atoms in the head of the same disjunctive rule in LP.

has the AFLP extension $\{a,b\}$. In this case, the AFLP state of ψ is (Γ) where $\Gamma = (\{\{a,c\}\}, \{\{a,c\}\}, \{\{a,b\}\})$. As such, three agents evaluate the common *AF* based on their private knowledge base, which results in three individual sets of extensions in the AFLP state. Observe that in this case, the private knowledge of the agents are related to p and q, and only the third agent is *influenced* by his private knowledge base in drawing the conclusions.

When multiple agents argue on the common AF, argument-wise aggregation is characterized using AFLP as follows. Suppose $\Gamma = (T_1, \ldots, T_k)$ with $T_i \subseteq 2^A$ where A is the set of arguments of AF. For any $E \subseteq A$, let $\mathscr{F}_\Gamma(E) = h$ where h is the number of occurrences of E in $T_1, \ldots, T_k$. Define $max\mathscr{F}_\Gamma = \{E \mid \mathscr{F}_\Gamma(E) \text{ is maximal}\}$.

Definition 4.3 (collective extension) Let $\psi = \langle AF_\omega, LP^1_{\mu_1}, \ldots, LP^m_{\mu_m} \rangle$ be a general AFLP that has the AFLP state (Γ) with $\Gamma = (T_1, \ldots, T_m)$. Then the *collective extension* by majority voting is any extension in $max\mathscr{F}_\Gamma$.

Applying it to the above example, $max\mathscr{F}_\Gamma = \{\{a,c\}\}$. In Definition 4.3, if there is $E \subseteq A$ such that $\mathscr{F}_\Gamma(E) = m$, then E is included in every T_i $(1 \leq i \leq m)$. In this case, all agents *agree on* E.

4.5 Multi-Context System

Heterogeneous non-monotonic multi-context system (MCS) has been introduced as a general formalism for integrating heterogeneous knowledge bases (Brewka and Eiter 2007). An MCS $M = (C_1, \ldots, C_n)$ consists of contexts $C_i = (L_i, kb_i, br_i)$ $(1 \leq i \leq n)$, where $L_i = (KB_i, BS_i, ACC_i)$ is a logic, $kb_i \in KB_i$ is a knowledge base of L_i, BS_i is the set of possible belief sets, $ACC_i : KB_i \mapsto 2^{BS_i}$ is a semantic function of L_i, and br_i is a set of L_i-bridge rules of the form:

$$s \leftarrow (c_1{:}p_1), \ldots, (c_j{:}p_j),\ not\ (c_{j+1}{:}p_{j+1}), \ldots,\ not\ (c_m{:}p_m)$$

where, for each $1 \leq k \leq m$, we have that: $1 \leq c_k \leq n$, p_k is an element of some belief set of L_{c_k}, and $kb_i \cup \{s\} \in KB_i$. Intuitively, a bridge rule allows us to add s to a context, depending on the beliefs in the other contexts. Given a rule r of the above form, we denote $head(r) = s$. The semantics of MCS is described by the notion of belief states. Let $M = (C_1, \ldots, C_n)$ be an MCS. A *belief state* is a tuple $S = (S_1, \ldots, S_n)$ where each S_i is an element of BS_i.

Given a belief state $S = (S_1, \ldots, S_n)$ and a bridge rule r of the above form, we say that r is *applicable* in S if $p_l \in S_{c_l}$ for each $1 \leq l \leq j$ and $p_k \notin S_{c_k}$ for each $j+1 \leq k \leq m$. By $app(B,S)$ we denote the set of the bridge rules $r \in B$ that are applicable in S. A belief state $S = (S_1, \ldots, S_n)$ of M is an *equilibrium* if, for all $1 \leq i \leq n$, we have that $S_i \in ACC_i(kb_i \cup \{head(r) \mid r \in app(br_i, S)\})$.

Given an LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$, the *corresponding MCS* of φ is defined by $\varphi_{mcs} = (C_1, C_2)$ where

- $C_1 = (L_1, LP_\mu, br_1)$ where L_1 is the logic of LP under the μ semantics and $br_1 = \{a \leftarrow (c_2 : a) \mid a \in LP|_A\}$.
- $C_2 = (L_2, AF_\omega, \varnothing)$ where L_2 is the logic of AF under the ω semantics.

Intuitively, the bridge rules transfer the acceptability of arguments in AF_ω to LP_μ.

Proposition 4.2 *Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be an LPAF framework and φ_{mcs} the* corresponding MCS *of φ. If AF_ω is consistent then (S_1, S_2) is an equilibrium of φ_{mcs} iff S_1 is an LPAF model of φ and S_2 is an ω extension of AF_ω.*

Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be an AFLP framework with $AF_\omega = (A, R)$. The *corresponding MCS* of ψ is defined by $\psi_{mcs} = (C_1, C_2)$ where

- $C_1 = (L_1, AF_\omega, br_1)$ where L_1 is the logic of AF under the ω semantics, and $br_1 = \{(y,x) \leftarrow (c_2 : a) \mid \exists a \exists x [a \in A \cap \mathscr{B}_{LP} \text{ and } (x,a) \in R]\}$ where y is a new argument.
- $C_2 = (L_2, LP_\mu, \varnothing)$ where L_2 is the logic of LP under the μ semantics.

As with LPAF, the bridge rules transfer the acceptability of arguments from LP_μ to AF_ω. We assume that new arguments and attacks introduced by the bridge rules br_1 are respectively added to the set of arguments and attacks of AF.

Proposition 4.3 *Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be an AFLP framework and ψ_{mcs} the* corresponding MCS *of ψ. If LP_μ is consistent then (S_1, S_2) is an equilibrium of ψ_{mcs} iff $S_1 \setminus Y$ is an AFLP extension of ψ and S_2 is a μ model of LP_μ, where Y is the set of new arguments introduced by br_1.*

A general LPAF $\varphi = \langle LP^1_{\mu_1}, \ldots, LP^m_{\mu_m}, AF^1_{\omega_1}, \ldots, AF^n_{\omega_n} \rangle$ can be viewed as a collection of MCS. Let C^j_i be the corresponding MCS of $\langle LP^i_{\mu_i}, AF^j_{\omega_j} \rangle$. It is easy to see that due to Proposition 4.2, $(C^1_i, \ldots, C^n_i)$ can be used to characterize the i-th element Σ_i of the LPAF state $(\Sigma_1, \ldots, \Sigma_m)$ of φ. A similar characterization of an AFLP state using MCS could be derived due to Proposition 4.3.

A simple LPAF/AFLP is captured as an MCS with a restriction of two systems (Propositions 4.2 and 4.3). However, φ_{mcs} (resp. ψ_{mcs}) is well-defined only if its submodule AF_ω (resp. LP_μ) is *consistent*. This is because an MCS assumes that each context is consistent. Moreover, a general LPAF/AFLP can handle LPs and AFs with different semantics in a single framework. As such, LPAF/AFLP shares a view similar to MCS while it is *different* from MCS.

4.6 Constrained Argumentation Frameworks

Constrained argumentation frameworks (CAF), proposed in (Coste-Marquis, *et al*. 2006), could be viewed as another attempt to extend AF with a logical component. A CAF is of the form $\langle A, R, C \rangle$ where (A, R) is an AF and C is a propositional formula over A. A set of arguments S satisfies C if $S \cup \{\neg a \mid a \in A \setminus S\} \models C$. For a semantics ω, an *ω C-extension* of $\langle A, R, C \rangle$ is an ω extension of (A, R) that satisfies C, i.e., the constraint C is used to eliminate undesirable extensions. Therefore, a CAF can be viewed as an LPAF (LP_μ, AF_ω) where AF_ω is the original argumentation framework of the CAF and LP_μ is used to verify the condition C.

Consider a CAF $\delta = \langle A, R, C \rangle$. For simplicity of the presentation, assume that C is in DNF. For $a \in A$, let na be a unique new atom associated with a, denoting that a is not acceptable. Let $\top$ be a special atom denoting *true*. Then, define the logic program $LP(C)$ as follows:

$$\begin{array}{rcl} LP(C) & = & \{\top \leftarrow l'_1,\ldots,l'_n \mid \text{a conjunct } l_1 \wedge \cdots \wedge l_n \text{ is in } C \\ & & \quad \text{and } l'_i = a \text{ if } l_i = a, \text{ and } l'_i = not\,a \text{ if } l_i = \neg a\} \\ & \cup & \{na \leftarrow not\,a, \quad \leftarrow a, na \mid a \in A\} \\ & \cup & \{\leftarrow not\,\top\}. \end{array}$$

We can easily verify that a set of arguments S satisfies C iff $S \cup \{na \mid a \in A \setminus S\} \cup \{\top\}$ is a stable model of $LP(C)$.

Proposition 4.4 *Let $\delta = \langle A,R,C \rangle$ be a CAF. Then, $(LP(C)_{stb}, AF_\omega)$ has an LPAF model M iff $M \setminus (\{na \mid a \in A\} \cup \{\top\})$ is an ω C-extension of δ.*

This highlights the flexibility of LPAF in that it can also be used to express preferences among extensions of AF.

5 Discussion

There is a number of studies that interrelates logic programming and argumentation frameworks. Caminada *et al.* (2015) introduce two different connections. First, given a logic program LP its associated argumentation framework AF_{LP} is defined. In AF_{LP} each rule is viewed as an argument A that has the conclusion in its head, and subarguments (as positive literals) and vulnerabilities (as negative literals) in its body. They show connections between a logic programming semantics of LP and a set of conclusions appearing in arguments of AF_{LP}. For instance, given the program LP:

$$p \leftarrow not\,q, \qquad q \leftarrow r, not\,p, \qquad r \leftarrow s, \qquad s \leftarrow$$

its associated AF_{LP} is defined as

$$\begin{array}{ll} A_1: \ s \leftarrow & \qquad A_2: \ r \leftarrow (A_1) \\ A_3: \ q \leftarrow (A_2), not\,p & \qquad A_4: \ p \leftarrow not\,q \end{array}$$

where $A_1 - A_4$ are arguments, and A_3 and A_4 mutually attack each other. AF_{LP} has two stable extensions: $\{A_1,A_2,A_3\}$ and $\{A_1,A_2,A_4\}$ where the sets of conclusions $\{q,s,r\}$ and $\{p,s,r\}$ are the stable models of P. On the other hand, AF_{LP} has the grounded extension $\{A_1,A_2\}$ where the set of conclusions $\{s,r\}$ is the well-founded model of LP.

Second, an argumentation framework is transformed to a logic program. Given an argumentation framework $AF = (A,R)$, its associated LP is defined as:

$$\begin{array}{rl} LP_{AF} = \{A \leftarrow not\,B_1,\ldots,not\,B_m \mid & A, B_1,\ldots,B_m \in A \\ & \text{and } \{B_i \mid (B_i,A) \in R\} = \{B_1,\ldots,B_m\}\}. \end{array}$$

Then there are 1-1 correspondence between argumentation semantics of AF and logic programming semantics of LP_{AF}. For instance, given $AF = (\{a,b,c\},\{(a,b),(b,a),(c,b),(c,c)\})$, LP_{AF} becomes $\{a \leftarrow not\,b, \quad b \leftarrow not\,a, not\,c, \quad c \leftarrow not\,c\}$. Then the stable extensions of AF are equivalent to the stable semantics of LP_{AF}, and the grounded extension of AF is equivalent to the well-founded semantics of LP_{AF}, etc.

AF_{LP} is similar to deductive argument discussed in Section 4.2. As argued there, however, the AFLP framework introduced in this paper separates argumentation and rules. Thus, it is different from AF_{LP} in which rules are parts of arguments. On the other hand, LP_{AF} translates AF into a program LP. Given a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$, AF_ω is translated into a logic program LP_{AF_ω}. Then two programs are combined as $\Pi = LP_\mu \cup LP_{AF_\omega}$. However, two programs have different semantics in general, and in this case we cannot handle Π as a single program. For instance, if $\mu = \omega = stable$ then the stable model semantics of Π is well-defined. However, if $\mu = stable$ and $\omega = grounded$ then the corresponding semantics of LP_{AF_ω} is the well-founded semantics, so we cannot combine two programs having different semantics. As such, an LPAF framework cannot straightforwardly be encoded in a single LP in general. It is often possible to convert a semantics to another semantics in the syntax level. For instance, supported models are encoded as answer sets using nested expressions (Lifschitz, *et al.* 1999), and partial stable models can be captured with standard stable models (Janhunen, *et al.* 2006). However, it requires an extra step to convert semantics from one to the other using program transformation. Moreover, it is unclear whether such conversion among different semantics is always possible for every LP or AF semantics. The proposed framework is much simpler because there is no need to merge two frameworks into one, and LP or AF can employ its own semantics independently with each other.

The complexity of LPAF/AFLP depends on the complexities of LP and AF. Let us consider the *model existence problem* of simple LPAF/AFLP frameworks, denoted with $\texttt{Exists}^M$, which is defined by "*given an LPAF/AFLP framework λ, determine whether λ has a model?*" For a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$, the existence of an LPAF model of φ depends on μ and ω. For example, if $\mu = well\text{-}founded$ and $\omega = grounded$ then φ has a unique LPAF model which can be computed in polynomial time (if LP is a normal logic program); on the other hand, $\mu = stable$ and $\omega = stable$ then the existence of an LPAF model of φ is not guaranteed.

Let C_μ and C_ω be the complexity classes of LP_μ and AF_ω in the polynomial hierarchy, respectively, and $\max(C_\mu, C_\omega)$ the higher complexity class among C_μ and C_ω. It is easy to see that the model existence problem of a simple LPAF belongs to the complexity class $\max(C_\mu, C_\omega)$. Intuitively, this follows from the observation that we can guess a pair (X,Y) and check whether Y is an ω extension of AF_ω and X is a μ model of $LP_\mu \cup \{a \leftarrow \mid a \in Y \cap LP|_A\}$. A similar argument is done for a simple AFLP framework. As an example, the existence of a stable model of a propositional disjunctive LP is in Σ_2^P (Eiter, *et al.* 1998) while the existence of extensions in AF is generally in NP or trivial, then $\texttt{Exists}^M$ for LPAF/AFLP involving $\mu = stable$ is in Σ_2^P where ω is one of the semantics of AF considered in this paper. Other semantics of AF (e.g. semi-stable, ideal, etc.) or LP (e.g. supported, possible models, etc.) can be easily adapted.

The model existence problem of simple LPAF/AFLP can be generalized to the state existence problem of general LPAF/AFLP frameworks, and it can be shown that it is the highest complexity class among all complexity classes involved in the general framework. Similar arguments can be used to determine the complexity class of credulous or skeptical reasoning in LPAF/AFLP. For example, the credulous entailment in LPAF, i.e., checking whether an atom a belongs to an LPAF model of $\varphi = \langle LP_{stb}, AF_\omega \rangle$ is also Σ_2^P. We omit the discussion for space limitation.

In the current frameworks, LP imports ω extensions from AF in LPAF, while AF imports μ models from LP in AFLP. We can also consider frameworks such that LP (resp. AF) imports *skeptical/credulous consequences* from AF_ω (resp. LP_μ). Such frameworks are realized by importing the intersection/union of ω extensions of AF to LP, or importing the intersection/union of μ models of LP to AF. In this paper we considered extension based semantics of AF. If we consider the *labelling based semantics* of AF, on the other hand, each argument has three different justification states, *in*, *out*, or *undecided*. In this case, selecting a 3-valued semantics of logic programs, LPAF/AFLP is defined in a similar manner.

6 Conclusion

We introduced several frameworks for interlinking LP and AF. LPAF and AFLP enable to combine different reasoning tasks while keeping independence of each knowledge representation. The potential of the proposed framework is shown by several applications to existing KR frameworks. LPAF or AFLP are realized by linking solvers of LP and AF.

LP and AF are two declarative KR frameworks and several studies have attempted to integrate them–translating from LP to AF and vice-versa, or incorporating rule bases into an AF in the context of structured argumentation. An approach taken in this paper is completely different from those approaches. We do not merge LP and AF while interlinking two components in different manners. Separation of two frameworks has an advantage of flexibility in dynamic environments, and several LPs and AFs are freely combined in general LPAF/AFLP frameworks. In addition, it supports an elaboration tolerant use of various knowledge representation frameworks.

The current framework can be further extended and applied in several ways. For instance, we can extend it to allow a single LP/AF to refer multiple AFs/LPs. If AF_ω is coupled with a *probabilistic logic program* LP_μ, an AFLP (AF_ω, LP_μ) could be used for computing probabilities of arguments in LP_μ and realizing *probabilistic argumentation* in AF_ω (Hunter 2013). As such, the proposed framework has potential for rich applications in AI.

References

Baumann, R. 2014. Context-free and context-sensitive kernels: update and deletion equivalence in abstract argumentation. In: *Proceedings of the 21st European Conference on Artificial Intelligence*, pp. 63–68, IOS Press.

Ben-Eliyahu, R. and Dechter, R. 1994. Propositional semantics for disjunctive logic programs. *Annals of Mathematics and Artificial Intelligence* 12: 53–87.

Besnard, P. and Hunter, A. 2014. Constructing argument graphs with deductive arguments: a tutorial. *Argument & Computation* 5(1):5–30.

Bodanza, G.; Tohmé, F.; and Auday, M. 2017. Collective argumentation: a survey of aggregation issues around argumentation frameworks. *Argument & Computation* 8(1):1–34.

Brewka, G. and Eiter, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In: *AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence*, pp. 385–390.

Brewka, G.; Eiter, T.; and Truszczynski, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54:93–103.

Caminada, M.; Sá, S.; Alcântara, J.; and Dvořák, W. 2015. On the equivalence between logic programming semantics and argumentation semantics. *Journal of Approximate Reasoning* 58: 87–111.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence* 77:321–357.

Dung P. M.; Kowalski, R. A.; and Toni, F. 2009. Assumption-based argumentation. In: *Argumentation in Artificial Intelligence* (I. Rahwan and G. R.. Simari, Eds.), Springer, 199–218.

Eiter, T.; Leone, N.; and Saccá, D. 1998. Expressive power and complexity of partial models for disjunctive deductive databases. *Theoretical Computer Science*, 206:181–218, Elsevier.

Eiter, T.; Fink, M.; Woltran, S. 2007. Semantical characterizations and complexity of equivalences in answer set programming. *ACM TOCL* 8(3), 17.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In: *Proceedings of the 5th International Conference and Symposium on Logic Programming*, MIT Press, pp. 1070–1080.

Hunter, A. 2013. A probabilistic approach to modelling uncertain logical arguments. *J. Approximate Reasoning* 54:47–81.

Inoue, K. 1994. Hypothetical reasoning in logic programs. *Journal of Logic Programming* 18(3):191–227.

Janhunen, T.; Niemela, I.; Seipel, D.; Simons, P.; and You, J.-H.: Unfolding partiality and disjunctions in stable model semantics. *ACM TOCL* 7(1), 1–37.

Kakas, A. C.; Kowalski, R. A.; and Toni, F. 1992. Abductive logic programming. *Journal of Logic and Computation* 2(6):719–770.

Coste-Marquis, S.; Devred, C.; and Marquis, P. 2006. Constrained argumentation frameworks. In: *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 112–122.

Lifschitz, V.; Tang, L. R.; and Turner, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and AI* 25(3):369–389.

Przymusinski, T. C. 1991. Stable semantics for disjunctive programs. *New Generation Computing* 9:401–424.

Van Gelder, A.; Ross, K.; and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM* 38:620–650.