Understanding Inconsistency –

A Contribution to the Field of Non-monotonic Reasoning

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet

INFORMATIK

vorgelegt

von Dipl.-Wirtsch.-Math. Markus Ulbricht

geboren am 21. September 1990 in Leipzig.

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Gerhard Brewka, Universität Leipzig, Deutschland
2. Prof. Anthony Hunter, University College London, UK

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung
am 04. July 2019 mit dem Gesamtprädikat summa cum laude.

Acknowledgements

First of all, I want to thank my supervisor Gerd Brewka for his continuous support, the words of advice he was never tired to give me, the freedom in my research and the pleasant working atmosphere he provided. Working with him could not have been more enjoyable. I greatly appreciate the opportunity to join his team as a PHD student. Thank you so much! I also have to say he takes good puns to the next level – unimaginable!

Thanks to Matthias Thimm for not only inspiring many parts of this work and being a great coauthor, but also helping to guide my first steps as a newcomer.

I want to thank Manfred Droste for making me aware of QuantLA and making me a part of his infrastructure and work environment during my time as scholarship holder. I also thank all the people I met during this time for the enjoyable atmosphere they created, especially the 1st generation QuantLA scholarship holders for welcoming me and answering countless annoying questions. I also thank Gerd Brewka's group for welcoming me in their team during the final months of my PHD studies.

This work and my employment are funded by the German Research Foundation (DFG), Research Training Group "Quantitative Logics and Automata" and Research Unit "Hybrid Reasoning for Intelligent Systems".

Without my family, especially my parents, none of this would have been possible. Thank you for your continuous support and love. Finally, I thank Teo for always being there for me –in real life as well as technical discussions–, for her support and genuine love. Thank you also for spending so much time with me turning this thesis into a better version.

Abstract

Conflicting information in an agent's knowledge base may lead to a semantical defect, that is, a situation where it is impossible to draw any plausible conclusion. Finding out the reasons for the observed inconsistency and restoring consistency in a certain minimal way are frequently occurring issues in the research area of knowledge representation and reasoning. In a seminal paper Raymond Reiter proves a duality between maximal consistent subsets of a propositional knowledge base and minimal hitting sets of each minimal conflict – the famous hitting set duality. We extend Reiter's result to arbitrary non-monotonic logics. To this end, we develop a refined notion of inconsistency, called *strong inconsistency*. We show that minimal strongly inconsistent subsets play a similar role as minimal inconsistent subsets in propositional logic. In particular, the duality between hitting sets of minimal inconsistent subsets and maximal consistent subsets generalizes to arbitrary logics if the stronger notion of inconsistency is used. We cover various notions of repairs and characterize them using analogous hitting set dualities. Our analysis also includes an investigation of structural properties of knowledge bases with respect to our notions.

Minimal inconsistent subsets of knowledge bases in monotonic logics play an important role when investigating the reasons for conflicts and trying to handle them, but also for inconsistency measurement. Our notion of strong inconsistency thus allows us to extend existing results to non-monotonic logics. While measuring inconsistency in propositional logic has been investigated for some time now, taking the non-monotony into account poses new challenges. In order to tackle them, we focus on the structure of minimal strongly inconsistent subsets of a knowledge base. We propose measures based on this notion and investigate their behavior in a non-monotonic setting by revisiting existing rationality postulates, and analyzing the compliance of the proposed measures with these postulates.

We provide a series of first results in the context of inconsistency in abstract argumentation theory regarding the two most important reasoning modes, namely credulous as well as skeptical acceptance. Our analysis includes the following problems regarding minimal repairs: existence, verification, computation of one and characterization of all solutions. The latter will be tackled with our previously obtained duality results.

Finally, we investigate the complexity of various related reasoning problems and compare our results to existing ones for monotonic logics.

Contents

Chapter 1

Introduction

The research area of knowledge representation and reasoning (KR) (cf. [33]) gave birth to various formalisms during the past decades. Among others, Reiter's default logic [95], answer set programming (ASP) [36; 57; 58], abstract argumentation frameworks (AFs) [49], and description logics [7] were introduced and studied intensively. Many of them have in common that they are *non-monotonic*. This means, in a nutshell, that learning new information might result in withdrawing conclusions one was able to make before. From a mathematical point of view, a feature like this appears unnatural: Given a set of axioms for an object under consideration and some properties we are able to infer, why would it make sense to assume that introducing a novel axiom *weakens* the properties of our object? The answer to this question lies within the application of our logic. Mathematical logics such as first and second order logic are without any doubt the appropriate tool to investigate the properties of algebraic structures, functions, and so on, but they lack features to model commonsense reasoning.

The latter requires to draw conclusions with incomplete information which is only realizable when making default assumptions: Typically, we assume that our world is as expected and whenever something unexpected happens, we seek for an explanation with the least possible deviation from our normal state. An excellent example is visiting the doctor: If we tell her we have a runny nose, a sore throat and cough, then our doctor will probably assume we have a cold. Although these symptoms might occur in other diseases as well, there is no reason to assume we have, e.g., the flu, except we also mention fever, headache, or shivering. Hence, our doctor is making a default assumption and if we mention additional symptoms after further inquiry, she might withdraw this conclusion. These considerations led to the development of non-monotonic formalisms, see [34] for an excellent overview.

What most of the knowledge representation formalisms have in common is that a knowledge base may be *inconsistent*, that is, contains conflicting information. In propositional logics, for example, this happens when the given axioms allow to infer two complementary literals "a" and "$\neg a$". Inconsistency in non-monotonic logics is typically more subtle: Due to the use of default negation it may happen that some atom assumed to be false is again derived. Classical logics usually suffer from the *principle of explosion* which renders reasoning meaningless, as everything can be derived from inconsistent theories. Again, this is not an appropriate way to deal with inconsistency when trying to model commonsense reasoning. Handling inconsistency is an important task we all need to achieve permanently: The information contained in a news paper article is already likely to incorporate novel conflicts into the knowledge we (believe to) have about the world.

Inconsistency is thus an omnipresent phenomenon in logical accounts of KR [33; 41; 48; 59; 63]. Therefore, reasoning under inconsistency [25; 76; 77] is an important research area in KR. In general, one can distinguish two paradigms in handling inconsistent information. The first paradigm advocates living with inconsistency but providing non-classical semantics that allow the derivation of non-trivial information, such as using paraconsistent reasoning [29], reasoning with possibilistic logic [48], or formal argumentation [9]. The second paradigm is about explicitly restoring consistency, thus changing the theory itself, as it is done in, e. g., belief revision [63] or belief merging [75]. In a seminal paper belonging to the latter category, Reiter [96] proves that consistency of a knowledge base can be restored by resolving each minimal conflict of it. This result will be of utmost importance in many parts of the present work. A quantitative approach for analyzing inconsistencies is given by the field *inconsistency measurement* which investigates functions $\mathcal{I}$ that assign real numbers to knowledge bases, with the intuitive meaning that larger values indicate more severe inconsistency, see [105; 106; 108] for surveys and [5; 28; 62; 73; 86; 104] for some recent approaches. Inconsistency measures are applied to assess inconsistencies in, e.g., news paper reports [65], multi-agent systems [69], or databases [84].

1.1 About this Work

This thesis is devoted to analyzing inconsistency, in particular in non-monotonic logics, since a thorough understanding of inconsistency includes the handling of non-monotony. Several results obtained in this thesis rely on identifying central features of monotonic logics, which are not guaranteed in the general case. This oftentimes yields undesired properties of non-monotonic logics. The process of overcoming these issues in a reasonable and concise way does not only lift results to non-monotonic logics, but also contributes to gaining comprehension of the behavior of monotonic ones. In addition, inconsistency in the non-monotonic case includes an important novel aspect: It is possible to resolve conflicts by adding information to a knowledge base. Although this feature does not have a straightforward counterpart in monotonic reasoning, we will see that we can obtain similar results, phrased for symmetric notions.

The research area of measuring inconsistency yields important contributions to understanding and analyzing inconsistency. The goal is to assess the severity of conflicts in a knowledge bases. Inconsistencies are studied from a syntactical as well as semantical point of view. Some approaches, for example, investigate conflicting subsets of knowledge bases and assess their contribution to inconsistency. Other approaches focus specifically on the role of single formulas and assess their "blame" for inconsistency of a knowledge base. Extending these insights and results to non-monotonic reasoning is a promising research direction when trying to gain a comprehensive understanding of inconsistency. The tools we develop in this work are indeed capable of providing some steps in this direction. Although our general approach allows us to phrase our results for arbitrary non-monotonic logics, many aspects of measuring inconsistency rely on specific properties of particular frameworks. This will be taken into account by a discussion on inconsistency in answer set programming.

An important aspect of real world applications is of course the computational complexity of the problems related to the notions we develop. If, for example, deciding inconsistency of a propositional knowledge base was an undecidable problem, then handling inconsistency based on minimal inconsistent subsets would remain a purely academical challenge. We thus perform an investigation of the computational complexity of some naturally arising decision and function problems. Thereby, we place value on a comparison between monotonic and non-monotonic logics.

We are interested in the structural properties of knowledge bases, the connection between consistent and inconsistent subsets, approaches to turn a given inconsistent knowledge base into a consistent one and tools to assess the severity of inconsistencies. Most of the obtained results are general and hence apply to arbitrary, monotonic and non-monotonic logics. We will thus give a general definition of an abstract logic, and demonstrate how to obtain established monotonic and non-monotonic frameworks from the literature as special cases. During our investigation, the actual challenge is covering non-monotonic logics. In fact, some of the results we obtain are folklore when it comes to monotonic reasoning (although rarely phrased in the setting of an abstract logic as we are going to do here). When extending results to non-monotonic logics we always place value on natural adjustments to the existing notions, a clear and accountable motivation, and ensure that our notions coincide with the established ones when considering the special case of a monotonic logic.

Although the goal of this work is to cover arbitrary logics, some parts investigate concrete frameworks in order to discover deeper results. For example, we analyze the connection between different reasoning modes and semantics for abstract argumentation frameworks (Chapter 5). Here, we are concerned about situations where a given framework is inconsistent in the sense that no argument is accepted. In Section 4.5 we demonstrate how to apply our observations about inconsistency measurement in non-monotonic logics to answer set programming. We will see that tailoring our notions to a specific framework provides us with powerful tools, despite being not as general.

The thesis is structured as follows:

In *Chapter 2* we provide the preliminaries for this work. In particular, we stipulate some basic notions and give the necessary background for propositional logic, answer set programming and abstract argumentation frameworks. For the most part, propositional logic will be our generic example of a monotonic logic, whereas ASP and AFs shall demonstrate features of non-monotonic logics. We define formally what we mean by a logic. We build on the abstract characterization of logics in [35], but extend this framework suitably to capture an equally abstract notion of consistency. Moreover, we introduce a new notion of monotony which covers logics with multiple belief sets.

In *Chapter 3* we investigate structural properties of knowledge bases (in monotonic as well as non-monotonic logics), especially the connection between consistent and inconsistent subsets. In particular we generalize Reiter's famous hitting set theorem to arbitrary logics. To achieve this generalization, we define strongly inconsistent subsets of a knowledge base, a central notion of this work (Section 3.1). Motivated by the observation that a knowledge base of a non-monotonic logic can also be repaired by *adding* formulas –whereas Reiter's duality is only concerned about *removing*–, we investigate situations where we are given potential additional assumptions to repair a knowledge base. For this, we characterize again the minimal modifications to a knowledge base in terms of a hitting set duality (Section 3.2). Our general definition of a logic and consistency resp. inconsistency allows us to satisfy more sophisticated requirements like not allowing for removal of certain parts of the knowledge base. This can be achieved even without giving novel proofs, but applying our main theorems to appropriate auxiliary logics. We demonstrate this in Section 3.3. We conclude this chapter with a discussion on possibly infinite knowledge bases (Section 3.4).

We devote *Chapter 4* to measuring inconsistency in non-monotonic logics. The results from Chapter 3 suggest that our notion of strong inconsistency is the appropriate refinement of inconsistency as it preserves important structural properties. Since many inconsistency measures from the literature are based on (the number of) minimal inconsistent subsets of a knowledge base, we may give natural generalizations of those measures (Section 4.1). In order to assess the quality of inconsistency measures, many rationality postulates for their behavior have been proposed in the literature. A few of them still make sense when considering non-monotonic logics, but in most cases adjustments are required in order to take the special features of non-monotonic logics into account (Section 4.2). We analyze the compliance of our generalized measures with the refined rationality postulates (Section 4.3). The observation that conflicts may be resolved due to additional formulas gives rise to the question how to assess inconsistencies of a knowledge base within the context of a larger one. It might be the case that some conflicts turn out to be less severe when given the "bigger picture". Situations like this are investigated in Section 4.4, with tools developed in Section 3.2. We conclude this chapter with a discussion on measuring inconsistency in ASP (Section 4.5).

In *Chapter 5* we investigate AFs which do not allow to draw a meaningful conclusion in the sense that no argument is accepted. We do so with respect to various semantics and the two standard reasoning modes, credulous and skeptical. This chapter shall demonstrate how to apply our techniques, in particular from Chapter 3, to a specific logic. We start by introducing the semantics we are going to consider and formally define inconsistency in this setting (Section 5.1). We investigate the existence of repairs wrt. different reasoning modes and semantics, as well as the relations between them (Section 5.2). We demonstrate how our previous results yield duality characterizations for repairs of AFs (Section 5.3). In order to refine our analysis, we consider specific situations where we have some additional knowledge about the AF under consideration, e.g., symmetry of the attack relation [43] or splitting [12] (Section 5.4). We conclude this chapter with a short case study, illustrating how to repair some given AFs (Section 5.5).

In *Chapter 6* we provide an analysis of the computational complexity of related decision and function problems to our notions, e.g., how difficult is it to verify that given subset of a knowledge base is strongly inconsistent? This chapter requires some technical background (Section 6.1). We start with decision problems about strong inconsistency. Since we cover arbitrary logics, our results will be given in comparison to the complexity of the corresponding satisfiability check. In order to assess these results appropriately, we compare them to the same problems for monotonic logics. More precisely we consider quantified boolean formulas as generic framework to capture the polynomial hierarchy. Most of the results are general upper bounds. We demonstrate how to obtain lower bounds in ASP (Section 6.2). In particular motivated by our discussion on inconsistency measurement we also investigate the corresponding counting problem, i. e., how many strongly inconsistent subsets does a knowledge base possess (Section 6.3)? In Section 6.4 we give complexity results for AFs, focusing on consistent rather than inconsistent subsets. This is motivated by the analysis provided in Chapter 5.

In *Chapter 7* we conclude and discuss future work.

1.2 Publications

Most of the result we present in this work have been published before. We name those publications and indicate to which sections they contribute.

Conference papers:

- [Ulbricht et al., 2016]: *Measuring Inconsistency in Answer Set Programs*, in: Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA' 16) contains parts of the results from Section 4.5.
- [Brewka et al., 2017]: *Strong Inconsistency in Non-monotonic Reasoning*, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI'17) contributes to Chapter 3, in particular Section 3.1, lays foundations for Chapter 4 and contains the results from Section 6.2.
- [Ulbricht et al., 2018]: *Measuring Strong Inconsistency*, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18) contains (most of) the results from Chapter 4 and Section 6.3.
- [Baumann and Ulbricht, 2018]: *If Nothing Is Accepted - Repairing Argumentation Frameworks*, in: In Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR'18) contains the results from Chapter 5 and Section 6.4; the paper was nominated for the Ray Reiter Best Paper Award.
- [Ulbricht, 2019]: *Repairing Non-monotonic Knowledge Bases*, accepted for publication in: Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA'19) contains the results from Section 3.2.

Journal paper:

- [Brewka et al., 2019]: *Strong Inconsistency*, in: Artificial Intelligence, is the extended version of [Brewka et al., 2017].

Book contribution:

- [Ulbricht et al., 2018]: *Inconsistency Measures for Disjunctive Logic Programs Under Answer Set Semantics*, in: Measuring Inconsistency in Information, is the extended version of [Ulbricht et al., 2016].

Chapter 2

Background

2.1 Basic Notations

In what follows, we establish some basic notions and notations we require.

Most of our results are concerned with sets and their relations. Given two sets X and Y we denote the ordinary reflexive subset relation by $\subseteq$, i. e., $X \subseteq Y$ iff $x \in X \Rightarrow x \in Y$. If X is a *proper* subset of Y, we write $X \subsetneq Y$, i. e., $X \subsetneq Y$ iff $X \subseteq Y$ and $X \neq Y$. We avoid the symbol $\subset$. The terms *minimality* and *maximality* are always to be understood wrt. to the $\subseteq$-relation of sets. Thus, if X is *minimal* with a certain property E, then if Y possess E as well, $Y \subseteq X$ implies $Y = X$. Similarly, if X is the *least* set with a certain property E, then if Y possess E as well, $X \subseteq Y$ holds.

In particular in Section 3.2.2, consideration of tuples of sets will be crucial. So given four sets X_1, Y_1, X_2, Y_2, we extend the basic set operations via $(X_1, Y_1) \subseteq (X_2, Y_2)$ iff $\subseteq$ holds component-wise, i. e., $X_1 \subseteq X_2$ and $Y_1 \subseteq Y_2$. In particular, "$=$" is extended component-wise as well. This definition also induces minimality and maximality for tuples: if (X_1, Y_1) is minimal with a certain property E, then if (X_2, Y_2) possesses E as well, $(X_2, Y_2) \subseteq (X_1, Y_2)$ implies $X_1 = X_2$ and $Y_1 = Y_2$. We also define the intersection component-wise, i. e., $(X_1, Y_1) \cap (X_2, Y_2) = (X_1 \cap X_2, Y_1 \cap Y_2)$. This yields in particular $(X_1, Y_1) \cap (X_2, Y_2) = \emptyset$ iff $X_1 \cap Y_1 = \emptyset$ and $X_2 \cap Y_2 = \emptyset$.

Most of our main results in Chapter 3 will make use of hitting sets:

Definition 2.1.1. Let $\mathcal{X}$ be a set of sets. We call $\mathcal{S}$ a *hitting set* of $\mathcal{X}$ if $\mathcal{S} \cap X \neq \emptyset$ for each $X \in \mathcal{X}$. Let $minHS(\mathcal{X})$ denote the set of all minimal hitting sets of $\mathcal{X}$.

Observe that due to our understanding of minimality, $\mathcal{S}$ is a *minimal* hitting set of $\mathcal{X}$ if $\mathcal{S}$ is a hitting set of $\mathcal{X}$ and no $\mathcal{S}' \subsetneq \mathcal{S}$ is a hitting set of $\mathcal{X}$.

Example 2.1.2 (Hitting sets)**.** Assume we have $\mathcal{X} = \{X_1, X_2, X_3\}$ with $X_1 = \{a, b, c\}$, $X_2 = \{b, c\}$ and $X_3 = \{c, d\}$. Then, $\mathcal{S} = \{a, b, d\}$ is a hitting set of $\mathcal{X}$ since

$$\mathcal{S} \cap X_1 = \{a, b\} \qquad \mathcal{S} \cap X_2 = \{b\} \qquad \mathcal{S} \cap X_3 = \{d\}.$$

The hitting set $\mathcal{S}$ is not *minimal* though since $\mathcal{S}' = \{b, d\}$ is a hitting set of $\mathcal{X}$ as well. The latter is minimal.

We also want to emphasize that now, the definition of a hitting set applies to a set $\mathcal{X}$ of tuples of sets in the natural way. Recall that the subset relation was extended to tuples component-wise, which induces emptiness of a tuple iff both components are.

Example 2.1.3 (Hitting sets of tuples)**.** Let $\mathcal{X}$ be the following set of tuples:

$$\mathcal{X} = \{(\{a,b\},\{b,c\}),\ (\{c\},\{d\})\}$$

Then, $\mathcal{S} = (\{a\},\{d\})$ is a hitting set of $\mathcal{X}$ since

$$(\{a\},\{d\}) \cap (\{a,b\},\{b,c\}) = (\{a\},\emptyset) \qquad (\{a\},\{d\}) \cap (\{c\},\{d\}) = (\emptyset,\{d\})$$

Note that $\mathcal{S}$ is even a *minimal* hitting set of $\mathcal{X}$, because moving to, e. g., $\mathcal{S}' = (\emptyset,\{d\})$ yields

$$(\emptyset,\{d\}) \cap (\{a,b\},\{b,c\}) = (\emptyset,\emptyset).$$

Similarly, $\mathcal{S}'' = (\{a\},\emptyset)$ is no hitting set of $\mathcal{X}$, either.

The last notion we require regarding sets is the definition of an *upward-closed* set of sets:

Definition 2.1.4. Let $\mathcal{X}$ be a set of sets and let $\mathcal{Y} \subseteq \mathcal{X}$. We call $\mathcal{Y}$ *upward-closed* wrt. $\mathcal{X}$ if $X \subseteq X'$ and $X \in \mathcal{Y}$ already implies $X' \in \mathcal{Y}$ for each $X, X' \in \mathcal{X}$.

Consider again two sets X and Y. If there is a subset $X' \subseteq X$ such that $f : X' \to Y$ is a mapping, then we call f a *partial* mapping from X to Y, denoted by $f : X \dashrightarrow Y$. If A is a set of propositional atoms, then a mapping $f : A \to \{0,1\}$ is called an *assignment* and a partial mapping $f : A \dashrightarrow \{0,1\}$ is called a *partial assignment*.

We denote the power set of X by 2^X, i. e., $2^X = \{Y \mid Y \subseteq X\}$. The cardinality of a set X is $|X|$. We will, however, only consider the cardinality of *finite* sets, yielding the simple definition $|X| = n \in \mathbb{N}$ iff there is a bijection $\rho : X \to \{1,\ldots,n\}$.

As usual, $\mathbb{R}$ denotes the real numbers. Let $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$, $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ and $\overline{\mathbb{R}}_{\geq 0} = \mathbb{R}_{\geq 0} \cup \{\infty\}$.

2.2 Propositional Logic

We define propositional logic as usual, so let us briefly recall the standard definitions. Let A be a set of propositional atoms, i. e., a propositional signature. Any atom $a \in A$ is a *well-formed formula* wrt. A. If ϕ and ψ are well-formed formulas wrt. A, then $\neg\phi$, $\phi \wedge \psi$, and $\phi \vee \psi$ are also well-formed formulas wrt. A (we also assume that the usual abbreviations $\to, \leftrightarrow$ are defined accordingly). A formula ϕ is in *conjunctive normal form* (CNF) if it is of the form $\phi = C_1 \wedge \ldots \wedge C_r$ where each C_k is a *clause*, i. e., C_k is of the form $C_k = a_{k,1} \vee \ldots \vee a_{k,n(k)}$ for atoms $a_{k,j}$. If each C_k contains at most 3 literals, then ϕ is in 3-CNF. We abuse notation and identify a formula ϕ of this form with the set $\{C_1,\ldots,C_r\}$ of clauses. Similarly, a formula ϕ is in *disjunctive normal form* (DNF) if $\phi = C_1 \vee \ldots \vee C_r$ where $C_k = a_{k,1} \wedge \ldots \wedge a_{k,n(k)}$. If each C_k contains at most 3 literals, then ϕ is in 3-DNF.

If $\omega : A \dashrightarrow \{0,1\}$ is a (partial) assignment, then ω is extended to formulas in the usual way:

- $\omega(\neg a) = 1 - a$,
- $\omega(\phi \wedge \psi) = \min\{\omega(\phi), \omega(\psi)\}$, and
- $\omega(\phi \vee \psi) = \max\{\omega(\phi), \omega(\psi)\}$.

If $\omega(\phi) = 1$, then we say ω *satisfies* ϕ. A *propositional knowledge base* $\mathcal{K}$ is a set of propositional formulas. As usual, ω satisfies $\mathcal{K}$ iff ω satisfies ϕ for all $\phi \in \mathcal{K}$. We say $\mathcal{K}$ *entails* a formula ϕ, denoted by $\mathcal{K} \models \phi$, iff each assignment ω satisfying $\mathcal{K}$ also satisfies ϕ.

We call a knowledge base $\mathcal{K}$ *consistent* iff there is an assignment ω satisfying $\mathcal{K}$, otherwise it is called *inconsistent*.

Example 2.2.1. Consider the propositional knowledge base $\mathcal{K} = \{a,\ a \rightarrow b,\ \neg b,\ c,\ \neg c\}$. Obviously, no assignment satisfies $\mathcal{K}$. Hence, $\mathcal{K}$ is inconsistent.

2.3 Answer Set Programming

Answer set programming (ASP, see [36] for an overview) is an emerging problem solving paradigm. It is based on logic programs under the answer set semantics [57; 58], a popular non-monotonic formalism for knowledge representation and reasoning which consists of rules possibly containing default-negated literals. Inconsistencies occur in ASP for two reasons, cf. [99]. First, the rules allow the derivation of two complementary literals l and $\neg l$ –also called *incoherence*, see [82]– thus producing inconsistencies similar to propositional logic. Second, due to the use of default negation it may happen that some literal assumed to be false is again derived (called *instability*).

Let us consider logic programs with disjunction in the head of rules and two kinds of negation, namely strong negation "$\neg$" and default negation "not", under the answer set semantics [57; 58]. In [58] such programs were called extended disjunctive databases, whereas Gelfond and Leone [57] simply speak of logic programs or A-Prolog programs. We will call these programs *extended disjunctive logic programs* or disjunctive logic programs for short.

Assume we are given an infinite set A of atoms. Let $Lit(A)$ be the set of all literals over A. Then, a disjunctive logic program P (over A) is a finite set of rules r of the form

$$l_0 \vee ... \vee l_k \leftarrow l_{k+1}, \dots, l_m, \text{not } l_{m+1}, \dots, \text{not } l_n. \tag{2.1}$$

where $l_0, \dots, l_n$ are literals over A and $0 \leq k \leq m \leq n$. If $k = 0$ holds for each rule $r \in P$, then we call P a *normal logic program*. When there is no risk of confusion, we will simply speak of *logic programs* instead of disjunctive logic programs resp. normal logic programs.

For a rule r of the form (2.1) let $head(r) = \{l_0, \dots, l_k\}$, $pos(r) = \{l_{k+1}, \dots, l_m\}$, and $neg(r) = \{l_{m+1}, \dots, l_n\}$. If $m = n = k$, then r is written "$head(r)$." instead of "$head(r) \leftarrow$." and if in addition $k = 0$ holds, then the rule is called a *fact*.

Now we are ready to define answer sets of a given program.

Definition 2.3.1. Let P be a logic program over A such that $neg(r) = \emptyset$ holds for each rule $r \in P$. Then, a set M of literals is a *model* of P if for all $r \in P$ the following is true: If $pos(r) \subseteq M$, then $head(r) \cap M \neq \emptyset$. If M is a model of P containing two complementary literals, then $M = Lit(A)$. A minimal model of P is called an *answer set* of P.

Let us consider some example programs where the condition $neg(r) = \emptyset$ is met:

Example 2.3.2. Let P be the program $P = \{a.\}$. Trivially, $\{a\}$ is model a of P, and in particular an answer set. If we consider P as a program over $A = \{a, b\}$, then $\{a, b\}$ is a model of P as well, but not minimal and hence no answer set.

Example 2.3.3. Now consider the program P:

$$P: \qquad a \vee b.$$

The program has two answer sets $\{a\}$ and $\{b\}$, as well as the model $\{a, b\}$. The latter is no answer set.

Example 2.3.4. The program P

$$P: \qquad a \vee b. \qquad a \leftarrow b. \qquad c. \qquad \neg c.$$

possesses the answer set $Lit(A)$.

The goal is of course to extend this definition of answer sets to arbitrary logic programs. For that assume we are given a logic program P and a set M of literals. We call

$$P^M = \{head(r) \leftarrow pos(r) \mid head(r) \leftarrow pos(r), neg(r) \in P, neg(r) \cap M = \emptyset\}$$

the *reduct* of P wrt. M. Observe that P^M itself is a logic program and $neg(r) = \emptyset$ holds for each $r \in P^M$. Now we define:

Definition 2.3.5. Let P be a logic program over A. A set M of literals is an *answer set* of P iff M is an answer set of P^M.

Example 2.3.6. Let P be the program

$$P: \qquad a \leftarrow \text{not } a.$$

Let us consider $M_1 = \emptyset$ and $M_2 = \{a\}$. We find

$$P^{M_1} = \{a.\} \qquad P^{M_2} = \{\}.$$

In particular, both M_1 and M_2 are *not* answer sets of P, because M_1 is not a model of P^{M_1} and M_2 –although being a model of P^{M_2}– is not minimal. Indeed, P is the simplest example of a logic program which possesses no answer set.

Example 2.3.7. Now consider the following program P, which will be one of our running examples:

$$\begin{array}{llll} P: & a \vee b. & & a \leftarrow b. \\ & c \leftarrow \text{not } b. & & \neg c \leftarrow \text{not } b. \end{array}$$

The program has no answer set. To see this, consider the three candidates $\{a\}$, $\{b\}$ and $\{a, b\}$ with

$$\begin{array}{rllll} P^{\{a\}}: & a \vee b. & a \leftarrow b. & c. & \neg c. \\ P^{\{b\}}: & a \vee b. & a \leftarrow b. & & \\ P^{\{a,b\}}: & a \vee b. & a \leftarrow b. & & \end{array}$$

We see that $\{a\}$ is not a model of $P^{\{a\}}$, $\{b\}$ is not a model of $P^{\{b\}}$ and $\{a, b\}$ is a model of $P^{\{a,b\}}$, but not minimal.

Note that so far, we defined what an answer set is, no matter whether it is consistent or not. Recall that if a model contains two complementary literals, it is extended to $Lit(A)$. Clearly, this should not be considered a *consistent* answer set. Moreover, in order for a program to be consistent, it should possess consistent answer sets. Hence, we define:

Definition 2.3.8. Let P be a logic program over A. An answer set M is *consistent* if it does not contain two complementary literals. The program P is consistent if it possesses at least one consistent answer set.

We thus see that the program from Example 2.3.7 is inconsistent.

Some of our example will make use of variables, usually denoted by X or Y. As usual, a rule of the form "$head(X) \leftarrow body(Y)$." is a shorthand for all ground instances of this rule, i. e., each variable may be replaced with an arbitrary literal over A. Moreover, "$head(X) \leftarrow body(Y), X \neq Y$." is defined similar, but we need to assign different literals to X and Y. As usual, we assume the set A consists of all atoms occurring in a given program P, if not stated otherwise.

Finally, we call a rule of the form

$$r: \quad a \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n, \text{not } a. \tag{2.2}$$

where a is an atom that does not occur elsewhere in a given program P a *constraint*. The intuitive meaning is that no answer set of P is allowed to contain all literals $l_1, \ldots, l_m$ and none of the literals $l_{m+1}, \ldots, l_n$. We use the established shorthand

$$\leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n.$$

for constraints of the form (2.2).

2.4 Abstract Argumentation Frameworks

In the original formulation, an abstract argumentation framework (AF) F is a directed graph $F = (A, R)$ where nodes in A represent arguments and the relation R models "attacks", i. e., for $a, b \in A$, if $(a, b) \in R$ then a is a counterargument for b and we say a attacks b. Abstract argumentation frameworks consider the problem of argumentation only at this abstract level and do neither consider the inner structure of arguments nor how the attack relation is derived. Semantics are given to an abstract argumentation framework $F = (A, R)$ by identifying sets $E \subseteq A$ of arguments (called extensions) that can be "jointly accepted". The literature offers various approaches on how to define "jointly accepted", see [49].

Chapter 5 of this work is devoted to inconsistency in abstract argumentation, taking a wide range of different semantics into account. However, for now, we will focus on so-called *stable semantics*. They are intuitive, easy to understand and thus an appropriate tool to illustrate our results with examples from abstract argumentation.

Definition 2.4.1. Let $F = (A, R)$ be an AF. A set $E \subseteq A$ is called *stable extension* if

- $a, b \in E$ implies $(a, b) \notin R$,
- $c \in A \setminus E$ implies there is an $a \in E$ with $(a, c) \in R$.

We denote the set of stable extensions of an AF F by $stb(F)$.

The first item ensures that E is *conflict free*, i. e., there are no two “accepted” arguments that attack each other. This requirement is quite usual for abstract argumentation semantics. The second item is what characterizes stable semantics: each argument which is not included in E shall be attacked by E. This is a rather decisive requirement, partitioning the arguments in “accepted” and “attacked” ones.

Example 2.4.2. Consider the AF $F = (A, R)$ where

$$A = \{a, b, c, d\} \qquad R = \{(a, b),\ (b, a),\ (c, b),\ (c, c),\ (d, c)\}.$$

The AF is depicted in Figure 2.1:

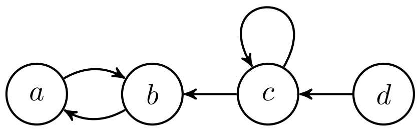


Figure 2.1: The argumentation framework F from Example 2.4.2

We have

$$stb(F) = \{\{a, d\},\ \{b, d\}\}$$

which is depicted in Figure 2.2:

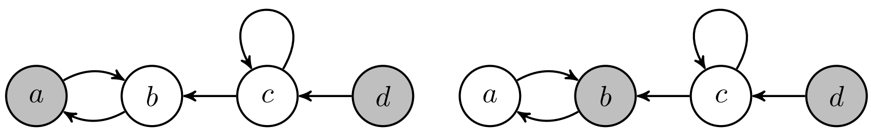


Figure 2.2: Stable extensions of F from Example 2.4.2

It can be verified straightforwardly that there is no other stable extension.

In comparison to other semantics, stable semantics possess a rare property, namely that an AF might have no extension at all. This is, for example the case for the following simplified version of the previous AF.

Example 2.4.3. Consider the AF $F = (A, R)$ (see Figure 2.3) with

$$A = \{a, b, c\} \qquad R = \{(a, b),\ (b, c),\ (c, c)\}\ :$$

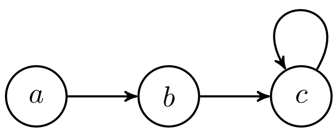


Figure 2.3: The argumentation framework F from Example 2.4.3

The argument c attacks itself, so $c \notin E$ if E is conflict free. However, we see that in order to attack c, the argument b must be included in our extension E, but then, a can neither be included in E nor attacked.

This motivates our definition of inconsistency of an AF: Similar to ASP where we call a program inconsistent whenever there is no (consistent) answer set, we will call an AF inconsistent whenever there is no stable extension.

Definition 2.4.4. Let F be an AF. If $stb(F) = \emptyset$, then we call F *inconsistent* wrt. stable semantics. If there is no risk of confusion, we will call F simply *inconsistent*.

This notion will be generalized and further investigated in Chapter 5. For now, we stick with this notion of inconsistency in abstract argumentation frameworks.

2.5 Logics - A General Approach

Most of the main results in this work are independent of the actual logic, i. e., they hold for propositional logic, ASP, AFs and many other frameworks. It is thus natural to phrase those results for an arbitrary but fixed logic L. To achieve this, we require a general definition of a logic, covering a wide range of frameworks as special cases.

In a nutshell, a logic L consists of syntax and semantics of formulas. To model the syntax properly, we stipulate a set $\mathcal{WF}$ of so-called well-formed formulas. Any knowledge base $\mathcal{K}$ is a subset of $\mathcal{WF}$. To model the semantics, we let $\mathcal{BS}$ be a set of so-called belief sets. Intuitively, given a knowledge base $\mathcal{K}$, the set of all that can be inferred from $\mathcal{K}$ is $B \subseteq \mathcal{BS}$. To formalize this, a mapping $\mathcal{ACC}$ assigns the set B of corresponding belief sets to each knowledge base $\mathcal{K}$. For example, if our knowledge base is a logic program P, then we want to assign all answer sets of P to it. Hence, $\mathcal{BS}$ should contain all potential answer sets of P and we expect $\mathcal{ACC}(P) = \{M \mid M \text{ is an answer set of } P\}$. Finally, some belief sets are considered inconsistent. We call the set of all inconsistent belief sets $\mathcal{INC}$. The inconsistent belief sets are supposed to model conflicting conclusions. We thus expect them to be upward-closed in $\mathcal{BS}$, i. e., if $B, C \in \mathcal{BS}$ with $B \subseteq C$ and B is in $\mathcal{INC}$, then $C \in \mathcal{INC}$ as well.

Hence, our definition of a logic is as follows.

Definition 2.5.1. A logic L is a tuple

$$L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$$

where $\mathcal{WF}$ is a set (of well-formed formulas), $\mathcal{BS}$ is a set (of belief sets), $\mathcal{INC} \subseteq \mathcal{BS}$ is upward closed wrt. $\mathcal{BS}$ and $\mathcal{ACC} : 2^{\mathcal{WF}} \to 2^{\mathcal{BS}}$ assigns a collection of belief sets to each subset of $\mathcal{WF}$. A *knowledge base* $\mathcal{K}$ of L is a finite subset of $\mathcal{WF}$.

Note that this definition only covers finite knowledge bases. This does not exclude infinite knowledge bases which can be represented in a finite way. For example in predicate calculus we might have an expression "$\forall x.(\forall y.(R(x, y) \to R(y, x)))$" which represents an infinite number of formulas if the underlying universe is infinite.

2.5.1 Logic Instances: Previous Frameworks

In order to familiarize us with this abstract definition of a logic, let us illustrate how to model propositional logic, ASP and AFs under stable semantics as a logic according to Definition 2.5.1.

Example 2.5.2 (Propositional logic)**.** Let A be a set of propositional atoms. We define a logic

$$L_{\mathrm{P}}^{A} = \left(\mathcal{WF}_{\mathrm{P}}^{A}, \mathcal{BS}_{\mathrm{P}}^{A}, \mathcal{INC}_{\mathrm{P}}^{A}, \mathcal{ACC}_{\mathrm{P}}^{A}\right).$$

We let $\mathcal{WF}_{\mathrm{P}}^{A}$ be the well-formed formulas over A (see the inductive definition in Section 2.2) and $\mathcal{BS}_{\mathrm{P}}^{A}$ the deductively closed sets of formulas, i. e.,

$$\mathcal{BS}_{\mathrm{P}}^{A} = \left\{\mathcal{K} \subseteq \mathcal{WF}_{\mathrm{P}}^{A} \mid \mathcal{K} = \{\phi \mid \mathcal{K} \vDash \phi\}\right\}.$$

The set $\mathcal{INC}_{\mathrm{P}}^{A}$ is supposed to contain the inconsistent belief sets. Since $\mathcal{INC}_{\mathrm{P}}^{A} \subseteq \mathcal{BS}_{\mathrm{P}}^{A}$, any set in $\mathcal{INC}_{\mathrm{P}}^{A}$ needs to be deductively closed as well. As anything can be derived from an inconsistent knowledge base, the is no other choice than $\mathcal{INC}_{\mathrm{P}}^{A} = \left\{\mathcal{WF}_{\mathrm{P}}^{A}\right\}$. Finally, the mapping $\mathcal{ACC}_{\mathrm{P}}^{A}$ assigns to each $\mathcal{K} \subseteq \mathcal{WF}_{\mathrm{P}}^{A}$ the set containing its set of theorems, i. e.,

$$\mathcal{ACC}_{\mathrm{P}}^{A}(\mathcal{K}) = \{\{\phi \mid \mathcal{K} \vDash \phi\}\}.$$

During the remainder of this work, we omit the superscript A whenever there is no risk of confusion.

Example 2.5.3 (Disjunctive logic programs)**.** Let A be a set of propositional atoms. Extended disjunctive logic programs under answer set semantics over A can be modeled as logic

$$L_{\mathrm{ASP}}^{A} = \left(\mathcal{WF}_{\mathrm{ASP}}^{A}, \mathcal{BS}_{\mathrm{ASP}}^{A}, \mathcal{INC}_{\mathrm{ASP}}^{A}, \mathcal{ACC}_{\mathrm{ASP}}^{A}\right).$$

Here, $\mathcal{WF}_{\mathrm{ASP}}^{A}$ is the set of all rules of the form (2.1) over A (see Section 2.3). Moreover, $\mathcal{BS}_{\mathrm{ASP}}^{A}$ consists of the sets of literals over A, i. e.,

$$\mathcal{BS}_{\mathrm{ASP}}^{A} = 2^{A}$$

and $\mathcal{INC}_{\mathrm{ASP}}^{A} = \{Lit(A)\}$. The mapping $\mathcal{ACC}_{\mathrm{ASP}}^{A}$ assigns to a logic program $P \subseteq \mathcal{WF}_{\mathrm{ASP}}$ the set of all answer sets of P, i. e.,

$$\mathcal{ACC}_{\mathrm{ASP}}^{A}(P) = \left\{M \in 2^{A} \mid M \text{ is an answer set of } P\right\}.$$

As before ,we omit the superscript A whenever there is no risk of confusion.

It is a quite simple, yet pleasing observation that moving to a subclass of a certain logic just requires restricting $\mathcal{WF}$.

Example 2.5.4 (Normal logic programs)**.** If we let $\mathcal{WF}_{\mathrm{ASP}^*}^{A} \subseteq \mathcal{WF}_{\mathrm{ASP}}^{A}$ be the set of all rules of the form (2.1) with $k = 0$, then

$$L_{\mathrm{ASP}^*}^{A} = \left(\mathcal{WF}_{\mathrm{ASP}^*}^{A}, \mathcal{BS}_{\mathrm{ASP}}^{A}, \mathcal{INC}_{\mathrm{ASP}}^{A}, \mathcal{ACC}_{\mathrm{ASP}}^{A}\right)$$

is the logic corresponding to normal logic programs under answer set semantics.

Example 2.5.5 (Abstract argumentation frameworks)**.** Representing an AF $F = (A, R)$ as a logic according to Definition 2.5.1 requires some caution since a knowledge base is supposed to be a subset of $\mathcal{WF}$, but an AF is a tuple. In order to obtain a simple and intuitive representation of AFs as a set, let us assume a finite set A of arguments is given. Now, each well-formed formula corresponds to one attack within the AF.

More precisely, we define a logic

$$L_{\text{AF}}^{A} = \left(\mathcal{WF}_{\text{AF}}^{A}, \mathcal{BS}_{\text{AF}}^{A}, \mathcal{INC}_{\text{AF}}^{A}, \mathcal{ACC}_{\text{AF}}^{A}\right).$$

The set $\mathcal{WF}_{\text{AF}}$ is the set of all possible attacks, i. e., $\mathcal{WF}_{\text{AF}} = (A \times A)$. Belief sets are arbitrary sets of arguments, i. e., $\mathcal{BS}_{\text{AF}} = 2^{A}$. We consider no notion of an inconsistent set of arguments (recall that an AF F is inconsistent if $stb(F) = \emptyset$). We thus let $\mathcal{INC}_{\text{AF}} = \emptyset$. Hence, to represent an AF $F = (A, R)$, we fix the set A of attacks and let R be our knowledge base. Now, the AF under consideration is F, but the knowledge base is the set R. So we have

$$\mathcal{ACC}_{\text{AF}}^{A}(R) = stb(F), \qquad \text{where } F = (A, R).$$

We will thus sometimes abuse terminology and speak of the AF R instead of $F = (A, R)$ when A is given. As usual, we omit the superscript A whenever it is implicitly clear.

The reader may verify that a wide spectrum of other logics can be modeled as well, e. g., first-order logic, modal logic, probabilistic and fuzzy logics.

2.5.2 Inconsistency

Consider a logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$. Until now, the meaning of the set $\mathcal{INC}$ –the inconsistent belief sets– is only intuitively clear. Formally $\mathcal{INC}$ is required to clarify what we mean by an inconsistent knowledge base. The definition of inconsistency is quite natural: A knowledge base should possess at least one consistent belief set in order to be consistent, otherwise we call it inconsistent. For example, a logic program should have at least one consistent answer set, as already noticed in Definition 2.3.8. A knowledge base $\mathcal{K}$ is thus inconsistent if all belief sets $\mathcal{ACC}(\mathcal{K})$ are.

Definition 2.5.6. A knowledge base $\mathcal{K}$ is called *inconsistent* iff $\mathcal{ACC}(\mathcal{K}) \subseteq \mathcal{INC}$. Let $I(\mathcal{K})$ denote the collection of all inconsistent subsets of $\mathcal{K}$. Let $I_{min}(\mathcal{K})$ be the set of all minimal inconsistent subsets of $\mathcal{K}$.

To be precise, inconsistency is a property a knowledge base has with respect to a given logic L. We should thus write $I_{min}(\mathcal{K})^{L}$ instead of $I_{min}(\mathcal{K})$. However, in most cases the underlying logic will be clear or simply not important, so we may omit the superscript without risking confusion. The only exception will be Section 3.3 where we give the superscripts whenever this is necessary. This remark holds for all notions we are going to give during this work, whenever they are phrased for an arbitrary but fixed logic.

Now let us discuss the above definition of inconsistency. For propositional logic, inconsistency means that every formula can be inferred from a knowledge base.

Example 2.5.7 (Inconsistency in propositional logic)**.** Consider the propositional knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$ from above. Recall that $\mathcal{K}$ entails a contradiction, so by definition of propositional logic we have $\mathcal{ACC}_{\text{P}}(\mathcal{K}) = \{\{a, \neg a, b, \ldots\}\} = \{\mathcal{WF}_{\text{P}}\}$. In particular, $\mathcal{ACC}_{\text{P}}(\mathcal{K}) \subseteq \mathcal{INC}$. Thus, $\mathcal{K}$ is as expected inconsistent according to Definition 2.5.6.

Observe that our definition of inconsistency also captures cases where a given knowledge base $\mathcal{K}$ has no belief set at all. Formally, if $\mathcal{ACC}(\mathcal{K}) = \emptyset$, then $\mathcal{ACC}(\mathcal{K}) \subseteq \mathcal{INC}$ holds trivially. At a first glance, this may look like an overlooked technical detail. It is however intended and of importance for many non-monotonic frameworks, including ASP and AFs. The following example illustrates such a case.

Example 2.5.8 (Inconsistency in ASP)**.** Consider again the logic program P given as

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

As pointed out in Example 2.3.7, P has no answer set. Therefore,

$$\mathcal{ACC}_{\text{ASP}}(P) = \emptyset \subseteq \mathcal{INC}_{\text{ASP}}$$

and hence P is considered inconsistent as well.

In fact, this is a quite common reason for a logic program to be inconsistent. In the original formulation, a logic program only contains atoms of the form "a" and no literals of the form "$\neg a$", so there is no notion of inconsistent answer sets. Hence, a program of this kind can only be inconsistent when possessing no answer set. Inconsistency in AFs is similar. Due to our definition, a given AF F is considered inconsistent whenever $stb(F) = \emptyset$.

Example 2.5.9 (Inconsistency in AFs)**.** Let us consider again the AF $F = (A, R)$ depicted in Figure 2.3 above. Recall that we assume A to be implicit, so our knowledge base is $R = \{(a, b),\ (b, c),\ (c, c)\}$. Since F has no stable extension we obtain

$$\mathcal{ACC}_{\text{AAF}}(R) = \emptyset \subseteq \mathcal{INC}_{\text{AAF}}.$$

The framework is thus considered inconsistent, as expected.

Having established a formal meaning of inconsistency, our definition of consistency is straightforward.

Definition 2.5.10. A knowledge base $\mathcal{K}$ is *consistent* if $\mathcal{ACC}(\mathcal{K}) \not\subseteq \mathcal{INC}$, i. e., it is not inconsistent. We let $C(\mathcal{K})$ and $C_{max}(\mathcal{K})$ denote the set of all consistent and maximal consistent subsets of $\mathcal{K}$, respectively.

Following the usual terminology we will sometimes call a set $\mathcal{H} \in C(\mathcal{K})$ a *repair* and $\mathcal{S} = \mathcal{K} \setminus \mathcal{H}$ (with $\mathcal{H} \in C(\mathcal{K})$) a *diagnosis* for $\mathcal{K}$.

2.5.3 Monotony

As already mentioned, the central issue of this work is to investigate the behavior of inconsistency in non-monotonic logics. We thus require a formal definition of a monotonic logic in our setting. The intuitive understanding of monotony is that a conclusion which is inferred from a knowledge base $\mathcal{K}$ is never withdrawn due to additional information. We want to formalize this idea for our general logic while taking the usual reasoning modes into account. Our definition generalizes the one of [35]. Whereas the latter requires monotonic logics to associate unique belief sets to knowledge bases, our definition shows that a reasonable notion of monotony can be defined for logics with multiple belief sets.

Definition 2.5.11. A logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ is *skeptically monotonic* or simply *monotonic* whenever $\mathcal{K} \subseteq \mathcal{K}' \subseteq \mathcal{WF}$ implies:

- if $B' \in \mathcal{ACC}(\mathcal{K}')$ then $B \subseteq B'$ for some $B \in \mathcal{ACC}(\mathcal{K})$.

The name "skeptically" monotonic is motivated by the observation that in a skeptically monotonic logic, skeptical reasoning based on the intersection of belief sets is monotonic. This is formalized in Proposition 2.5.15 below. In this sense, the natural counterpart would be a notion of a "credulously" monotonic logic. This is indeed possible by requiring $B \subseteq B'$ for some $B' \in \mathcal{ACC}(\mathcal{K}')$ if $B \in \mathcal{ACC}(\mathcal{K})$. However, within the scope of this work the crucial monotony notion is the one given in Definition 2.5.11. The reason is that it ensures that conflicts within a knowledge base cannot be resolved by adding new information (see Lemma 2.5.18 below).

When there is no risk of confusion, we will call a knowledge base monotonic whenever its associated logic is. This is a slight abuse of terminology since monotony is a property of a logic, not of a knowledge base. However, leaving the actual logic implicit does no harm in many cases, and we prefer the simpler terminology.

Before we formally prove some properties of monotonic logics, let us consider some examples in order to get familiar with this notion.

Example 2.5.12 (Monotony in propositional logic)**.** Consider the propositional knowledge bases $\mathcal{K} = \{a \wedge b\}$ and $\mathcal{K}' = \{a \wedge b, b \rightarrow c.\}$. Observe that both $\mathcal{ACC}_{\text{P}}(\mathcal{K})$ as well as $\mathcal{ACC}_{\text{P}}(\mathcal{K}')$ are singletons. More precisely,

$$\mathcal{ACC}_{\text{P}}(\mathcal{K}) = \{\{a, b, a \wedge b, a \vee b, a \vee c, \ldots\}\},$$
$$\mathcal{ACC}_{\text{P}}(\mathcal{K}') = \{\{a, b, c, c, a \wedge b, a \wedge c, b \wedge c, \ldots\}\}.$$

So set $B = \{a, b, a \wedge b, a \vee b, a \vee c, \ldots\}$ and $B' = \{a, b, c, c, a \wedge b, a \wedge c, b \wedge c, \ldots\}$. Since $B \subseteq B'$ we see monotony according to Definition 2.5.11. As this is the case for any two propositional knowledge bases $\mathcal{K} \subseteq \mathcal{K}'$, we see that this logic is skeptically monotonic.

The following example illustrates that ASP is non-monotonic. The intuitive reason is as follows: Given two logic programs $P \subseteq P'$, it might happen that P' possesses a novel answer set in the sense that it is not a superset of an answer set of P. Within ASP, this is a common feature. It is thus sufficient to consider a rather simple example.

Example 2.5.13 (Monotony in ASP)**.** Consider $P \subseteq P'$ given as follows:

$P:$	$a \leftarrow \text{not } b.$	$P':$	$a \leftarrow \text{not } b.$
			$b \leftarrow \text{not } a.$

We have

$$\mathcal{ACC}_{\text{ASP}}(P) = \{\{a\}\}, \qquad \mathcal{ACC}_{\text{ASP}}(P') = \{\{a\}, \{b\}\}.$$

If we set $B' = \{b\} \in \mathcal{ACC}_{\text{ASP}}(P')$, then there is no $B \in \mathcal{ACC}_{\text{ASP}}(P)$ with $B \subseteq B'$. ASP is thus not skeptically monotonic.

It is also a straightforward observation that AFs are non-monotonic as well.

Example 2.5.14 (Monotony in AFs). Recall the AF from Figure 2.3 which we represent as knowledge base $R = \{(a,b),(b,c),(c,c)\}$. Consider $R' = R \cup \{(a,c)\}$ which yields an AF with one additional attack. We see $\mathcal{ACC}_{\text{AAF}}(R) = \{\}$ and $\mathcal{ACC}_{\text{AAF}}(R') = \{\{a\}\}$. If we set $B' = \{a\}$, then there is no $B \in \mathcal{ACC}_{\text{AAF}}(R)$ satisfying $B \subseteq B'$ which is trivial since $\mathcal{ACC}_{\text{AAF}}(R)$ is empty.

As already mentioned, skeptical monotony guarantees monotony of skeptical reasoning. Formally, we have:

Proposition 2.5.15. *Let L be a skeptically monotonic logic. If $\mathcal{K}$ and $\mathcal{K}'$ are consistent knowledge bases and $\mathcal{K} \subseteq \mathcal{K}'$, then*

$$\bigcap_{B \in \mathcal{ACC}(\mathcal{K})} B \subseteq \bigcap_{B' \in \mathcal{ACC}(\mathcal{K}')} B'.$$

Proof. Let $p \in \bigcap_{B \in \mathcal{ACC}(\mathcal{K})} B$, i. e., $p \in B$ for each $B \in \mathcal{ACC}(\mathcal{K})$. Now consider an arbitrary B' with $B' \in \mathcal{ACC}(\mathcal{K}')$. Due to skeptical monotony, there is a $B \in \mathcal{ACC}(\mathcal{K})$ with $B \subseteq B'$. We thus have $p \in B \subseteq B'$, so $p \in B'$. Since B' was an arbitrary set in $\mathcal{ACC}(\mathcal{K}')$, $p \in \bigcap_{B' \in \mathcal{ACC}(\mathcal{K}')} B'$. □

Let us now collect some examples of monotonic logics. As we already mentioned, it is easy to extend Example 2.5.12 to a general statement. We thus see:

Proposition 2.5.16. *The propositional logic*

$$L_P^A = \left(\mathcal{WF}_P^A, \mathcal{BS}_P^A, \mathcal{INC}_P^A, \mathcal{ACC}_P^A\right)$$

is skeptically monotonic.

We expect the presence of default negation to cause non-monotony of ASP. Indeed, without occurrences of "not", ASP is monotonic.

Proposition 2.5.17. *Let $\mathcal{WF}_{ASP-not}^A$ consist of rules of the form* (2.1) *with $m = n$, i. e., rules r such that $neg(r) = \emptyset$. The logic*

$$L_{ASP-not}^A = \left(\mathcal{WF}_{ASP-not}^A, \mathcal{BS}_{ASP}^A, \mathcal{INC}_{ASP}^A, \mathcal{ACC}_{ASP}^A\right)$$

is skeptically monotonic.

Proof. Let $P, P' \subseteq \mathcal{WF}_{\text{ASP-not}}^A$. Let $P \subseteq P'$. Let B' be an answer set of P', i. e., $B' \in \mathcal{ACC}_{\text{ASP}}^A(P')$. Since B' is a minimal model of P', it is a model of $P \subseteq P'$ as well. Now there is a set $B \subseteq B'$ which is a minimal model of P. Since this means B is an answer set of P, $B \in \mathcal{ACC}_{\text{ASP}}^A(P)$ follows. □

Let us now formally state that in a monotonic logic an inconsistent knowledge base cannot be turned into a consistent one by moving to a superset.

Lemma 2.5.18. *Let $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ be monotonic and $\mathcal{K} \subseteq \mathcal{K}'$. If $\mathcal{K}$ is inconsistent, then so is $\mathcal{K}'$.*

Proof. Let $\mathcal{K}$ be inconsistent, i. e., $\mathcal{ACC}(\mathcal{K}) \subseteq \mathcal{INC}$. Let $\mathcal{K} \subseteq \mathcal{K}'$. If $B' \in \mathcal{ACC}(\mathcal{K}')$, then $B \subseteq B'$ for a $B \in \mathcal{ACC}(\mathcal{K})$. However, $B \in \mathcal{ACC}(\mathcal{K})$ implies $B \in \mathcal{INC}$ and since $\mathcal{INC}$ is upward closed, $B' \in \mathcal{INC}$. Thus, $\mathcal{ACC}(\mathcal{K}') \subseteq \mathcal{INC}$. □

Throughout this work, our results regarding monotonic logics depend especially on the above Lemma 2.5.18 which states that inconsistency survives moving to supersets. Regarding inconsistency in non-monotonic logics, the loss of this property is the central issue we need to handle.

Chapter 3

Duality Characterizations for Non-monotonic Logics

In [96], Reiter points out a duality between maximal consistent and minimal inconsistent subsets of a knowledge base – the well-known hitting set duality. Within the context of propositional logic Reiter's result reads: A subset $\mathcal{S}$ of a knowledge base $\mathcal{K}$ is a minimal hitting set of $I_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$. This result establishes a connection between the inconsistent and the consistent subsets of a given knowledge base. We want to emphasize that the above statement is an equivalence, and hence the set $C_{max}(\mathcal{K})$, the set of maximal consistent subsets of $\mathcal{K}$, is characterized. Besides being interesting from a structural point of view, this result has various applications. For example, many algorithms and systems for enumerating minimal inconsistent sets –see [8; 78; 79]– utilize Reiter's duality. The reader is referred to Section 3.5 for a more comprehensive discussion of related work.

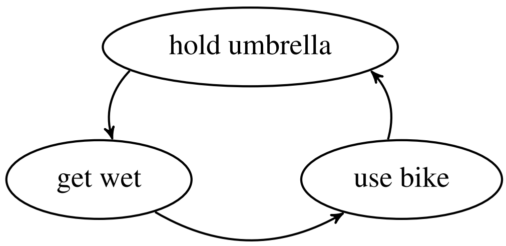
Because of the importance of this result and in order to understand the behavior and structure of inconsistent subsets of a knowledge base, our main goal of this chapter is to generalize the hitting set duality to non-monotonic logics. There are various situations we are going to cover. First, we investigate a given non-monotonic knowledge base $\mathcal{K}$, aiming at the maximal consistent subsets $C_{max}(\mathcal{K})$ as usual. In order to characterize those via a hitting set duality, ordinary inconsistency will turn out to be insufficient. We will thus introduce a refinement of this notion, called *strong inconsistency*, which will play a central role for the remainder of this work. Moreover, motivated by the observation that supersets of a given knowledge base may resolve conflicts, we investigate repairs based on adding information, in contrast to the notion of maximal consistent subsets which is based on removing information. We will see that the notions we introduce are symmetric in their fashion and amplify each other in a natural way. By applying our results appropriately, we also obtain characterizations for repairs based on refining instead of deleting or adding formulas, even without the necessity for novel proofs. The latter observation shall also demonstrate the versitality of the hitting set characterizations we are going to establish within the subsequent sections.

We will finish this chapter with a discussion on possibly infinite knowledge bases. Even though infinite knowledge or data does not occur in real world applications, oftentimes it is impossible or at least unclear how to establish an upper bound for the size of a knowledge base. In this case, we need to assume it is infinite or arbitrarily large. Therefore, we also want to provide formal results for knowledge bases of infinite size.

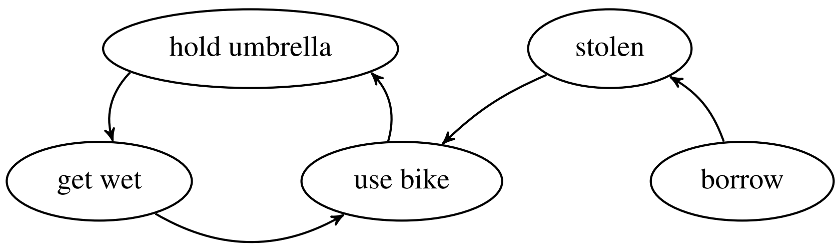
3.1 Strong Inconsistency

In classical frameworks like propositional logic, inconsistency does not only render reasoning meaningless as anything can be derived from an inconsistent knowledge base, but is also permanent. More precisely, an inconsistent knowledge base $\mathcal{K}$ cannot be turned into a consistent one via additional formulas. Inconsistency is thus quite straightforward and seeking for the minimal conflicts $I_{min}(\mathcal{K})$ suffices when trying to tackle it as formalized in [96] by Reiter's hitting set duality. In non-monotonic logics, inconsistency oftentimes stems from an incorrect use of default assumptions as the famous penguin example illustrates. However, default assumptions can typically be overwritten which leads to resolution of the conflicts. Inconsistency in non-monotonic logics appears thus hardly comparable to inconsistency in, e.g., propositional logic. To illustrate the dynamic behavior of this notion in non-monotonic frameworks, let us consider the following example.

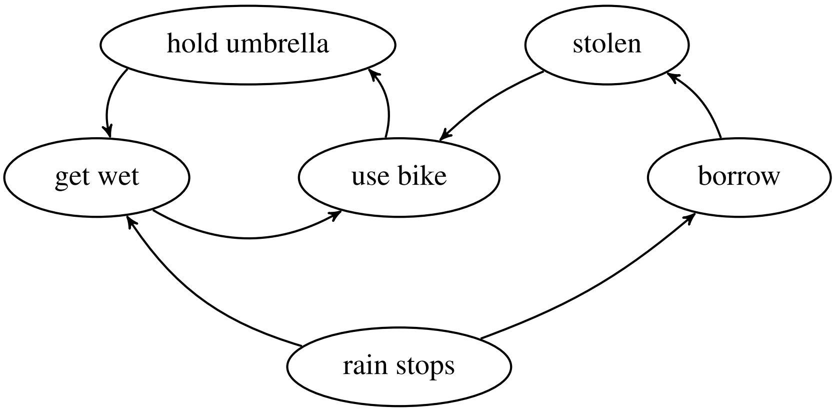
Example 3.1.1. Assume an agent came to work by bike in the morning, but now that she is getting ready to leave, it starts raining. Since walking home would take much time, using the bike is the preferred option. However, holding an umbrella while cycling is dangerous, but without it, our agent faces a hard time, gets wet (and maybe even a cold). This situation can be captured with the following concise AF:



Since this AF does not possess a stable extension, it is not possible for our agent to formally decide whether to use the bike or not. Unfortunately, the agent realizes that her bike got stolen. Besides being bad news, this would in principle resolve the conflict: Without a bike, the only option is walking home holding the umbrella. However, she could also borrow a bike from a colleague who already decided to stay at the office overnight. This yields the following AF:



We see that the option to borrow a bike brings back the initial situation. Assume now that the rain stops. Suddenly the bike is the preferred option again, but also for the agent's colleague. This new situation is modeled by the following AF:



The unique stable extension of this AF formalizes that there is no bike which can be used and no rain to worry about anymore.

The previous example demonstrates that a non-monotonic knowledge base might be quite messy when it comes to consistent and inconsistent subsets. The AF itself is consistent, but while constructing it we found two inconsistent subframeworks. It appears therefore hard to identify the actual reasons for the presence or absence of stable extensions. This raises the following questions:

- How does inconsistency in non-monotonic logics behave? Is it possible to identify some structure?
- How meaningful is the set $I_{min}(\mathcal{K})$, i. e., the set of minimal inconsistent subsets?
- Is it possible to generalize Reiter's hitting set duality to non-monotonic logics? If so, which adjustments to the established definitions are required?

In this section, we motivate and develop the answer to those questions, called *strong inconsistency*. Although simple in its spirit, it facilitates a generalization of Reiter's duality to *arbitrary* logics. However, this is not the only reason to consider this extension of ordinary inconsistency. It will turn out to be a key tool to understand the structural behavior of knowledge bases with respect to inconsistency. With this in mind, further extensions and generalizations of the achieved results can be obtained in a similar fashion and the required notions are natural and symmetric to each other. Moreover, strong inconsistency will play a central role in Chapter 4, where we discuss measuring inconsistency in non-monotonic logics.

3.1.1 The Classical Hitting Set Dualtiy

In order to motivate and understand the necessity of a refined notion of inconsistency, let us recall Reiter's hitting set duality from [96, Theorem 4.4]. It is stated within the setting of diagnoses of a system description. In case of arbitrary monotonic logics, it is folklore. We state it explicitly here in order to demonstrate how to phrase it within our setting. Although this theorem is a corollary of Proposition 3.1.11, second item, and Theorem 3.1.12 from below, we give a straight proof of this result. This may also help to familiarize with the notions we have established so far.

Theorem 3.1.2 (MinHS duality)**.** *Let $\mathcal{K}$ be a monotonic knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $I_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$.*

Proof. "⇒": Let $\mathcal{S}$ be a minimal hitting set of $I_{min}(\mathcal{K})$. For the sake of contradiction assume that $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$ is not maximal consistent, i. e., $\mathcal{H} \notin C_{max}(\mathcal{K})$.

First, assume $\mathcal{H}$ is inconsistent. Since $\mathcal{K}$ is finite, $\mathcal{H} \subseteq \mathcal{K}$ is finite as well and thus contains a minimal inconsistent subset $\mathcal{H}' \subseteq \mathcal{H}$. Due to $\mathcal{H}' \subseteq \mathcal{H} = \mathcal{K} \setminus \mathcal{S}$ we have $\mathcal{S} \cap \mathcal{H}' = \emptyset$ yielding a contradiction as $\mathcal{S}$ was assumed to be a hitting set of $I_{min}(\mathcal{K})$. Hence, $\mathcal{H} \in C(\mathcal{K})$.

Now assume $\mathcal{H}$ is consistent, but not maximal, so there is a consistent set $\mathcal{H}'$ with $\mathcal{H} \subsetneq \mathcal{H}' \subseteq \mathcal{K}$. Again due to finiteness, we may assume $\mathcal{H}'$ is maximal consistent, i. e., $\mathcal{H}' \in C_{max}(\mathcal{K})$. We claim that in this case, $\mathcal{S}' := \mathcal{K} \setminus \mathcal{H}' \subsetneq \mathcal{K} \setminus \mathcal{H} = \mathcal{S}$ is a hitting set of $I_{min}(\mathcal{K})$ as well, contradicting minimality of $\mathcal{S}$. So assume the contrary, i. e., assume there is a set $\mathcal{I} \in I_{min}(\mathcal{K})$ with $\mathcal{I} \cap \mathcal{S}' = \emptyset$. We infer $\mathcal{I} \subseteq \mathcal{K} \setminus \mathcal{S}'$. Now

$$\mathcal{I} \subseteq \mathcal{K} \setminus \mathcal{S}' = \mathcal{H}'$$

so there is an inconsistent set $\mathcal{I}$ with $\mathcal{I} \subseteq \mathcal{H}'$. Since $\mathcal{K}$ is monotonic, we may apply Lemma 2.5.18 which implies inconsistency of $\mathcal{H}'$. This contradicts the assumed consistency of $\mathcal{H}'$. Hence, $\mathcal{H}$ must be maximal consistent.

"⇐": Let $\mathcal{H} \subseteq \mathcal{K}$ be a maximal consistent set. Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$. If $\mathcal{S}$ is no hitting set of $I_{min}(\mathcal{K})$, then we see as above that $\mathcal{H}$ contains an inconsistent subset, yielding a contradiction. Hence, $\mathcal{S}$ is a hitting set of $I_{min}(\mathcal{K})$. Now assume there is a (w. l. o. g. minimal) set $\mathcal{S}' \subsetneq \mathcal{S}$ that is a hitting set of $I_{min}(\mathcal{K})$ as well. Then $\mathcal{K} \setminus \mathcal{S}' \in C(\mathcal{K})$ as already shown above. However, $\mathcal{S}' \subsetneq \mathcal{S}$ implies $\mathcal{K} \setminus \mathcal{S} \subsetneq \mathcal{K} \setminus \mathcal{S}'$. Hence, $\mathcal{K} \setminus \mathcal{S}$ is not maximal in $C(\mathcal{K})$, which is again a contradiction. □

Let us illustrate this results within the setting of propositional logic.

Example 3.1.3. Consider again $\mathcal{K} = \{a,\ a \rightarrow b,\ \neg b,\ c,\ \neg c\}$. As already discussed, $I_{min}(\mathcal{K}) = \{\{a,\ a \rightarrow b,\ \neg b\},\ \{c,\ \neg c\}\}$. We see that there are six minimal hitting sets of $I_{min}(\mathcal{K})$, namely $\{a, c\}$, $\{a, \neg c\}$, $\{a \rightarrow b, c\}$, $\{a \rightarrow b, \neg c\}$, $\{\neg b, c\}$, $\{\neg b, \neg c\}$. Consider $\{a, c\}$. Indeed, $\mathcal{K} \setminus \{a, c\} = \{a \rightarrow b,\ \neg b,\ \neg c\}$ is maximal consistent. One can verify that $\mathcal{K}$ has exactly the six maximal consistent subsets which can be obtained by removing the hitting sets of $I_{min}(\mathcal{K})$.

The process of turning an inconsistent knowledge base into maximal consistent ones is depicted in Figure 3.1.

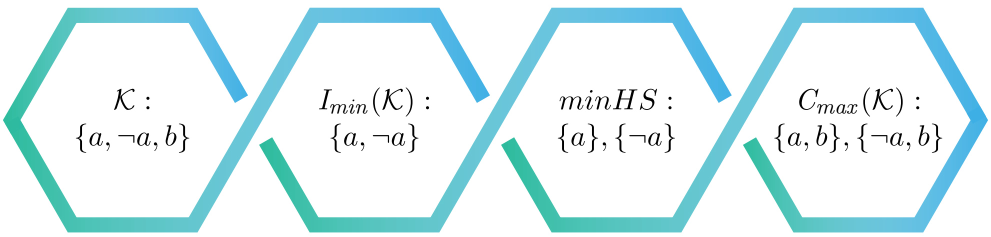


Figure 3.1: Reiter's hitting set duality in a nutshell: Given an inconsistent knowledge base, we first compute the minimal conflicts, and then proceed with computing their minimal hitting sets. This way we find the maximal consistent subsets.

For non-monotonic logics this result does not apply anymore, because a knowledge base may contain inconsistent subsets. This yields the following undesired behavior: It might happen that

- for a minimal hitting set $\mathcal{S}$ of $I_{min}(\mathcal{K})$, the set $\mathcal{K} \setminus \mathcal{S}$ is not maximal consistent,
- there are sets $\mathcal{H} \in C_{max}(\mathcal{K})$ which are not of the form $\mathcal{K} \setminus \mathcal{S}$ for a minimal hitting set $\mathcal{S}$ of $I_{min}(\mathcal{K})$.

The following examples illustrate this. We may give a quite simple one for the first item.

Example 3.1.4. Let P be the consistent program

$$P: \qquad a \leftarrow \text{not } a. \qquad a.$$

with $I_{min}(P) = \{a \leftarrow \text{not } a.\}$. Of course, $P \setminus \{a \leftarrow \text{not } a.\}$ is not maximal consistent.

For the second item from above we consider again our running examples.

Example 3.1.5. Recall the logic program P given as follows:

$$\begin{array}{lrr} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

In Example 2.3.7 we noticed that P is inconsistent. Clearly,

$$I_{min}(P) = \{\{c \leftarrow \text{not } b., \ \neg c \leftarrow \text{not } b.\}\}.$$

By removing minimal hitting sets of $I_{min}(P)$ from P we obtain

$$\begin{array}{rccr} P \setminus \{c \leftarrow \text{not } b.\}: & a \vee b. & a \leftarrow b. & \neg c \leftarrow \text{not } b. \\ P \setminus \{\neg c \leftarrow \text{not } b.\}: & a \vee b. & a \leftarrow b. & c \leftarrow \text{not } b. \end{array}$$

Both are indeed maximal consistent subsets of P, but the third maximal consistent subset

$$P \setminus \{a \leftarrow b.\}: \qquad a \vee b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

is missing.

A similar observation can be made for abstract argumentation frameworks:

Example 3.1.6. Recall the AF $F = (A, R)$ depicted in Figure 2.3 with the implicitly given set $A = \{a, b, c\}$ and thus our knowledge base is $R = \{(a, b), \ (b, c), \ (c, c)\}$. Clearly $I_{min}(R) = \{(c, c)\}$, so removing the obvious hitting set $\{(c, c)\}$ yields the consistent AF $F_1 = (A, R_1)$ with $R_1 = \{(a, b), (b, c)\}$. However, this way we do not find the consistent AF $F_2 = (A, R_2)$ with $R_2 = \{(b, c), (c, c)\}$:

Figure 3.2: The consistent argumentation frameworks F_1 and F_2 from Example 3.1.6

3.1.2 Refining Inconsistency

In order to generalize the hitting set duality to non-monotonic logics, we require a general notion of inconsistency which is comparable to minimal inconsistency in monotonic logics. A minimal inconsistent subset $\mathcal{H}$ of a monotonic knowledge base $\mathcal{K}$ is characterized by the following properties:

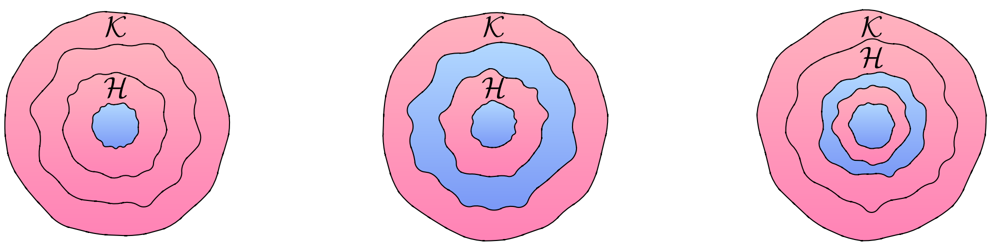
- $\mathcal{H}$ is inconsistent; more precisely, due to monotony of $\mathcal{K}$, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent,
- $\mathcal{H}$ is minimal with the above property.

In a non-monotonic logic, the observation made in the first item is not true anymore. In order to simulate this behavior we require this property as an axiom since it is no longer guaranteed. This yields our central notion called strong inconsistency.

Definition 3.1.7. For $\mathcal{H}, \mathcal{K} \subseteq \mathcal{WF}$ with $\mathcal{H} \subseteq \mathcal{K}$ we call $\mathcal{H}$ *strongly $\mathcal{K}$-inconsistent* if $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ implies $\mathcal{H}'$ is inconsistent. If there is no risk of confusion, we call $\mathcal{H}$ *strongly inconsistent* for short. Denote by $SI(\mathcal{K})$ and $SI_{min}(\mathcal{K})$ the set of all strongly inconsistent and minimal strongly inconsistent subsets of $\mathcal{K}$, respectively.

In other words, a subset of a knowledge base $\mathcal{K}$ is strongly $\mathcal{K}$-inconsistent if all its supersets within $\mathcal{K}$ are inconsistent as well. Intuitively, one can think of a conflict that cannot be resolved by formulas in $\mathcal{K}$ itself.

A comparison between mere and strong inconsistency is depicted in Figure 3.3. Circles shall correspond to subsets of $\mathcal{K}$, consistent ones filled blue, inconsistent ones filled red. We see that for a minimal inconsistent subset $\mathcal{H}$ of a knowledge base $\mathcal{K}$, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent as well (1), whereas this is not necessarily the case for a non-monotonic logic. This is indicated by the blue circle around $\mathcal{H}$ (2). The last picture (3) illustrates a minimal strongly inconsistent subset $\mathcal{H}$. As in the monotonic case, there is no blue circle between $\mathcal{H}$ and $\mathcal{K}$.



$\mathcal{H}$ minimal inconsistent (monotonic)	$\mathcal{H}$ minimal inconsistent (non-monotonic)	$\mathcal{H}$ minimal strongly inconsistent (non-monotonic)
(1)	(2)	(3)

Figure 3.3: Minimal inconsistency vs. minimal strong inconsistency

Example 3.1.8. Consider again $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$. Recall that $\{c, \neg c\}$ is an inconsistent subset of $\mathcal{K}$. Of course, any set $\mathcal{H}'$ with $\{c, \neg c\} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent as well. Hence, $\{c, \neg c\}$ is strongly $\mathcal{K}$-inconsistent.

More generally, Lemma 2.5.18 ensures that mere and strong $\mathcal{K}$-inconsistency coincide whenever our logic is monotonic (cf. Proposition 3.1.11 below). Let us thus consider examples involving non-monotonic logics to see our definition at work.

Example 3.1.9. Consider again the logic program P from above.

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

Recall that $H = \{c \leftarrow \text{not } b., \neg c \leftarrow \text{not } b.\}$ is an inconsistent subset of P. There is a consistent program H' with $H \subseteq H' \subseteq P$, namely

$$H': \qquad a \vee b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

Hence, H is not strongly inconsistent. However,

$$H'': \qquad a \leftarrow b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

is strongly inconsistent; even minimal. In particular, there is no other minimal strongly inconsistent set, therefore

$$SI_{min}(P) = \{\{a \leftarrow b., \; c \leftarrow \text{not } b., \; \neg c \leftarrow \text{not } b.\}\}.$$

Example 3.1.10. Recall the argumentation framework depicted in Figure 2.3 which corresponds to the knowledge base $R = \{(a, b), (b, c), (c, c)\}$. We already found the inconsistent subset $\{(c, c)\}$. However, the framework over $A = \{a, b, c\}$ with attacks $\{(b, c), (c, c)\}$ has a stable extension, namely $\{b\}$. Hence, the set $\{(c, c)\}$ of attacks is not strongly inconsistent. The reader may verify that

$$SI_{min}(R) = \{\{(a, b), \; (c, c)\}\}.$$

Let us now collect some basic properties of $SI(\mathcal{K})$ and $SI_{min}(\mathcal{K})$ in monotonic and non-monotonic logics.

Proposition 3.1.11. *Let $\mathcal{K}$ be a knowledge base.*

(a) If $\mathcal{K}$ is monotonic, then $I(\mathcal{K}) = SI(\mathcal{K})$.

(b) If $\mathcal{K}$ is monotonic, then $I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$.

(c) $\mathcal{K}$ is inconsistent iff $SI(\mathcal{K}) \neq \emptyset$ iff $\mathcal{K} \in SI(\mathcal{K})$.

(d) If $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent and $\mathcal{H} \subseteq \mathcal{K}' \subseteq \mathcal{K}$, then $\mathcal{H}$ is strongly $\mathcal{K}'$-inconsistent.

Proof. Let L be a monotonic logic and let $\mathcal{K}$ be a knowledge base.

(a): The inclusion $I(\mathcal{K}) \supseteq SI(\mathcal{K})$ is clear. For $I(\mathcal{K}) \subseteq SI(\mathcal{K})$ let $\mathcal{H} \subseteq \mathcal{K}$ with $\mathcal{H} \in I(\mathcal{K})$. Due to Lemma 2.5.18, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}'$ is inconsistent as well. In particular, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. Thus, $\mathcal{H}$ is strongly inconsistent.

(b): This is a corollary of the first item.

Now let L be an arbitrary logic. As above let $\mathcal{K}$ be a knowledge base.

(c): If $\mathcal{K}$ is inconsistent, then $\mathcal{K} \in SI(\mathcal{K})$ and hence, $SI(\mathcal{K}) \neq \emptyset$. If $SI(\mathcal{K}) \neq \emptyset$, then there is a set $\mathcal{H}$ such that each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. In particular, $\mathcal{K} \in SI(\mathcal{K})$. By definition of $SI(\mathcal{K})$, $\mathcal{K}$ is inconsistent if $\mathcal{K} \in SI(\mathcal{K})$.

(d): Let $\mathcal{H} \in SI(\mathcal{K})$. Hence, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. In particular for $\mathcal{K}' \subseteq \mathcal{K}$, each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}' \subseteq \mathcal{K}$ is also inconsistent, i. e., $\mathcal{H} \in SI(\mathcal{K}')$. □

It is also easy to see that a minimal strongly inconsistent set $\mathcal{H} \in SI_{min}(\mathcal{K})$ always contains a minimal inconsistent subset, i. e., there is $\mathcal{H}' \in I_{min}(\mathcal{K})$ with $\mathcal{H}' \subseteq \mathcal{H}$. Otherwise, $\mathcal{H}$ would not contain a conflict. Thus, a strongly inconsistent set consists of a set of formulas that yield a conflict and some additional formula(s) that make sure the conflict cannot be resolved within $\mathcal{K}$:

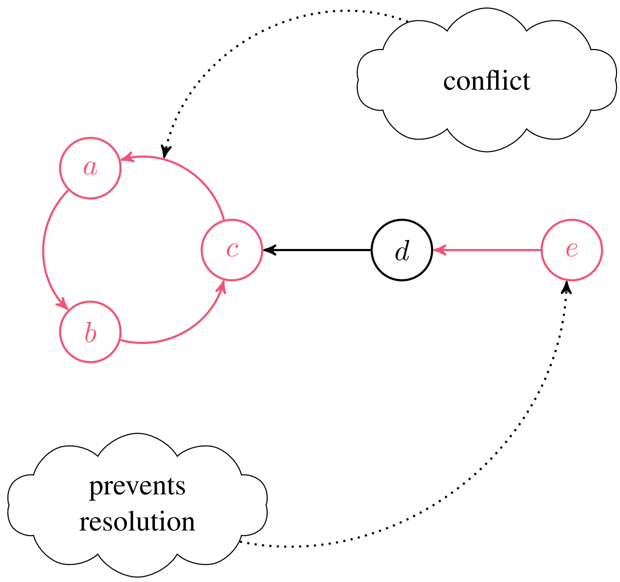


Figure 3.4: Depiction of a strongly inconsistent set (red): It consists of an odd loop and some additional arguments resp. attacks that prevent resolution of the conflict.

Until now, we have developed the notion strong inconsistency for non-monotonic logics which is comparable to mere inconsistency for monotonic logics (recall Figure 3.3). The next step is to investigate which subsets of $\mathcal{K}$ are characterized by the minimal hitting sets of $SI_{min}(\mathcal{K})$. Due to the similarity to Theorem 3.1.2 we would expect some notion of consistency. As it turns out, no adjustment to this notion is required.

Theorem 3.1.12 (Generalized MinHS duality)**.** *Let $\mathcal{K}$ be a knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$.*

Proof. "⇒": Let $\mathcal{S}$ be a minimal hitting set of $SI_{min}(\mathcal{K})$ and let $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$. First, we show that any set $\mathcal{H}'$ with $\mathcal{H} \subsetneq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. Such $\mathcal{H}'$ is of the form

$$\mathcal{H}' = \mathcal{H} \cup \mathcal{I} \text{ with } \emptyset \neq \mathcal{I} \subseteq \mathcal{K} \setminus \mathcal{H}.$$

Hence, it holds that $\emptyset \neq \mathcal{I} \subseteq \mathcal{S}$. Since $\mathcal{S}$ was assumed to be a minimal hitting set of $SI_{min}(\mathcal{K})$, $\mathcal{S} \setminus \mathcal{I}$ is not a hitting set of $SI_{min}(\mathcal{K})$. This implies that $\mathcal{K} \setminus (\mathcal{S} \setminus \mathcal{I})$ contains a strongly inconsistent set. The set $\mathcal{H}' = \mathcal{H} \cup \mathcal{I}$ is of the form

$$\mathcal{H}' = \mathcal{H} \cup \mathcal{I} = (\mathcal{K} \setminus \mathcal{S}) \cup \mathcal{I} = \mathcal{K} \setminus (\mathcal{S} \setminus \mathcal{I}).$$

Hence $\mathcal{H}'$ contains a strongly inconsistent set, implying that $\mathcal{H}'$ is inconsistent by definition. Since $\mathcal{H}'$ was an arbitrary set of satisfying $\mathcal{H} \subsetneq \mathcal{H}' \subseteq \mathcal{K}$ we see that this is the case for any proper superset $\mathcal{H}'$ of $\mathcal{H}$. So as a first step we obtained:

$$\mathcal{H} \subsetneq \mathcal{H}' \subseteq \mathcal{K} \Rightarrow \mathcal{H}' \text{ is inconsistent.} \tag{3.1}$$

Now we show $\mathcal{H} \in C(\mathcal{K})$. Then, maximality follows from (3.1). Let us assume $\mathcal{H}$ is inconsistent. Then, $\mathcal{H}$ is strongly inconsistent due to (3.1). Due to finiteness of $\mathcal{H} \subseteq \mathcal{K}$, $\mathcal{H}$ contains a minimal strongly inconsistent subset $\mathcal{H}'' \subseteq \mathcal{H}$, and since

$$\mathcal{H}'' \subseteq \mathcal{H} = \mathcal{K} \setminus \mathcal{S}$$

we have $\mathcal{S} \cap \mathcal{H}'' = \emptyset$, yielding a contradiction as $\mathcal{S}$ was assumed to be a hitting set of $SI_{min}(\mathcal{K})$. Hence, $\mathcal{H} \in C(\mathcal{K})$. Together with (3.1), we obtain that $\mathcal{H} \in C_{max}(\mathcal{K})$.

"$\Leftarrow$": Let $\mathcal{H} \subseteq \mathcal{K}$ be a maximal consistent set. Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$. If $\mathcal{S}$ is no hitting set of $SI_{min}(\mathcal{K})$, then we see as above that $\mathcal{H}$ contains a strongly inconsistent set, yielding a contradiction. Hence, $\mathcal{S}$ is a hitting set of $SI_{min}(\mathcal{K})$. Now assume that there is a (w. l. o. g. minimal) set $\mathcal{S}' \subsetneq \mathcal{S}$ that is a hitting set of $SI_{min}(\mathcal{K})$ as well. Then, $\mathcal{K} \setminus \mathcal{S}' \in C(\mathcal{K})$ as already shown above. However, $\mathcal{S}' \subsetneq \mathcal{S}$ implies $\mathcal{K} \setminus \mathcal{S} \subsetneq \mathcal{K} \setminus \mathcal{S}'$. Hence, $\mathcal{K} \setminus \mathcal{S}$ is not maximal in $C(\mathcal{K})$, a contradiction. □

Theorem 3.1.12 suggests that strong inconsistency can indeed play a similar role in non-monotonic frameworks as ordinary inconsistency does in monotonic ones. Even though restoring consistency in one part of $\mathcal{K}$ by removing a formula may render another part of $\mathcal{K}$ inconsistent, we can resolve inconsistency as in the monotonic case using the notion of strong inconsistency.

Let us consider again our examples to illustrate the theorem in monotonic and non-monotonic frameworks.

Example 3.1.13. Consider again $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$. As already pointed out, we have $I_{min}(\mathcal{K}) = \{\{a,\ a \to b,\ \neg b\},\ \{c,\ \neg c\}\}$. Due to monotony of propositional logic, $SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$, so we obtain the same hitting sets as in Example 3.1.3 above.

Example 3.1.14. Consider the logic program P:

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

As we saw in Example 3.1.9,

$$SI_{min}(P) = \{\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}\}.$$

Since P is inconsistent, at least one rule needs to be removed in order to obtain a maximal consistent subprogram. Let us consider the three possibilities given by the hitting sets of $SI_{min}(P)$:

$$P \setminus \{c \leftarrow \text{not } b.\}: \qquad a \vee b. \qquad a \leftarrow b. \qquad \neg c \leftarrow \text{not } b.$$
$$P \setminus \{\neg c \leftarrow \text{not } b.\}: \qquad a \vee b. \qquad a \leftarrow b. \qquad c \leftarrow \text{not } b.$$
$$P \setminus \{a \leftarrow b.\}: \qquad a \vee b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

Since removing "$a \vee b$." does not yield a consistent subprogram, the collection above is indeed the set $C_{max}(P)$, as stated in Theorem 3.1.12.

Example 3.1.15. Now consider the argumentation framework (A, R) from before with our representation $R = \{(a, b), (b, c), (c, c)\}$. Recall that we obtained

$$SI_{min}(R) = \{\{(a, b), (c, c)\}\}$$

in Example 3.1.10. Indeed,

$$C_{max}(R) = \{\{(b, c), (c, c)\},\ \{(a, b), (b, c)\}\}.$$

The former set of attacks yields an AF with $\{b\}$ as a stable extension, the latter an AF with $\{a, c\}$ (recall Figure 3.2).

We continue our investigation of the structure of knowledge bases and how strong inconsistency transfers properties from monotonic to non-monotonic logics. So far, we used hitting sets of $SI_{min}(\mathcal{K})$ in order to characterize $C_{max}(\mathcal{K})$. However, consistency and inconsistency are complementary concepts, and minimality and maximality can be exchanged: For example, a set $\mathcal{H}$ is maximal consistent if $\mathcal{D}$ with $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ is minimal such that $\mathcal{K}$ is consistent. So it is a natural question whether we can reverse the hitting set duality: Is it possible to characterize $SI_{min}(\mathcal{K})$ via hitting sets of $C_{max}(\mathcal{K})$?

For monotonic logics, the affirmative answer to this question has been stated in [30], Theorem 4.5, item (d). To be precise, we do not require hitting sets of $C_{max}(\mathcal{K})$, but $coC_{max}(\mathcal{K})$, defined as expected:

Definition 3.1.16. A set $\overline{\mathcal{H}} \subseteq \mathcal{K}$ is in $coC_{max}(\mathcal{K})$ if there is a set $\mathcal{H} \in C_{max}(\mathcal{K})$ such that $\overline{\mathcal{H}} = \mathcal{K} \setminus \mathcal{H}$.

Phrased within our setting, the duality result from [30] reads as follows.

Theorem 3.1.17. *Let $\mathcal{K}$ be a monotonic knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $coC_{max}(\mathcal{K})$ if and only if $\mathcal{S} \in I_{min}(\mathcal{K})$.*

So we can indeed characterize $I_{min}(\mathcal{K})$ via a hitting set duality. This result emphasizes the close link between minimal inconsistency and maximal consistency. We want to investigate whether this works similarly straightforward for non-monotonic logics as well.

As it turns out, there is a much deeper connection to be discovered. Independent of logics and (in)consistency, there is a simple lemma answering our questions:

Lemma 3.1.18. *[26] Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ be a set of sets with $X_i \not\subseteq X_j$ for $i \neq j$. Then $minHS(minHS(\mathcal{X})) = \mathcal{X}$.*

Now consider an arbitrary (possibly non-monotonic) knowledge base $\mathcal{K}$. Since $\mathcal{K}$ is finite, we may apply Lemma 3.1.18 to $SI_{min}(\mathcal{K})$. Note that our hitting set duality from Theorem 3.1.12 reads

$$minHS(SI_{min}(\mathcal{K})) = coC_{max}(\mathcal{K}).$$

We infer by moving to minimal hitting sets on both sides

$$minHS(minHS(SI_{min}(\mathcal{K}))) = minHS(coC_{max}(\mathcal{K}))$$

and after applying Lemma 3.1.18 to the left-hand side we obtain

$$SI_{min}(\mathcal{K}) = minHS(coC_{max}(\mathcal{K})).$$

Hereby we proved the desired duality characterization:

Theorem 3.1.19. *Let $\mathcal{K}$ be a knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $coC_{max}(\mathcal{K})$ if and only if $\mathcal{S} \in SI_{min}(\mathcal{K})$.*

Let us reconsider our examples in order to familiarize with this result.

Example 3.1.20. For the propositional knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$ we have

$$\begin{aligned} C_{max}(\mathcal{K}) = \{&\{a \to b, \neg b, c\},\ \{a, \neg b, c\},\ \{a, a \to b, c\}, \\ &\{a \to b, \neg b, \neg c\},\ \{a, \neg b, \neg c\},\ \{a, a \to b, \neg c\}\} \end{aligned}$$

and thus

$$\begin{aligned} coC_{max}(\mathcal{K}) = \{&\{a, \neg c\},\ \{a \to b, \neg c\},\ \{\neg b, \neg c\} \\ &\{a, c\},\ \{a \to b, c\},\ \{\neg b, c\}\} \end{aligned}$$

We see that $coC_{max}(\mathcal{K})$ possesses two minimal hitting sets, namely $\{a, a \to b, \neg b\}$ and $\{c, \neg c\}$. Indeed, these are the minimal inconsistent subsets of $\mathcal{K}$.

Example 3.1.21. Consider again the program P

$$\begin{array}{lll} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

again. As already pointed out, we have

$$\begin{array}{ll} C_{max}(P): & coC_{max}(P): \\ \{a \vee b.,\ a \leftarrow b.,\ c \leftarrow \text{not } b.\} & \{\neg c \leftarrow \text{not } b.\} \\ \{a \vee b.,\ a \leftarrow b.,\ \neg c \leftarrow \text{not } b. & \{c \leftarrow \text{not } b.\} \\ \{a \vee b.,\ c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\} & \{a \leftarrow b.\} \end{array}$$

Indeed, the unique minimal hitting set of $coC_{max}(P)$ is the only set contained in

$$SI_{min}(P) = \{\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}\}.$$

Example 3.1.22. Our argumentation framework with $R = \{(a, b), (b, c), (c, c)\}$ from above possesses

$$C_{max}(R) = \{\{(b, c), (c, c)\},\ \{(a, b), (b, c)\}\}, \quad coC_{max}(R) = \{\{(a, b)\},\ \{(c, c)\}\}.$$

We see that the unique hitting set of $coC_{max}(R)$ is $\{(a, b),\ (c, c)\}$, which is the only set in $SI_{min}(R)$.

The procedure of finding the minimal conflicts using this dual version of our hitting set duality is as before, depicted in Figure 3.5:

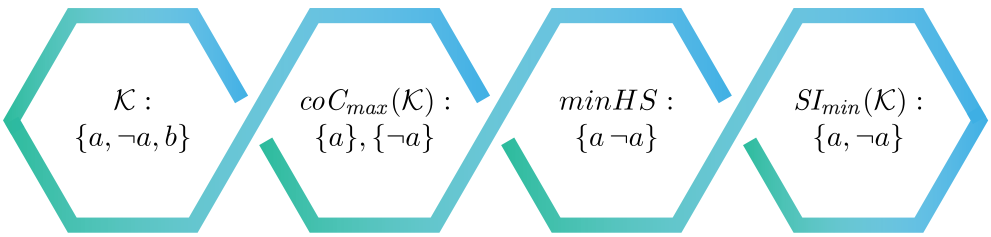


Figure 3.5: The "dual" version of the hitting set duality: Given an inconsistent knowledge base, we first compute $coC_{max}(\mathcal{K})$ and proceed with computing its minimal hitting sets. We obtain the minimal inconsistent subsets.

Let us summarize the results we obtained so far. Motivated by the observation that Reiter's hitting set duality does not generalize to non-monotonic logics, we developed strong inconsistency. This notion is based on the idea to turn the following property of inconsistent subsets of monotonic knowledge bases into an axiom:

- If $\mathcal{H} \subseteq \mathcal{K}$ is inconsistent, then so is each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$.

We showed natural properties of strong inconsistency in Proposition 3.1.11 and in view of our hitting set characterizations (Theorem 3.1.12 and the corollary Theorem 3.1.19) we are convinced that strong inconsistency is an appropriate generalization of mere inconsistency. We will extend this investigation to cases where one might *add* information to repair knowledge bases in Section 3.2. Before we do so, we want to continue our investigation of strong inconsistency by providing links to the related notion of *strong equivalence*. We will also discuss a hitting set duality based on refining consistency rather than inconsistency, which shall illustrate that a symmetric approach yields symmetric results. This further emphasizes the well-behaved structure of knowledge bases, even for non-monotonic logics.

3.1.3 Strong Inconsistency and Strong Equivalence

In propositional logic, equivalence is an important notion as it guarantees substitutability: Whenever two formulas ϕ and ϕ' are equivalent, that is, possess the same models, and ϕ is a subformula of ψ, then replacing ϕ by ϕ' in ψ yields a formula equivalent to ψ. In non-monotonic formalisms this is no longer the case, which is illustrated in Example 3.1.23 below. This observation has led to a body of literature on so-called *strong equivalence* (not only defined for ASP, but also for example for AFs), a more adequate notion of equivalence for non-monotonic reasoning (see for instance [54; 81; 89]). Let us consider the following example to motivate the notion of strong equivalence.

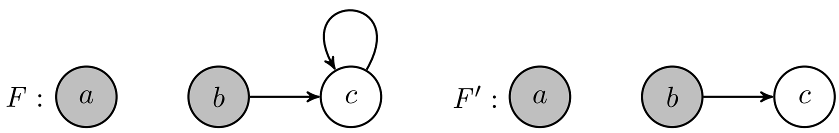
Example 3.1.23. Let P and P' be the following logic programs.

$$P: \; b \leftarrow \text{not } a. \qquad\qquad P': \; b.$$
$$c \leftarrow \text{not } a. \qquad\qquad c.$$

Even though P and P' are equivalent in the sense that they possess the same answer set, it is quite obvious that they encode different information. Moreover, they might behave differently when being considered as parts of a larger program. For example if G is the program consisting of only the rule "a.", then $P \cup G$ and $P' \cup G$ have different answer sets.

The previous quite simple example already shows that equivalence does not guarantee substitutability in ASP as it does in propositional logic. We expect expect this behavior in other non-monotonic formalisms as well. Indeed we can make similar observations for abstract argumentation frameworks.

Example 3.1.24. Assume we have $A = \{a, b, c\}$ as usual. Then, both $F = (A, R)$ and $F' = (A, R')$ with $R = \{(b, c), (c, c)\}$ and $R' = \{(b, c)\}$ have $\{a, b\}$ as stable extension. However, the AF with attacks $R \cup \{(a, b)\}$ has none while the AF with attacks $R' \cup \{(a, b)\}$ has $\{a, c\}$. The AFs F and F' are depicted below:



As already mentioned above, this observation has led to the development of so-called strong equivalence. Speaking in terms of answer set programming, P and P' are strongly equivalent iff $P \cup G$ and $P' \cup G$ have the same answer sets for every logic program G. For argumentation frameworks, strong equivalence is defined analogously.

We now consider a connection between strong equivalence and strong inconsistency. Strong equivalence can naturally be generalized to arbitrary logics in the following way:

Definition 3.1.25. Let $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ be a logic. The knowledge bases $\mathcal{K}$ and $\mathcal{K}'$ are *strongly equivalent* if $\mathcal{ACC}(\mathcal{K} \cup \mathcal{G}) = \mathcal{ACC}(\mathcal{K}' \cup \mathcal{G})$ for each $\mathcal{G} \subseteq \mathcal{WF}$.

The following proposition shows compatibility of strong inconsistency with strong equivalence.

Proposition 3.1.26. *Let $\mathcal{K}$, $\mathcal{K}'$ and $\mathcal{G}$ be knowledge bases. If $\mathcal{K}$ and $\mathcal{K}'$ are strongly equivalent, then $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent iff $\mathcal{K}'$ is strongly $(\mathcal{K}' \cup \mathcal{G})$-inconsistent.*

Proof. The knowledge base $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent if and only if $\mathcal{K} \cup \mathcal{A}$ is inconsistent for any $\mathcal{A} \subseteq \mathcal{G}$. Since $\mathcal{K}$ and $\mathcal{K}'$ are strongly equivalent, this is the case if and only if $\mathcal{K}' \cup \mathcal{A}$ is inconsistent for any $\mathcal{A} \subseteq \mathcal{G}$, which is the definition of $\mathcal{K}'$ being strongly $(\mathcal{K}' \cup \mathcal{G})$-inconsistent. □

Example 3.1.27. The programs

$$
\begin{aligned}
P:\ & a. & \qquad\qquad P':\ & a.\\
& b. & & b \leftarrow a.\\
& \neg b \leftarrow \text{not } c. & & \neg b \leftarrow \text{not } c.
\end{aligned}
$$

are strongly equivalent. If we let $G = \{\neg a.\}$, we see that P is strongly $(P \cup G)$-inconsistent as well as P' is strongly $(P' \cup G)$-inconsistent. On the other hand, augmenting the programs with $G' = \{c.\}$ renders none of them strongly inconsistent.

Example 3.1.28. Now let us consider a case where equivalence is used instead of strong equivalence. Take

$$
\begin{aligned}
P:\ & a. & \qquad\qquad P':\ & a.\\
& b. & & b \leftarrow a.\\
& \neg b. & & \neg b \leftarrow \text{not } c.
\end{aligned}
$$

Observe that P and P' are not strongly equivalent. Now if $G = \{c.\}$, then P is strongly $(P \cup G)$-inconsistent, while P' is not strongly $(P' \cup G)$-inconsistent.

Let us also consider an example for abstract argumentation. Identifying strong equivalence for two AFs is far from trivial. For the ones we consider during the following example, we obtain strong equivalence due to [89] since both AFs possess the same s-kernel, defined as

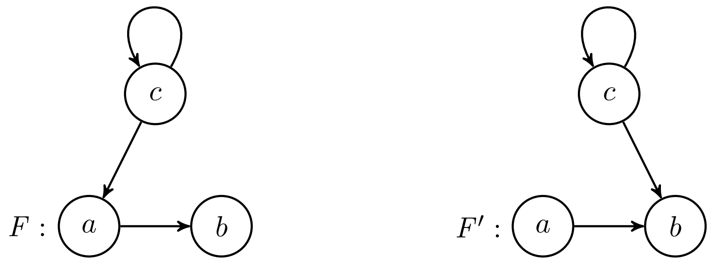
$$F^{sk} = (A, R^{sk}) \qquad \text{with} \qquad R^{sk} = \{R \setminus (a, b) \mid a \neq b, (a, a) \in R\}$$

for a given AF $F = (A, R)$.

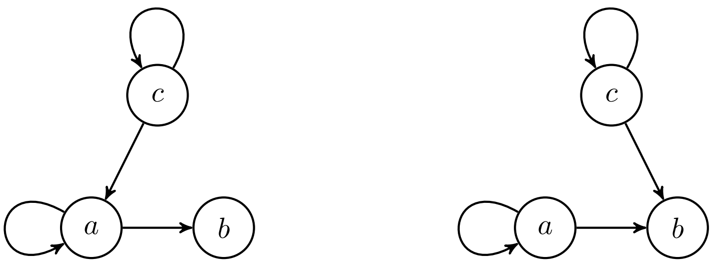
Example 3.1.29. Now let $F = (A, R)$ and $F' = (A, R')$ with

$$R = \{(a, b), (c, a), (c, c)\} \qquad \text{and} \qquad R' = \{(a, b), (c, b), (c, c)\}$$

and $A = \{a, b, c\}$ as before:



As already mentioned, F and F' are strongly equivalent. Consider for instance the attack (a, a). We see that the knowledge base R is strongly $(R \cup \{(a, a)\})$-inconsistent as well as R' is strongly $(R' \cup \{(a, a)\})$-inconsistent:



However, introducing an attack (a, c) resolves inconsistency in both cases, i. e., $R \cup \{(a, c)\}$ as well as $R' \cup \{(a, c)\}$ are consistent with stable extension $\{a\}$:

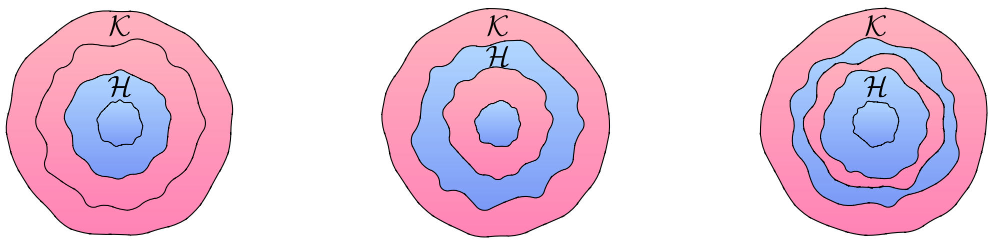
As a final remark to this section, we want to mention that the connection between strong equivalence and monotony is in general not as simple as the examples we chose suggest: Judging from our cases (i. e., propositional logic, ASP and AFs) it might appear that equivalence and strong equivalence coincide whenever the underlying logic is monotonic, so the latter notion is only interesting for non-monotonic logics. However, this is not true in general. We refer the reader to [22] for a discussion on this issue.

3.1.4 Strong Consistency

The analysis so far was based on strong inconsistency. Instead, one could also think of a refined notion of consistency. In fact, the structure of knowledge bases is robust enough to obtain similar results by utilizing a symmetrical notion. Recall that strong inconsistency was chosen to impose a monotonic behavior on $\mathcal{K}$ with respect to inconsistency. The same can be achieved by refining the notion of consistency in a way that it does not allow inconsistent subsets anymore. Even though the rest of this work will focus on strong inconsistency, we will discuss this aspect briefly here. We are convinced that it is interesting on its own, contributing to understanding the structural properties of non-monotonic knowledge bases.

Definition 3.1.30. For $\mathcal{H}, \mathcal{K} \subseteq \mathcal{WF}$ with $\mathcal{H} \subseteq \mathcal{K}$ we call $\mathcal{H}$ *strongly consistent* if $\mathcal{H}' \subseteq \mathcal{H}$ implies $\mathcal{H}'$ is consistent. Let $SC(\mathcal{K})$ and $SC_{max}(\mathcal{K})$ denote the strongly consistent and maximal strongly consistent subsets of $\mathcal{K}$, respectively.

A comparison between mere and strong consistency is depicted in Figure 3.6. As before, circles correspond to subsets of $\mathcal{K}$, consistent ones filled blue, inconsistent ones filled red. For a monotonic knowledge base, (maximal) consistency of $\mathcal{H}$ implies consistency of each subset of $\mathcal{H}$ (1), which is not true for a non-monotonic logic (2). Picture (3) illustrates a maximal strongly consistent subset $\mathcal{H}$. As in the monotonic case, there is no red circle inside the blue one for $\mathcal{H}$.



$\mathcal{H}$ maximal consistent (monotonic)	$\mathcal{H}$ maximal consistent (non-monotonic)	$\mathcal{H}$ maximal strongly consistent (non-monotonic)
(1)	(2)	(3)

Figure 3.6: Maximal consistency vs. maximal strong consistency

Example 3.1.31. Let $\mathcal{K} = \{a,\ a \rightarrow b,\ \neg b,\ c,\ \neg c\}$. The set $\mathcal{H} = \{a,\ a \rightarrow b,\ c\}$ is consistent. Of course, each subset of $\mathcal{H}$ is consistent as well. Hence, $\mathcal{H}$ is strongly consistent.

Example 3.1.32. Now let P be the logic program

$$\begin{array}{lrr} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

The subset

$$H: \quad a \vee b. \quad c \leftarrow \text{not } b. \quad \neg c \leftarrow \text{not } b.$$

is consistent, but not strongly consistent since $\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}$ is an inconsistent subset. We see however that $\{a \vee b.,\ c \leftarrow \text{not } b.\}$ is strongly consistent.

Let us also consider our running example AF:

Example 3.1.33. For the knowledge base $R = \{(a,b),(b,c),(c,c)\}$, we have the consistent subset $\{(b,c),(c,c)\}$, but it is not strongly consistent. However, it is easy to see that $\{(a,b),(b,c)\}$ is strongly consistent.

As pointed out in Proposition 3.1.11, ordinary and strong inconsistency coincide for monotonic logics. Unsurprisingly, this is also true for strong consistency. The following result is an immediate consequence of Lemma 2.5.18.

Proposition 3.1.34. *Let $\mathcal{K}$ be a knowledge base.*

1. *If $\mathcal{K}$ is monotonic, then $C(\mathcal{K}) = SC(\mathcal{K})$.*
2. *If $\mathcal{K}$ is monotonic, then $C_{max}(\mathcal{K}) = SC_{max}(\mathcal{K})$.*

Before stating the analogous duality results, we need the following notion:

Definition 3.1.35. A set $\overline{\mathcal{H}} \subseteq \mathcal{K}$ is in *coSC$_{max}$*$(\mathcal{K})$ if there is a set $\mathcal{H} \in SC_{max}(\mathcal{K})$ such that $\overline{\mathcal{H}} = \mathcal{K} \setminus \mathcal{H}$.

Now we can phrase Theorems 3.1.12 and 3.1.19 in terms of strong consistency. We want to emphasize that ordinary inconsistency is used rather than strong inconsistency. This shows that it is sufficient to move either from mere to strong *inconsistency* or from mere to strong *consistency*; it is not necessary to refine both notions.

Theorem 3.1.36. *Let $\mathcal{K}$ be a knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $I_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in SC_{max}(\mathcal{K})$. Moreover, $\mathcal{S}$ is a minimal hitting set of coSC$_{max}(\mathcal{K})$ if and only if $\mathcal{S} \in I_{min}(\mathcal{K})$.*

Proof. It suffices to prove the first statement since the latter can then be inferred from Lemma 3.1.18.

"$\Rightarrow$": Let $\mathcal{S}$ be a minimal hitting set of $I_{min}(\mathcal{K})$. We first show that $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$ is strongly consistent. Assume $\mathcal{H}' \subseteq \mathcal{H}$ is inconsistent. Let $\mathcal{H}'$ w. l. o. g. be minimal, i. e., $\mathcal{H}' \in I_{min}(\mathcal{K})$. However, we have $\mathcal{S} \cap \mathcal{H} = \emptyset$ by definition of $\mathcal{H}$ and hence, $\mathcal{S} \cap \mathcal{H}' = \emptyset$ which is a contradiction since $\mathcal{S}$ was assumed to be a hitting set of $I_{min}(\mathcal{K})$.

Now assume $\mathcal{H}$ is not maximal, i. e., $\mathcal{H}'$ with $\mathcal{H} \subsetneq \mathcal{H}'$ is strongly consistent. Then, $\mathcal{S}' = \mathcal{K} \setminus \mathcal{H}'$ is a hitting set of $I_{min}(\mathcal{K})$ (otherwise, $\mathcal{H}'$ would contain an inconsistent subset). However, we now have $\mathcal{S}' = \mathcal{K} \setminus \mathcal{H}' \subsetneq \mathcal{K} \setminus \mathcal{H} = \mathcal{S}$ contradicting minimality of $\mathcal{S}$.

"$\Leftarrow$": If $\mathcal{K} \setminus \mathcal{S} \in SC(\mathcal{K})$, then $\mathcal{S}$ must clearly be a hitting set of $I_{min}(\mathcal{K})$. Now assume $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$ is maximal consistent. Hence, any proper superset of $\mathcal{H}$ contains an inconsistent set and thus, a proper subset of $\mathcal{S}$ cannot be a hitting set. We obtain minimality of $\mathcal{S}$. □

In order to familiarize with this duality results, let us consider our running examples – both monotonic and non-monotonic

Example 3.1.37. For the propositional knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$, Theorem 3.1.36 corresponds to the classical duality results.

Example 3.1.38. For our program

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

we have

$$I_{min}(P) = \{\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}\}$$

with the two obvious hitting sets. Indeed,

$$\begin{aligned} SC_{max}(P) = \{&\{a \vee b.,\ a \leftarrow b.,\ c \leftarrow \text{not } b.\},\\ &\{a \vee b.,\ a \leftarrow b.,\ \neg c \leftarrow \text{not } b.\}\}. \end{aligned}$$

Example 3.1.39. For our AF represented by $R = \{(a, b), (b, c), (c, c)\}$ we have

$$I_{min}(R) = \{\{(c, c)\}\}, \qquad SC_{max}(R) = \{\{(a, b), (b, c)\}\}.$$

The duality is easy to verify.

Note that here, we use strong consistency and the classical notion of inconsistency. It is interesting that these complementary results can be obtained by considering strong consistency instead of strong inconsistency. However, we believe that in non-monotonic reasoning, strong inconsistency and ordinary consistency as used for the previous results are more appropriate. For example, inconsistent subsets of a knowledge base should not be an issue as long as the knowledge base itself is consistent. That is why our investigation focuses on strong inconsistency rather then strong consistency.

3.2 Augmenting Knowledge Bases: Additional Information

Let us recall the motivation for our notion of strong inconsistency. In monotonic logics, we have: If $\mathcal{H} \subseteq \mathcal{K}$ is inconsistent, then the same is true for each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$. In a non-monotonic logic this is not necessarily the case which led to Definition 3.1.7. However, instead of insisting on moving to maximal consistent subsets of $\mathcal{K}$, this observation suggests a novel approach, namely resolving conflicts via *adding* information. Especially in frameworks like ASP where the absence of answer sets is oftentimes due to a minimality requirement, this appears to be a quite promising method to restore consistency. To illustrate this, consider the following example:

Example 3.2.1. Assume a self-driving car is aware of the fact that it should never injure anyone, so the constraint

$$\leftarrow injure(X).$$

is implemented. Moreover, driving while people are on the street might cause this undesired outcome, so we have

$$injure(X) \leftarrow driving,\ on\_street(X).$$

Now assume the following situation occurs: Our car has green light ahead which means it is allowed to cross the intersection by default:

$$light(green). \qquad driving \leftarrow light(green),\ \text{not } abn.$$

However, since a young gentleman is still on the street, the car also learns:

$$on\_street(A).$$

Of course, the above rules form an inconsistent logic program P_{car}. Our car is thus not able to act (in which case it should probably brake immediately, but this is beyond our example). It does not appear to be reasonable to drop one of the above rules. In fact, it is quite obvious that our car should treat the presence of people on the street as an "abn" instance, which then prevents it from applying "$driving \leftarrow light(green),\ \text{not } abn.$". So instead of removing anything, we augment P_{car} with the rule

$$r:\ abn \leftarrow on\_street(X).$$

This yields the consistent program

$$\begin{array}{lrr} P_{car} \cup \{r\}: & \leftarrow injure(X). & injure(X) \leftarrow driving,\ on\_street(X). \\ & light(green). & driving \leftarrow light(green),\ \text{not } abn. \\ & on\_street(A). & abn \leftarrow on\_street(X). \end{array}$$

with answer set $\{light(green),\ on\_street(A),\ abn\}$.

In the previous example addition of a simple fact was sufficient to restore consistency of the given program. Of course, this will not always be the case and some conflicts might not be resolvable via additional information at all, e.g., two complementary facts "$a.$" and "$\neg a.$". We should thus not only focus on adding information, but also combining both approaches. Hence, the following questions arise:

- How should one formally define repairs of a given knowledge based on adding information? How can we combine adding and deleting?
- Is there a hitting set characterization for the above defined repair notions?
- If so, how does the corresponding inconsistency notion look like? Is it natural? Does it compare to strong inconsistency in a certain sense?
- Does it generalize our former notions and results?

During this section, we will answer the above questions. The results we are going to obtain are quite pleasant. In particular, the hitting set characterization can be achieved via natural inconsistency notions, similar in spirit to strong inconsistency. We will also find interesting relationships to the notions and results we have considered so far.

3.2.1 A Hitting Set Duality for Addition-Based Repairs

Let us start with repairs based on additional information. In general, it is not quite clear which additional information might be appropriate, especially when considering an arbitrary abstract logic as in Definition 2.5.1. Moreover, phrasing meaningful results appears hard when investigating an *arbitrary* superset of a knowledge base $\mathcal{K}$. We thus assume the set of potential additional information is given.

More precisely, we consider knowledge bases $\mathcal{K}$ (as usual) and $\mathcal{G}$ (of potential additional assumptions). The set $\mathcal{G}$ itself is not necessarily consistent. For technical convenience we assume $\mathcal{K}$ and $\mathcal{G}$ to be disjoint. This assumption also matches the intuitive meaning of $\mathcal{G}$ as a set of potential additional information. The following definition formally introduces repairs that utilize $\mathcal{G}$.

Definition 3.2.2. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. If for $\mathcal{A} \subseteq \mathcal{G}$, $\mathcal{K} \cup \mathcal{A}$ is consistent, then we call $\mathcal{A}$ a *repairing subset of* $\mathcal{G}$ wrt. $\mathcal{K}$. Let $\text{REP}(\mathcal{K}, \mathcal{G})$ and $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$ denote the repairing subset of $\mathcal{G}$ wrt. $\mathcal{K}$ and the minimal ones, respectively.

Of course, no additional information is capable of repairing an inconsistent knowledge base of a monotonic logic. So our running example of a propositional knowledge base is meaningless in this context, and we move straight to our non-monotonic formalisms.

Example 3.2.3. Recall our running example P.

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

Assume we have

$$G: \qquad a. \qquad b.$$
$$d. \qquad b \leftarrow d.$$

We see that $P \cup \{b.\}$ and $P \cup \{d.,\ b \leftarrow d.\}$ are already consistent and thus

$$\text{REP}_{min}(P, G) = \{\{b.\}, \{d.,\ b \leftarrow d.\}\}.$$

Minimality is clear.

Let us now extend our running example for abstract argumentation. Recall Example 2.5.5 where we defined the logic

$$L^A_{\text{AF}} = (\mathcal{WF}_{\text{AF}}, \mathcal{BS}_{\text{AF}}, \mathcal{INC}_{\text{AF}}, \mathcal{ACC}_{\text{AF}}).$$

We pointed out that an AF is actually a tuple and not a set. We thus represent AFs as a knowledge base in a way that a set A of arguments is fixed and $\mathcal{K}$ contains the attacks. Hence augmenting F with another knowledge base G means in our setup additional *attacks* rather than novel arguments. Of course, it would be possible to interpret our running example framework F as an AF over, e. g., $A' = \{a, b, c, d\}$ resulting in an AF containing the argument "d" which does not participate in any attack. We will stick however with an AF over $A = \{a, b, c\}$.

Example 3.2.4. So consider the AF represented by $R = \{(a, b), (b, c), (c, c)\}$. Assume we are given additional attacks $G = \{(a, c), (b, a), (c, b)\}$.

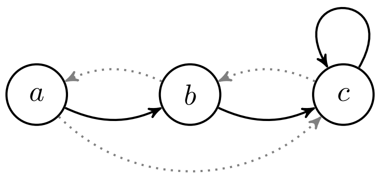


Figure 3.7: The argumentation framework represented by $R \cup G$ (attacks in G dotted and gray) from Example 3.2.4

Observe that $R \cup \{(a, c)\}$ and $R \cup \{(b, a)\}$ represent AFs that possess stable extensions.

Figure 3.8: The argumentation frameworks represented by $R \cup \{(a, c)\}$ and $R \cup \{(b, a)\}$ from Example 3.2.4 including their respective stable extensions (gray)

We thus see

$$\text{REP}_{min}(R, G) = \{\{(a, c)\}, \{(b, a)\}\}$$

Again, the repairs are clearly minimal.

Our goal is to characterize the minimal repairs for a given knowledge base $\mathcal{K}$ in terms of a hitting set duality similar in spirit to Theorem 3.1.12. In the latter theorem the result is the notion of strong inconsistency i. e., subsets $\mathcal{H}$ of a knowledge base $\mathcal{K}$ such that each set $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. For addition-based repairs, our notion is a natural counterpart thereto which we want to develop by considering Example 3.2.3 again.

Assume for the moment the goal was already achieved, implying we had certain sets whose hitting sets are

$$\text{REP}_{min}(P, G) = \{\{b.\}, \{d., \, b \leftarrow d.\}\}.$$

Even without being aware of a general technique, this is easy to obtain for this particular example: We consider $G_1 = \{b., \, d.\}$ and $G_2 = \{b., \, b \leftarrow d.\}$. The reader may observe that the two hitting sets of G_1 and G_2 are $\{b.\}$ and $\{d., \, b \leftarrow d.\}$. By removing the G_i from G, we find their meaning: We obtain

$P:$	$a \vee b.$	$a \leftarrow b.$
	$c \leftarrow \text{not } b.$	$\neg c \leftarrow \text{not } b.$
$G \setminus G_1:$	$a.$	$b \leftarrow d.$
$G \setminus G_2:$	$a.$	$d.$

and we see that P is now strongly inconsistent in both cases. More precisely, P is strongly $(P \cup G \setminus G_1)$-inconsistent as well as strongly $(P \cup G \setminus G_2)$-inconsistent. The intuitive reason is that in both cases we removed all possibilities to repair P via G. Hence, in general we are looking for sets $\mathcal{H} \subseteq \mathcal{G}$ such that $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{H})$-inconsistent. If we set $\mathcal{A} = \mathcal{G} \setminus \mathcal{H}$, this means $\mathcal{K}$ has to be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent. When comparing to strong inconsistency, this is symmetric, taking supersets of $\mathcal{K}$ into account rather than subsets.

Definition 3.2.5. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. If for $\mathcal{A} \subseteq \mathcal{G}$, $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, i. e., $\mathcal{K} \in SI(\mathcal{K} \cup \mathcal{A})$, then we call $\mathcal{A}$ a *non-repairing subset* of $\mathcal{G}$ wrt. $\mathcal{K}$. Let $\text{NREP}(\mathcal{K}, \mathcal{G})$ and $\text{NREP}_{max}(\mathcal{K}, \mathcal{G})$ denote the set of non-repairing subsets of $\mathcal{G}$ wrt. $\mathcal{K}$ and the maximal ones, respectively.

A maximal non-repairing subset $\mathcal{A}$ of $\mathcal{G}$ wrt. $\mathcal{K}$ is depicted in Figure 3.9 (2). As we can see, this notion is similar to strong inconsistency in the sense that all sets within a certain range need to be inconsistent. Picture (3) illustrates the trivial case where the underlying logic is monotonic. Here, considering supersets does of course not yield insightful results.

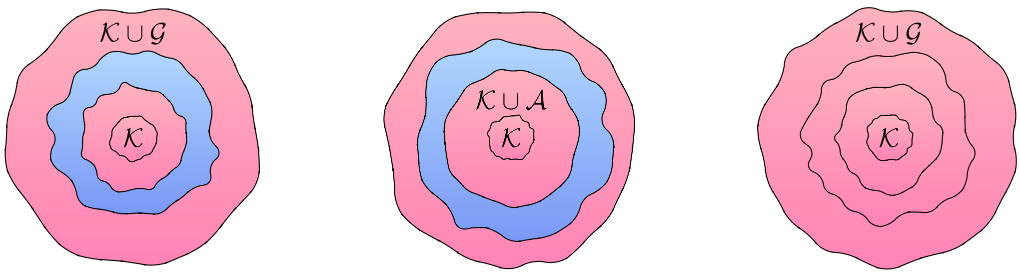


Illustration of $\mathcal{K} \cup \mathcal{G}$ | $\mathcal{A}$ maximal non-repairing subset of $\mathcal{G}$ | $\mathcal{G}$ maximal non-repairing subset of $\mathcal{G}$ (monotonic)

(1) | (2) | (3)

Figure 3.9: Maximal non-repairing subsets of $\mathcal{G}$

To see the non-repairing subsets of $\mathcal{G}$ at work, we consider the following examples:

Example 3.2.6. Recall our running example P from above

$$P: \quad a \vee b. \quad a \leftarrow b. \quad c \leftarrow \text{not } b. \quad \neg c \leftarrow \text{not } b.$$

with

$$G: \quad a. \quad b. \quad d. \quad b \leftarrow d.$$

Indeed, the maximal non-repairing subsets of G are A_1 and A_2 where

$$A_1: \quad a. \quad b \leftarrow d.$$
$$A_2: \quad a. \quad d.$$

Example 3.2.7. Now recall the framework represented by $R = \{(a, b), (b, c), (c, c)\}$ with $G = \{(a, c), (b, a), (c, b)\}$. There is only one non-repairing subset of G, namely $\{(c, b)\}$. To see this recall from Figure 3.8 that adding "(a, c)" or "(b, a)" results in a consistent AF. Hence

$$\textsc{NRep}_{max}(R, G) = \{\{(c, b)\}\}.$$

We are almost ready to phrase a duality result similar in spirit to Theorem 3.1.12. As before, we require one additional auxiliary notion, namely co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ which is defined as expected:

Definition 3.2.8. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. The set co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ consists of all $\overline{\mathcal{A}} \subseteq \mathcal{G}$ such that $\mathcal{G} \setminus \overline{\mathcal{A}}$ is in $\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.

As desired, the following theorem gives a characterization of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ in terms of a hitting set duality.

Theorem 3.2.9 (Superset Duality)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.*

In order to prove this theorem, we utilize Lemma 3.1.18 as we did to infer Theorem 3.1.19 from Theorem 3.1.12. So instead of proving Theorem 3.2.9 directly, we start with the following:

Theorem 3.2.10 (Superset Duality II)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.*

Proof. "$\Rightarrow$": Let $\mathcal{S}$ be a minimal hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $\mathcal{G} \setminus \mathcal{S} \notin \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.

First assume $\mathcal{G} \setminus \mathcal{S} \notin \textsc{NRep}(\mathcal{K}, \mathcal{G})$. Then, there is a set $\mathcal{S}'$ with $\mathcal{S} \subseteq \mathcal{S}'$ such that $(\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}'$ is consistent. Due to finiteness of $\mathcal{G}$, we might assume $\mathcal{S}'$ is maximal among all subsets of $\mathcal{G}$ that render $(\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}'$ consistent. Set $\mathcal{A} = \mathcal{G} \setminus \mathcal{S}'$. Then, $\mathcal{K} \cup \mathcal{A}$ is consistent. In particular, $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. Due to

$$\mathcal{A} \cap \mathcal{S} \subseteq \mathcal{A} \cap \mathcal{S}' = (\mathcal{G} \setminus \mathcal{S}') \cap \mathcal{S}' = \emptyset$$

we infer $\mathcal{A} \cap \mathcal{S} = \emptyset$. Thus, $\mathcal{S}$ is no hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$, which is a contradiction.

Now assume $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}(\mathcal{K}, \mathcal{G})$, but it is not maximal. We thus find a set $\mathcal{S}' \subsetneq \mathcal{S}$ such that $\mathcal{G} \setminus \mathcal{S}' \in \textsc{NRep}(\mathcal{K}, \mathcal{G})$. Again due to finiteness we might assume maximality, i. e., $\mathcal{G} \setminus \mathcal{S}' \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. We claim that $\mathcal{S}'$ is a hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ as well, which contradicts minimality of $\mathcal{S}$. This can be seen as follows: Assume $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{K} \cup \mathcal{A}$ is consistent and $\mathcal{A}$ minimal, i. e., $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. In case $\mathcal{A} \cap \mathcal{S}' = \emptyset$ holds, then $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}'$. In particular, $\mathcal{K} \cup \mathcal{A}$ is consistent with

$$\mathcal{K} \subseteq \mathcal{K} \cup \mathcal{A} \subseteq \mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}',$$

i. e., $\mathcal{K} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}')$, which is again a contradiction.

"$\Leftarrow$": Let $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $\mathcal{S}$ is not a minimal hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.

First assume $\mathcal{S}$ is no hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. Hence, there is an $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} \cap \mathcal{S} = \emptyset$. We infer a contradiction as above. Since $\mathcal{A} \cap \mathcal{S} = \emptyset$ implies $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}$, and thus $\mathcal{K} \cup \mathcal{A} \subseteq \mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}$, we can find a consistent subset of $\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}$ which means $\mathcal{G} \setminus \mathcal{S} \notin \text{NREP}(\mathcal{K}, \mathcal{G})$.

Now assume $\mathcal{S}$ is a hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$, but not minimal. So let $\mathcal{S}' \subsetneq \mathcal{S}$ be another hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$. We claim that this implies $\mathcal{G} \setminus \mathcal{S}' \in \text{NREP}(\mathcal{K}, \mathcal{G})$ contradicting the assumed maximality of $\mathcal{G} \setminus \mathcal{S}$. This can be seen as follows: Assume there is a set $\mathcal{A}$ with $\mathcal{A} \subseteq (\mathcal{G} \setminus \mathcal{S}')$ and $\mathcal{K} \cup \mathcal{A}$ is consistent. Due to finiteness assume minimality of $\mathcal{A}$, i. e., $\mathcal{A} \in \text{REP}_{min}(\mathcal{K}, \mathcal{G})$. Now $\mathcal{A} \subseteq (\mathcal{G} \setminus \mathcal{S}')$ implies $\mathcal{A} \cap \mathcal{S}' = \emptyset$ and in particular, $\mathcal{S}'$ is no hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$, which is a contradiction. □

Now let us turn to the main theorem. As already mentioned, we apply Lemma 3.1.18 as before.

Proof of Theorem 3.2.9. Due to Theorem 3.2.10, $\mathcal{S}$ is a minimal hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$. Hence,

$$minHS(\text{REP}_{min}(\mathcal{K}, \mathcal{G})) = co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$$

and thus

$$minHS(minHS(\text{REP}_{min}(\mathcal{K}, \mathcal{G}))) = minHS(co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})).$$

Now we apply Lemma 3.1.18 to $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$ and obtain:

$$\text{REP}_{min}(\mathcal{K}, \mathcal{G}) = minHS(co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})).$$

This proves Theorem 3.2.9. □

Example 3.2.11. Consider again P and G:

$$\begin{array}{lllllll} P: & a \vee b. & a \leftarrow b. & \quad G: & a. & b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & d. & b \leftarrow d. \end{array}$$

Let us summarize:

$$\begin{aligned} \text{REP}_{min}(P, G) &= \{\{b.\}, \{d., \ b \leftarrow d.\}\}, \\ \text{NREP}_{max}(P, G) &= \{\{a., \ b \leftarrow d.\}, \{a., d.\}\}, \\ co\text{-NREP}_{max}(P, G) &= \{\{b., \ d.\}, \{b., \ b \leftarrow d.\}\}. \end{aligned}$$

Indeed, $\text{REP}_{min}(P, G)$ consists of the minimal hitting sets of $co\text{-NREP}_{max}(P, G)$ (Theorem 3.2.9) and vice versa (Theorem 3.2.10).

Example 3.2.12. Recall $R = \{(a, b), (b, c), (c, c)\}$ with $G = \{(a, c), (b, a), (c, b)\}$. Here, the relevant sets are:

$$\begin{aligned} \text{REP}_{min}(R, G) &= \{\{(a, c)\}, \{(b, a)\}\}, \\ \text{NREP}_{max}(R, G) &= \{\{(c, b)\}\}, \\ co\text{-NREP}_{max}(R, G) &= \{\{(a, c), (b, a)\}\}. \end{aligned}$$

The set $\text{REP}_{min}(R, G)$ consists of the two hitting sets of $co\text{-NREP}_{max}(R, G)$. Moreover, $co\text{-NREP}_{max}(R, G)$ contains the unique hitting set of $\text{REP}_{min}(R, G)$.

3.2.2 A Hitting Set Duality for Arbitrary Repairs

Of course, Theorem 3.2.9 is only meaningful if $\mathcal{K}$ is not strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent, i. e., whenever $\mathcal{G} \notin \text{NREP}_{max}(\mathcal{K}, \mathcal{G})$. For example, this is naturally violated whenever the underlying logic is monotonic, but also when $\mathcal{G}$ is inappropriate when it comes to providing repair options for $\mathcal{K}$. This is an advantage of Theorem 3.1.12: Usually, a knowledge base contains consistent subsets and thus the theorem yields non-trivial results. Clearly, the finest solution would be combining the benefits of both Theorem 3.1.12 and Theorem 3.2.9. As it turns out, this can be achieved in a smooth and natural way.

So assume we are given knowledge bases $\mathcal{K}$ and $\mathcal{G}$ with $\mathcal{K} \cap \mathcal{G} = \emptyset$ as before. Our goal is to find a consistent knowledge base $\mathcal{H}$ which is as close as possible to $\mathcal{K}$. In Theorem 3.1.12 the result was a maximal consistent subset of $\mathcal{K}$, i. e., a knowledge base $\mathcal{H}$ of the form $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ where $\mathcal{D}$ is minimal such that $\mathcal{H}$ is consistent. In Theorem 3.2.9 the result was a minimal consistent superset of $\mathcal{K}$, i. e., a knowledge base $\mathcal{H}$ of the form $\mathcal{H} = \mathcal{K} \cup \mathcal{A}$ where $\mathcal{A}$ is minimal such that $\mathcal{H}$ is consistent. Combining both approaches yields the following notion:

Definition 3.2.13. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. We call $(\mathcal{D}, \mathcal{A})$ a *bidirectional repair* for $\mathcal{K}$ with respect to $\mathcal{G}$ if

- $\mathcal{D} \subseteq \mathcal{K}$ and $\mathcal{A} \subseteq \mathcal{G}$,
- $\mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$ is consistent.

By $\text{BI-REP}(\mathcal{K}, \mathcal{G})$ we denote the set of all bidirectional repairs for $\mathcal{K}$ with respect to $\mathcal{G}$. Let $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ be the set of all minimal ones, i. e., if $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$, then $(\mathcal{D}', \mathcal{A}') \in \text{BI-REP}(\mathcal{K}, \mathcal{G})$ and $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{D}' \subseteq \mathcal{D}$ implies $(\mathcal{D}', \mathcal{A}') = (\mathcal{D}, \mathcal{A})$.

Example 3.2.14. Recall our programs P and G.

$$\begin{array}{lllllll} P: & a \vee b. & a \leftarrow b. & G: & a. & b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & d. & b \leftarrow d. \end{array}$$

We see that $\text{BI-REP}_{min}(P, G)$ consists of the tuples

$$(\{b \leftarrow a.\}, \emptyset),\ (\{c \leftarrow \text{not } b.\}, \emptyset),\ (\{\neg c \leftarrow \text{not } b.\}, \emptyset),\ (\emptyset, \{b.\}),\ (\emptyset, \{d.,\ b \leftarrow d.\}).$$

Here, one of the sets in $(\mathcal{D}, \mathcal{A})$ is always empty for $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(P, G)$. We want to illustrate that this is not necessarily the case in general.

Example 3.2.15. Consider P' given via

$$P': \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.$$

and G as before. Note that "$\leftarrow$ not c." will cause inconsistency no matter which rules from G are added. We thus find

$$\text{BI-REP}_{min}(P', G) = \{(P, \emptyset),\ (\{\leftarrow \text{not } c.\}, \{b.\}),\ (\{\leftarrow \text{not } c.\}, \{d.,\ b \leftarrow d.\})\}.$$

Indeed, for all tuples $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(P', G)$ we have $\leftarrow$ not $c. \in \mathcal{D}$ which formalizes that this constraint needs to be removed.

Let us now continue with our running AF example.

Example 3.2.16. For $R = \{(a,b),(b,c),(c,c)\}$ and $G = \{(a,c),(b,a),(c,b)\}$ we recall from previous examples the following options to turn the represented AF F into one which possesses a stable extension (see Figure 3.10): Remove (c,c) (F_1), remove (a,b) (F_2), or add (a,c) (F_3) or (b,a) (F_4):

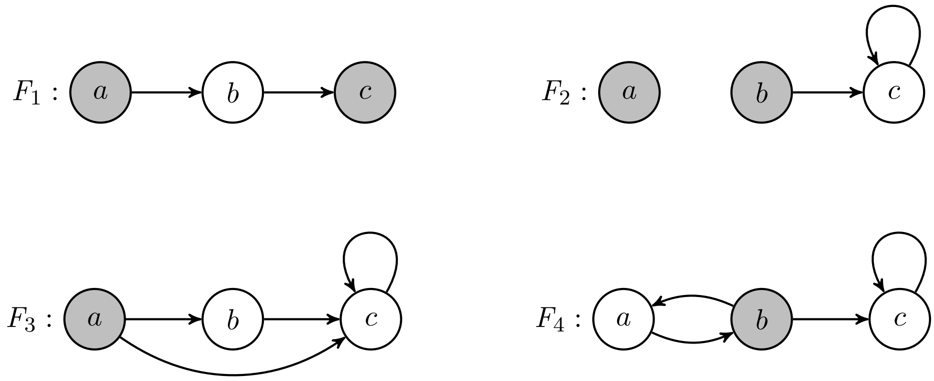


Figure 3.10: The consistent argumentation frameworks $F_1, \ldots, F_4$ from Example 3.2.16

Hence

$$\text{BI-REP}_{min}(R,G) = \{(\{(c,c)\},\emptyset),(\{(a,b)\},\emptyset),(\emptyset,\{(a,c)\}),(\emptyset,\{(b,a)\})\}.$$

We want to emphasize that repair options generalize the notion of consistent subsets of a knowledge base.

Proposition 3.2.17. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\mathcal{D},\emptyset)$ is in $\text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in C_{max}(\mathcal{K})$.*

Proof. "$\Rightarrow$": If $(\mathcal{D},\emptyset) \in \text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$, then $\mathcal{D}$ is minimal such that $\mathcal{K}\setminus\mathcal{D}$ is consistent. So, there is no superset of $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ which is consistent. Hence $\mathcal{H} \in C_{max}(\mathcal{K})$.
"$\Leftarrow$": Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in C_{max}(\mathcal{K})$. Of course, $(\mathcal{D},\emptyset) \in \text{BI-REP}(\mathcal{K},\mathcal{G})$ since $\mathcal{K} \setminus \mathcal{D}$ is consistent. Further, there is no set $\mathcal{D}' \subsetneq \mathcal{D}$ such that $\mathcal{K} \setminus \mathcal{D}'$ is consistent. Hence $(\mathcal{D},\emptyset)$ is necessarily minimal in $\text{BI-REP}(\mathcal{K},\mathcal{G})$, i. e., $(\mathcal{D},\emptyset) \in \text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$. □

The same is true for addition-based repairs, which demonstrates the symmetry of these notions.

Proposition 3.2.18. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\emptyset,\mathcal{A})$ is in $\text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$ if and only if $\mathcal{A} \in \text{REP}_{min}(\mathcal{K},\mathcal{G})$.*

Proof. "$\Rightarrow$": If $(\emptyset,\mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$, then $\mathcal{A}$ is minimal such that $\mathcal{K}\cup\mathcal{A}$ is consistent. Hence $\mathcal{A} \in \text{REP}_{min}(\mathcal{K},\mathcal{G})$.
"$\Leftarrow$": Let $\mathcal{A} \in \text{REP}_{min}(\mathcal{K},\mathcal{G})$. Of course, $(\emptyset,\mathcal{A}) \in \text{BI-REP}(\mathcal{K},\mathcal{G})$ since $\mathcal{K}\cup\mathcal{A}$ is consistent. Further, there is no set $\mathcal{A}' \subsetneq \mathcal{A}$ such that $\mathcal{K} \cup \mathcal{A}'$ is consistent. Hence $(\emptyset,\mathcal{A})$ is necessarily minimal in $\text{BI-REP}(\mathcal{K},\mathcal{G})$, i. e., $(\emptyset,\mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$. □

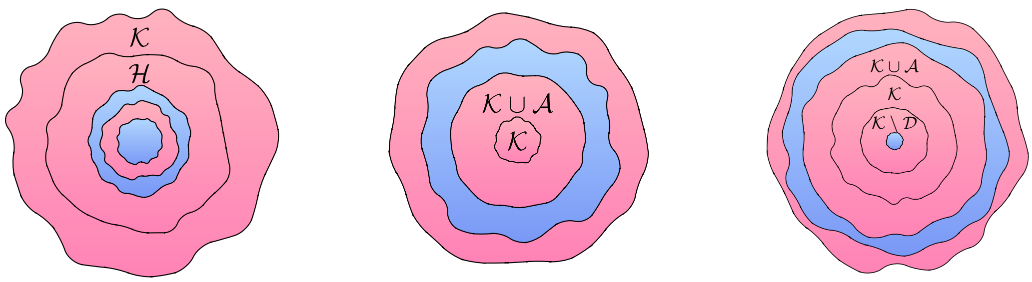
Let us reconsider the notions which led to the hitting set dualities in the previous theorems. In Theorem 3.1.12 the solution is the notion of strong inconsistency. Recall that $\mathcal{H}$ is minimal strongly $\mathcal{K}$-inconsistent if $\mathcal{H}$ is minimal such that $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ implies inconsistency of $\mathcal{H}'$. To put it another way, $\mathcal{D}$ is maximal such that $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ implies inconsistency of $\mathcal{H}'$ Analogously, Theorem 3.2.9 was based on the notion of maximal non-repairing subsets of $\mathcal{G}$. Here, a similar property is required considering sets $\mathcal{K} \cup \mathcal{A}$ rather than $\mathcal{K} \setminus \mathcal{D}$, so roughly speaking, we always face a situation where $\mathcal{K}$ is surrounded by inconsistent sets (recall Figures 3.3 and 3.9). Hence the following comes natural.

Definition 3.2.19. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. We call $(\mathcal{D}, \mathcal{A})$ a *bidirectional non-repair* for $\mathcal{K}$ with respect to $\mathcal{G}$ if

- $\mathcal{D} \subseteq \mathcal{K}$ and $\mathcal{A} \subseteq \mathcal{G}$,
- $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, i. e., $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K} \cup \mathcal{A})$.

Denote by $\text{BI-NREP}(\mathcal{K}, \mathcal{G})$ the set of all bidirectional non-repair for $\mathcal{K}$ with respect to $\mathcal{G}$ and by $\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ the maximal ones.

Consider now Figure 3.11. Intuitively a bidirectional non-repair ensures a "stripe" of inconsistent sets around $\mathcal{K}$ (3). This is similar in spirit to strong inconsistency (1) and non-repairing subsets of $\mathcal{G}$ wrt. $\mathcal{K}$ (2), but taking both directions into account.



$\mathcal{H}$ minimal strongly inconsistent (1) | $\mathcal{A}$ maximal non-repairing subset of $\mathcal{G}$ (2) | $(\mathcal{D}, \mathcal{A})$ maximal bidirectional non-repair (3)

Figure 3.11: Comparison of inconsistency notions

To see the notion of bidirectional non-repairs at work, let us reconsider the three examples from above.

Example 3.2.20. Recall our programs P and G.

$$
\begin{array}{lcccccc}
P: & a \vee b. & a \leftarrow b. & \quad G: & a. & b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & d. & b \leftarrow d.
\end{array}
$$

We already noted in Section 3.2.1 that the non-repairing subsets of G are $A_1 = \{a., \, b \leftarrow d.\}$ and $A_2 = \{a., \, d.\}$. Moreover, removal of "$a \vee b.$" cannot never help turning P into a consistent program. Hence

$$\text{BI-NREP}_{max}(P, G) = \{(\{a \vee b.\}, \{a., b \leftarrow d.\}), \, (\{a \vee b.\}, \{a., d.\})\}.$$

Example 3.2.21. Consider again P'

$$P' : \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.$$

and G as above. Since "$\leftarrow$ not c." will cause inconsistency no matter which rules from G are added we see $P' \setminus \{\leftarrow \text{not } b.\} \in SI(P' \cup G)$, i. e., $(\{\leftarrow \text{not } b.\}, G) \in$ BI-NREP$_{max}(P', G)$. Moreover, even if "$\leftarrow$ not c." is removed, $\{b.\} \subseteq G$ or $\{d., b \leftarrow d.\} \subseteq G$ is also required in order to repair P'. Thus, BI-REP$_{min}(P', G)$ consists of the following tuples:

$$(\{\leftarrow \text{not } b.\}, G),\ (\{\leftarrow \text{not } c.\}, \{a.,\ b \leftarrow d.\}),\ (\{\leftarrow \text{not } c.\}, \{a., d.\}).$$

Example 3.2.22. Recall the AF represented by $R = \{(a, b), (b, c), (c, c)\}$ and the additional attacks $G = \{(a, c), (b, a), (c, b)\}$. Observe that removing "(b, c)" or adding "(c, b)" does not turn the (by R) represented AF F into one possessing a stable extension. Hence,

$$\text{BI-NREP}_{max}(R, G) = \{(\{(b, c)\},\ \{(c, b)\})\}.$$

To verify this, recall that in Example 3.2.16 we saw that any other modification to F yields one possessing a stable extension (also recall Figure 3.10).

We make the following observations to emphasize how BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ generalizes minimal (strong) inconsistency. First let us consider a monotonic logic. Recall that in this case, $SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$. We do not expect $\mathcal{G}$ to play any role here. Indeed, we find:

Proposition 3.2.23. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases of a monotonic logic. If $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{A} = \mathcal{G}$. Moreover, $(\mathcal{D}, \mathcal{G}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

Proof. The first statement is clear. So let us prove the equivalence.

"$\Leftarrow$": Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in I_{min}(\mathcal{K})$. Due to monotony of $\mathcal{K}$, $I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$, so $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent. Moreover, adding formulas from $\mathcal{G}$ cannot render $\mathcal{H}$ consistent. Hence $\mathcal{H}$ is even strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent, i. e., $(\mathcal{G}, \mathcal{D}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

"$\Rightarrow$": Let $(\mathcal{G}, \mathcal{D}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. Due to monotony, this is equivalent to inconsistency of $\mathcal{K} \setminus \mathcal{D}$. Since $\mathcal{D}$ is maximal st. $\mathcal{K} \setminus \mathcal{D}$ is inconsistent, $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ is minimal, i. e., $\mathcal{H} \in I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$. □

In the previous proposition $\mathcal{G}$ was irrelevant as the underlying logic was assumed to be monotonic. Clearly, a similar result holds whenever there is no set $\mathcal{G}$ at all.

Proposition 3.2.24. *Let $\mathcal{K}$ be a knowledge base and $\mathcal{G} = \emptyset$. A tuple of the form $(\mathcal{D}, \emptyset)$ is in* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ *if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

Proof. Let $(\mathcal{D}, \emptyset) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. By definition, $\mathcal{D}$ is maximal such that $\mathcal{K} \setminus \mathcal{D}$ is strongly $\mathcal{K}$-inconsistent. Equivalently, $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ is minimal strongly $\mathcal{K}$-inconsistent. □

A more advanced version of this result without restricting $\mathcal{G}$ is the following. It shows that there is a general connection between BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ and $SI_{min}(\mathcal{K})$, but it is not as straightforward as the connection between BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ and $C_{max}(\mathcal{K})$ (see Proposition 3.2.17). The difference is that we are now looking for maximal instead of minimal tuples with a certain property. Assume we are given a set $\mathcal{H} \in SI_{min}(\mathcal{K})$. We can be sure that $(\mathcal{D}, \emptyset) \in$ BI-NREP$(\mathcal{K}, \mathcal{G})$ with $\mathcal{D} = \mathcal{K} \setminus \mathcal{H}$, but there are in general several sets $\mathcal{A} \subseteq \mathcal{G}$ such that $(\mathcal{D}, \mathcal{A})$ is maximal in BI-NREP$(\mathcal{K}, \mathcal{G})$. On the other hand, given $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ we can guarantee $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$; but not minimality:

Proposition 3.2.25. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$. In particular, there is a set $\mathcal{D}'$ with $\mathcal{D} \subseteq \mathcal{D}'$ such that $\mathcal{K} \setminus \mathcal{D}' \in SI_{min}(\mathcal{K})$.*
- *If $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is a (not necessarily uniquely defined) $\mathcal{A} \subseteq \mathcal{G}$ such that $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.*

Proof.

- Let $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, hence it is also strongly $\mathcal{K}$-inconsistent, so $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$.
- Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$. In particular, $\mathcal{H} \in SI(\mathcal{K})$. Chose a maximal set $\mathcal{A} \subseteq \mathcal{G}$ such that $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{A})$. By assumption, such $\mathcal{A}$ exists since $\mathcal{H} \in SI(\mathcal{K} \cup \emptyset)$ (and $\mathcal{G}$ is finite). By definition, $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. ☐

Thus, the second item is quite nice, but the first one suggests some caution: Given a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, we cannot just "project away" the second component when looking for sets in $SI_{min}(\mathcal{K})$.

Example 3.2.26. Example 3.2.20 shows that BI-NREP$_{max}(P, G)$ contains two distinct tuples (D, A) with $D = \{a \vee b.\}$. We already have $H = P \setminus D \in SI_{min}(P)$, so we can choose $D' = D$ for the first item in Proposition 3.2.25. Since there are two tuples of this form in Example 3.2.20, it illustrates in particular that A in the second item in Proposition 3.2.25 is not uniquely defined in general.

Now let us compare BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ to the non-repairing subsets of $\mathcal{G}$ from Definition 3.2.5. Considering cases where the underlying logic is monotonic or $\mathcal{G}$ is empty will clearly not yield insightful results when investigating NREP$(\mathcal{K}, \mathcal{G})$. However, we find a counterpart to Proposition 3.2.25.

Proposition 3.2.27. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ then $\mathcal{A} \in$ NREP$(\mathcal{K}, \mathcal{G})$. In particular, there is a set $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}'$ such that $\mathcal{A}' \in$ NREP$_{max}(\mathcal{K})$.*
- *If $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$, then there is a (not necessarily uniquely defined) $\mathcal{D} \subseteq \mathcal{K}$ such that $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.*

Proof.

- Let $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, hence $\mathcal{K}$ is also strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, so $\mathcal{A} \in$ NREP$(\mathcal{K}, \mathcal{G})$.
- Let $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$. In particular, $\mathcal{K} \in SI(\mathcal{K} \cup \mathcal{A})$ and $\mathcal{A}$ is maximal with this property. Chose a maximal set $\mathcal{D} \subseteq \mathcal{K}$ such that $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K} \cup \mathcal{A})$. By assumption, such $\mathcal{D}$ exists since $\mathcal{K} \setminus \emptyset \in SI(\mathcal{K} \cup \mathcal{A})$, and $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. ☐

We now return to the main goal of this section, namely our duality characterization for BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. It should not be surprising that a notion of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ is required. The following is natural and well-behaving, extending the previous one component-wise.

Definition 3.2.28. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. The set co-BI-NREP$_{max}(\mathcal{K},\mathcal{G})$ consists of all $(\overline{\mathcal{A}},\overline{\mathcal{D}})$ such that $(\mathcal{G}\setminus\overline{\mathcal{A}},\mathcal{K}\setminus\overline{\mathcal{D}})$ is in BI-NREP$_{max}(\mathcal{K},\mathcal{G})$.

Let us summarize the important sets from the previous examples.

Example 3.2.29. For P and G we found that BI-REP$_{min}(P,G)$ consists of the following tuples:

$$(\{b\leftarrow a.\},\emptyset),\ (\{c\leftarrow \text{not } b.\},\emptyset),\ (\{\neg c\leftarrow \text{not } b.\},\emptyset),\ (\emptyset,\{b.\}),\ (\emptyset,\{d.,\ b\leftarrow d.\}).$$

Moreover,

$$\text{BI-NREP}_{max}(P,G)=\{(\{a\vee b.\},\{a.,b\leftarrow d.\}),\ (\{a\vee b.\},\{a.,d.\})\}.$$

Hence if we set $H=P\setminus\{a\vee b.\}$ we obtain

$$co\text{-BI-NREP}_{max}(P,G)=\{(H,\{b.,\ d.\}),(H,\{b.\,b\leftarrow d.\})\}.$$

Example 3.2.30. Consider again P'

$$P':\qquad\qquad \leftarrow \text{not } b. \qquad\qquad \leftarrow \text{not } c.$$

We had

$$\text{BI-REP}_{min}(P',G)=\{(P,\emptyset),\ (\{\leftarrow \text{not } c.\},\{b.\}),\ (\{\leftarrow \text{not } c.\},\{d.,\ b\leftarrow d.\})\}$$

and BI-NREP$_{max}(P',G)$ consists of

$$(\{\leftarrow \text{not } b.\},G),\ (\{\leftarrow \text{not } c.\},\{a.,\ b\leftarrow d.\}),\ (\{\leftarrow \text{not } c.\},\{a.,d.\}).$$

In particular, co-BI-NREP$_{max}(P',G)$ contains the following tuples:

$$(\{\leftarrow \text{not } c.\},\emptyset),\ (\{\leftarrow \text{not } b.\},\{b.,\ d.\}),\ (\{\leftarrow \text{not } b.\},\{b.,b\leftarrow d.\}).$$

Example 3.2.31. For the AF represented by $R=\{(a,b),(b,c),(c,c)\}$ and additional attacks $G=\{(a,c),(b,a),(c,b)\}$ we obtain the following sets:

$$\begin{aligned}
\text{BI-REP}_{min}(R,\,G)&=\{(\{(c,c)\},\emptyset),(\{(a,b)\},\emptyset),(\emptyset,\{(a,c)\}),(\emptyset,\{(b,a)\})\},\\
\text{BI-NREP}_{max}(R,\,G)&=\{(\{(b,c)\},\ \{(c,b)\})\},\\
co\text{-BI-NREP}_{max}(R,\,G)&=\{(\{(a,b),(c,c)\},\ \{(a,c),(b,a)\})\}.
\end{aligned}$$

The following theorem states that the desired duality result is indeed obtained.

Theorem 3.2.32 (Subset-Superset Duality)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of co-*BI-NREP$_{max}(\mathcal{K},\mathcal{G})$ *iff* $\mathcal{S}\in$ BI-REP$_{min}(\mathcal{K},\mathcal{G})$.

We will proceed as above and give the proof of Theorem 3.2.32 as a corollary of a theorem considering hitting sets of BI-REP$_{min}(\mathcal{K},\mathcal{G})$. Thereto, the following Lemma is convenient as it establishes a technical property we are going to require.

Lemma 3.2.33. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{D} \subseteq \mathcal{K}$. Any set $\mathcal{H}$ with $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$ can be written as $\mathcal{H} = (\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ with $\mathcal{D}' \subseteq \mathcal{D}$ and $\mathcal{A}' \subseteq \mathcal{A}$.*

Proof. Let $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$. First observe that $\mathcal{H}$ can be written as

$$\mathcal{H} = (\mathcal{H} \cap \mathcal{K}) \dot{\cup} (\mathcal{H} \cap \mathcal{A})$$

since $\mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$ with $\mathcal{K} \cap \mathcal{A} \subseteq \mathcal{K} \cap \mathcal{G} = \emptyset$. Clearly, $\mathcal{H} \cap \mathcal{A} \subseteq \mathcal{A}$ so we may set $\mathcal{A}' = \mathcal{H} \cap \mathcal{A}$. Now set $\mathcal{D}' = \mathcal{K} \setminus (\mathcal{H} \cap \mathcal{K})$. We have

$$\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \cap \mathcal{K}$$

and thus

$$\mathcal{D} = \mathcal{K} \setminus (\mathcal{K} \setminus \mathcal{D}) \supseteq \mathcal{K} \setminus (\mathcal{H} \cap \mathcal{K}).$$

Moreover,

$$\mathcal{K} \setminus \mathcal{D}' = \mathcal{K} \setminus (\mathcal{K} \setminus (H \cap \mathcal{K})) = \mathcal{H} \cap \mathcal{K},$$

and hence we obtain

$$(\mathcal{K} \setminus \mathcal{D}') \dot{\cup} (A') = (\mathcal{H} \cap \mathcal{K}) \dot{\cup} (\mathcal{H} \cap \mathcal{A})$$

with $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{D}' \subseteq \mathcal{D}$. □

Equipped with this lemma, we are ready to prove:

Theorem 3.2.34 (Subset-Superset Duality II)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of* BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ *iff $\mathcal{S} \in$ co-*BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

Proof. "⇒": Let $\mathcal{S} = (\mathcal{S}_\mathcal{A}, \mathcal{S}_\mathcal{D})$ be a minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $(\mathcal{K} \setminus \mathcal{S}_\mathcal{A}, \mathcal{G} \setminus \mathcal{S}_\mathcal{D}) \notin$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

First assume $(\mathcal{G} \setminus \mathcal{S}_\mathcal{A}, \mathcal{K} \setminus \mathcal{S}_\mathcal{D}) \notin$ BI-NREP$(\mathcal{K}, \mathcal{G})$. Then, by definition,

$$\mathcal{K} \setminus (\mathcal{K} \setminus \mathcal{S}_\mathcal{D}) \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_\mathcal{A}),$$

and thus,

$$\mathcal{S}_\mathcal{D} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_\mathcal{A}).$$

So there is a consistent set $\mathcal{H}$ with $\mathcal{S}_\mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_\mathcal{A})$. Due to Lemma 3.2.33, we find $\mathcal{D} \subseteq \mathcal{K} \setminus \mathcal{S}_\mathcal{D}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_\mathcal{A}$ with $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$. Due to finiteness of both knowledge bases we might assume $(\mathcal{D}, \mathcal{A}) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. Now $\mathcal{S}_\mathcal{A} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_\mathcal{D} \cap \mathcal{D} = \emptyset$ implies that $\mathcal{S} = (\mathcal{S}_\mathcal{A}, \mathcal{S}_\mathcal{D})$ is no hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$, which is a contradiction.

Now assume $(\mathcal{G} \setminus \mathcal{S}_\mathcal{A}, \mathcal{K} \setminus \mathcal{S}_\mathcal{D}) \in$ NREP$(\mathcal{K}, \mathcal{G})$, but the tuple is not maximal. We thus find a tuple $\mathcal{S}' = (\mathcal{S}_{\mathcal{A}'}, \mathcal{S}_{\mathcal{D}'}) \subseteq (\mathcal{S}_\mathcal{A}, \mathcal{S}_\mathcal{D}) = \mathcal{S}$ such that $(\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}) \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$. We claim that $\mathcal{S}'$ is a hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ as well. This can bee seen as follows: Assume this is not the case, i. e., there is $(\mathcal{D}, \mathcal{A}) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{S}_{\mathcal{A}'} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}'} \cap \mathcal{D} = \emptyset$. By assumption, $\mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$ is consistent. Due to $\mathcal{S}_{\mathcal{A}'} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}'} \cap \mathcal{D} = \emptyset$, we obtain $\mathcal{S}_{\mathcal{D}'} \subseteq \mathcal{K} \setminus \mathcal{D}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}$, so

$$\mathcal{S}_{\mathcal{D}'} \subseteq (\mathcal{K} \setminus \mathcal{D}) \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})).$$

In particular,

$$\mathcal{S}_{\mathcal{D}'} \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})).$$

Due to consistency of $(\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D}$ we infer that $\mathcal{S}_{\mathcal{D}'} \notin SI(\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}))$. So by definition, $(\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}) \notin$ NREP$(\mathcal{K}, \mathcal{G})$ which is a contradiction. Hence, $\mathcal{S}'$ must be a hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ which contradicts minimality of $\mathcal{S}$.

"$\Leftarrow$": Let $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $\mathcal{S} = (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}})$ is no minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$.

First assume that $\mathcal{S}$ is no hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. As above we find a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{S}_{\mathcal{A}} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}} \cap \mathcal{D} = \emptyset$. Similarly we obtain

$$\mathcal{S}_{\mathcal{D}} \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}})).$$

where $(\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D}$ is consistent. Thus, $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}}) \notin$ BI-NREP$(\mathcal{K}, \mathcal{G})$, which is a contradiction.

Now assume that $\mathcal{S}$ is a hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$, but not minimal. Let $\mathcal{S}'$ with $\mathcal{S}' = (\mathcal{S}_{\mathcal{A}'}, \mathcal{S}_{\mathcal{D}'}) \subseteq (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}}) = \mathcal{S}$ be a minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. We claim that $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}'}) \in$ BI-NREP$(\mathcal{K}, \mathcal{G})$ contradicting maximality of $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}})$. For this, assume

$$\mathcal{S}_{\mathcal{D}'} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}).$$

Let $\mathcal{H}$ with $\mathcal{S}_{\mathcal{D}'} \subseteq \mathcal{H} \subseteq (\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}_{\mathcal{A}'}$ be consistent. As above we apply Lemma 3.2.33 to find $\mathcal{D} \subseteq \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}$ with $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$. Again due to finiteness we might assume $(\mathcal{D}, \mathcal{A}) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. Hence, $\mathcal{S}'$ is no hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$, which is again a contradiction. □

Now we are almost ready to prove Theorem 3.2.32. Before we do so, let us make sure that Lemma 3.1.18 is still applicable, even though we consider hitting sets of *tuples* of sets. There is a simple reason why this is no issue: Since we assume $\mathcal{K} \cap \mathcal{G} = \emptyset$, consideration of tuples is simply for ease of presentation. More precisely, if $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{D} \subseteq \mathcal{K}$, then $\mathcal{A}$ and $\mathcal{D}$ are disjoint as well and thus, there is a canonical bijection between the tuples of the form $(\mathcal{D}, \mathcal{A})$ and sets of the form $\mathcal{A} \cup \mathcal{D}$. So if $\mathcal{S} = (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}})$ with $\mathcal{S}_{\mathcal{A}} \subseteq \mathcal{G}$ and $\mathcal{S}_{\mathcal{D}} \subseteq \mathcal{K}$, then $\mathcal{S} \cap (\mathcal{D}, \mathcal{A}) \neq \emptyset$ iff $\mathcal{S}_{\mathcal{A}} \cap \mathcal{A} \neq \emptyset$ or $\mathcal{S}_{\mathcal{D}} \cap \mathcal{D} \neq \emptyset$. Due to $\mathcal{A} \cap \mathcal{D} = \emptyset$ as well as $\mathcal{S}_{\mathcal{A}} \cap \mathcal{S}_{\mathcal{D}} = \emptyset$ this is the case if and only if $(\mathcal{A} \cup \mathcal{D}) \cap (\mathcal{S}_{\mathcal{A}} \cup \mathcal{S}_{\mathcal{A}}) \neq \emptyset$. However, in the latter term no tuple is mentioned. So we may apply Lemma 3.1.18 as before.

Proof of Theorem 3.2.32. By Theorem 3.2.34, $\mathcal{S}$ is a minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Hence we see as above,

$$minHS(minHS(\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}))) = minHS(co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})),$$

which yields

$$\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}) = minHS(co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G}))$$

after applying Lemma 3.1.18 as usual. □

To see the duality at work, we recall our examples.

Example 3.2.35. Consider again P'

$$P': \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.$$

with G as usual. We found

$$\text{BI-REP}_{min}(P',G) = \{(P,\emptyset),\ (\{\leftarrow \text{not } c.\},\{b.\}),\ (\{\leftarrow \text{not } c.\},\{d.,\ b \leftarrow d.\})\}$$

and $co\text{-BI-NREP}_{max}(P',G)$ consists of

$$(\{\leftarrow \text{not } c.\},\emptyset),\ (\{\leftarrow \text{not } b.\},\{b.,\ d.\}),\ (\{\leftarrow \text{not } b.\},\{b., b \leftarrow d.\}).$$

Take $(\{\leftarrow \text{not } c.\},\{b.\}) \in \text{BI-REP}_{min}(P',G)$. Indeed,

$$\begin{aligned}(\{\leftarrow \text{not } c.\},\{b.\}) \cap (\{\leftarrow \text{not } c.\},\emptyset) &= (\{\leftarrow \text{not } c.\},\emptyset)\\ (\{\leftarrow \text{not } c.\},\{b.\}) \cap (\{\leftarrow \text{not } b.\},\{b.,\ d.\}) &= (\emptyset,\{b.\})\\ (\{\leftarrow \text{not } c.\},\{b.\}) \cap (\{\leftarrow \text{not } b.\},\{b., b \leftarrow d.\}) &= (\emptyset,\{b.\})\end{aligned}$$

where all intersections are non-empty. It is thus a hitting set of $co\text{-BI-NREP}_{max}(P',G)$. Minimality can bee seen straightforwardly.

Example 3.2.36. For the AF represented by $R = \{(a,b),(b,c),(c,c)\}$ and additional attacks $G = \{(a,c),(b,a),(c,b)\}$ we found:

$$\begin{aligned}\text{BI-REP}_{min}(R,\ G) &= \{(\{(c,c)\},\emptyset),(\{(a,b)\},\emptyset),(\emptyset,\{(a,c)\}),(\emptyset,\{(b,a)\})\},\\ \text{BI-NREP}_{max}(R,\ G) &= \{(\{(b,c)\},\ \{(c,b)\})\},\\ co\text{-BI-NREP}_{max}(R,\ G) &= \{(\{(a,b),(c,c)\},\ \{(a,c),(b,a)\})\}.\end{aligned}$$

Now consider $(\{(c,c)\},\emptyset) \in \text{BI-REP}_{min}(R,\ G)$. Then

$$(\{(c,c)\},\emptyset) \cap (\{(a,b),(c,c)\},\ \{(a,c),(b,a)\}) = (\{(c,c)\},\emptyset),$$

so it is a hitting set of (the singleton) $co\text{-BI-NREP}_{max}(R,\ G)$. As above, minimality is clear.

In principle, the process of applying our novel duality results works as before, as depicted in Figure 3.12:

Figure 3.12: Summary of Theorems 3.2.32 and 3.2.34: Given an inconsistent knowledge base and potential additional assumptions, we compute $\text{BI-NREP}_{max}(\mathcal{K},\mathcal{G})$ (resp. $\text{BI-REP}_{min}(\mathcal{K},\mathcal{G})$) and consider minimal hitting sets as before.

We have now established the main results of this chapter, so let us briefly summarize. We considered natural notions of repairs of a given knowledge base with the goal to characterize them via a hitting set duality. As it turns out, the corresponding inconsistency notions are symmetric and natural generalizations of $I_{min}(\mathcal{K})$ of a monotonic knowledge base:

- *Minimal strong inconsistency* (see Definition 3.1.7) requires that $\mathcal{H} \subseteq \mathcal{K}$ is minimal such that $\mathcal{H}'$ is inconsistent if $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$. Equivalently, if we set $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$, then $\mathcal{D} \subseteq \mathcal{K}$ is *maximal* st. $\mathcal{H}'$ is inconsistent if $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H}' \subseteq \mathcal{K}$.

- A *maximal non-repairing subset* of $\mathcal{G}$ wrt. $\mathcal{K}$ (see Definition 3.2.5) is characterized by an analogous behavior wrt. supersets of $\mathcal{K}$.

- A *maixmal bidirectional non-repair* (see Definition 3.2.19) ensures that for $(\mathcal{D}, \mathcal{A})$ with $\mathcal{D} \subseteq \mathcal{K}$ and $\mathcal{A} \subseteq \mathcal{G}$, we have inconsistency of $\mathcal{H}$ if $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$.

Moreover in Propositions 3.2.25 and 3.2.27 we pointed out that bidirectional non-repairs generalize the two former notions. Motivated by this observation, we will illustrate in the following section how to see that our duality characterization for bidirectional repairs generalizes the previous duality results.

3.2.3 Properties of Hitting Sets And Former Dualities

As already mentioned, we want to demonstrate how to infer Theorem 3.1.12 as well as Theorem 3.2.9 from the more general Theorem 3.2.32. To see this, we need to investigate the structure of (the hitting sets of) co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Let us start with Theorem 3.1.12. Here, the key observation is that –when trying to restore consistency of $\mathcal{K}$– one is not reliant on $\mathcal{G}$ as long as $\mathcal{K}$ possesses consistent subsets. We need to formally find what this means regarding the hitting sets of $SI_{min}(\mathcal{K})$ resp. co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. We may then translate the duality characterization from Theorem 3.2.32 into the special case Theorem 3.1.12. The first and most important step is the following observation.

Proposition 3.2.37. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets. A set $\mathcal{S}_D$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a minimal hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.*

Proof. "$\Rightarrow$": Let $\mathcal{S}$ be a minimal hitting set of $SI_{min}(\mathcal{K})$. Assume the tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ is in co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} = \mathcal{G} \setminus \overline{\mathcal{A}}$ and $\mathcal{D} = \mathcal{K} \setminus \overline{\mathcal{D}}$. Due to Proposition 3.2.25, $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$, i. e., $\overline{\mathcal{D}} \in SI(\mathcal{K})$. Due to finiteness, there is a set $\overline{\mathcal{D}}' \in SI_{min}(\mathcal{K})$ with $\overline{\mathcal{D}}' \subseteq \overline{\mathcal{D}}$. Since $\mathcal{S}_\mathcal{D}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$, we conclude $\emptyset \neq \mathcal{S}_\mathcal{D} \cap \overline{\mathcal{D}}' \subseteq \mathcal{S}_\mathcal{D} \cap \overline{\mathcal{D}}$. Since $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ was an arbitrary tuple in co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ we see that $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

We have left to prove minimality of $(\mathcal{S}_\mathcal{D}, \emptyset)$. Again due to Proposition 3.2.25, for any $\mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, there is a tuple of the form $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ and hence a tuple of the form $(\mathcal{K} \setminus \mathcal{D}, \mathcal{G} \setminus \mathcal{A}) \in co$-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. So if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_\mathcal{D}$ is a hitting set of $SI_{min}(\mathcal{K})$. Since $\mathcal{S}_\mathcal{D}$ is minimal (as a hitting set of $SI_{min}(\mathcal{K})$), we conclude that $(\mathcal{S}_\mathcal{D}, \emptyset)$ is minimal (as a hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$).

"$\Leftarrow$": If $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ due to Proposition 3.2.25. In particular, there is a tuple $(\mathcal{K} \setminus \mathcal{D}, \mathcal{G} \setminus \mathcal{A}) \in co$-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. So if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_\mathcal{D}\mathcal{S}_\mathcal{D} \cap \mathcal{K} \setminus \mathcal{D} \neq \emptyset$. Hence $\mathcal{S}_\mathcal{D}$ is a hitting set of $SI_{min}(\mathcal{K})$.

Minimality is a consequence of "$\Rightarrow$": Assume $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a minimal hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. If there is a hitting set $\mathcal{S}'_\mathcal{D} \subsetneq \mathcal{S}_\mathcal{D}$ of $SI_{min}(\mathcal{K})$, then $(\mathcal{S}'_\mathcal{D}, \emptyset)$ is also a hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, contradicting minimality of $(\mathcal{S}_\mathcal{D}, \emptyset)$. □

We want to emphasize that Proposition 3.2.37 in particular implies the following: If (the conditions of Proposition 3.2.37 are met and) $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$, then $\overline{\mathcal{D}} \neq \emptyset$. Otherwise, $(\mathcal{S}_{\mathcal{D}}, \emptyset)$ could not be a hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$:

Proposition 3.2.38. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets. Then there is no tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$ with $\overline{\mathcal{D}} = \emptyset$.*

Proof. Although this is a corollary of Proposition 3.2.37, we want to give a straightforward proof which directly illustrates why we require $\mathcal{K}$ to possess consistent subsets:

For any tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$ we have $\mathcal{D} \subsetneq \mathcal{K}$. Otherwise, $\mathcal{K} \setminus \mathcal{K} = \emptyset$ would be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent. However, as $\mathcal{K}$ possesses consistent subsets, this is a contradiction. Hence, whenever $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$, we can conclude $\overline{\mathcal{D}} \neq \emptyset$. □

Having established those properties of hitting sets, we are now ready to infer Theorem 3.1.12 from Theorem 3.2.32 as a corollary.

Corollary 3.2.39 (Theorem 3.1.12)**.** *Let $\mathcal{K}$ be a knowledge base. Then $\mathcal{S}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$.*

Proof. In case $\emptyset \in SI(\mathcal{K})$, i. e., any subset of $\mathcal{K}$ is inconsistent, then the claim holds trivially. So assume $\mathcal{K}$ possesses consistent subsets.

Let $\mathcal{S}$ be a minimal hitting set of $SI_{min}(\mathcal{K})$. Consider an arbitrary knowledge base $\mathcal{G}$ with $\mathcal{K} \cap \mathcal{G} = \emptyset$. Due to Proposition 3.2.37, $\mathcal{S}$ is a hitting set of $SI_{min}(\mathcal{K})$ if and only if $(\mathcal{S}, \emptyset)$ is a minimal hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. By Theorem 3.2.32 this is equivalent to $(\mathcal{S}, \emptyset) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$. Due to Proposition 3.2.17, this is the case if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$. □

Let us now see how to analogously derive Theorem 3.2.9. Note that Corollary 3.2.39 is based on Proposition 3.2.37, so towards Theorem 3.2.9 we require a counterpart to it:

Proposition 3.2.40. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses addition-based repairs. A set $\mathcal{S}_{\mathcal{A}}$ is a minimal hitting set of $co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a minimal hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$.*

Proof. “⇒”: Let $\mathcal{S}_{\mathcal{A}}$ be a minimal hitting set of $co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. Assume the tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ is in $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. Then there is a tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} = \mathcal{G} \setminus \overline{\mathcal{A}}$ and $\mathcal{D} = \mathcal{K} \setminus \overline{\mathcal{D}}$. Due to Proposition 3.2.27, $\mathcal{K} \cup \mathcal{A} \in \textsc{NRep}(\mathcal{K}, \mathcal{G})$. So there is a set $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}' \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. Since $\mathcal{S}_{\mathcal{A}}$ is a minimal hitting set of $co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$, we conclude

$$\emptyset \neq \mathcal{S}_{\mathcal{A}} \cap (\mathcal{G} \setminus \mathcal{A}') \subseteq \mathcal{S}_{\mathcal{A}} \cap (\mathcal{G} \setminus \mathcal{A}) = \mathcal{S}_{\mathcal{A}} \cap (\overline{\mathcal{A}}).$$

Since $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ was an arbitrary tuple in $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$ we see that $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$.

As before, we have left to prove minimality of $(\emptyset, \mathcal{S}_{\mathcal{A}})$. Again due to Proposition 3.2.27, for any $\mathcal{A} \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$, there is a tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. So if $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_{\mathcal{A}}$ is a hitting set of $co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. Since $\mathcal{S}_{\mathcal{A}}$ is minimal (as a hitting set of $co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$), we conclude that $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is minimal (as a hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$) as well.

"$\Leftarrow$": If $\mathcal{A} \in \text{NREP}_{max}(\mathcal{K}, \mathcal{G})$, then there is a tuple $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ due to Proposition 3.2.27. So, if $(\emptyset, \mathcal{S}_\mathcal{A})$ is a hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_\mathcal{A}$ is a hitting set of $co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$ as well.

Minimality is a consequence of "$\Rightarrow$" as in the proof of Proposition 3.2.37. Assume $(\emptyset, \mathcal{S}_\mathcal{A})$ is a minimal hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$. If there is a set $\mathcal{S}'_\mathcal{A} \subsetneq \mathcal{S}_\mathcal{A}$ that is a hitting set of $\text{NREP}_{max}(\mathcal{K}, \mathcal{G})$, then $(\emptyset, \mathcal{S}'_\mathcal{A})$ is a hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ as well, contradicting minimality of $(\emptyset, \mathcal{S}_\mathcal{A})$. □

We may infer an analogous result about the tuples in $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$:

Proposition 3.2.41. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\text{REP}_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$, i. e., $\mathcal{G}$ possesses repairing subsets wrt. $\mathcal{K}$. Then there is no $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ with $\overline{\mathcal{A}} = \emptyset$.*

Proof. For any tuple $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ it holds that $\mathcal{A} \subsetneq \mathcal{G}$. Otherwise, $\mathcal{K}$ would be strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. However, this contradicts $\text{REP}_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$. So, whenever $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ holds, we conclude $\overline{\mathcal{A}} \neq \emptyset$. □

Now let us infer Theorem 3.2.9 from Theorem 3.2.32.

Corollary 3.2.42 (Theorem 3.2.9)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of $co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in \text{REP}_{min}(\mathcal{K}, \mathcal{G})$.*

Proof. The case $\text{REP}_{min}(\mathcal{K}, \mathcal{G}) = \emptyset$ is clear. So let $\text{REP}_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$. Let $\mathcal{S}$ be a minimal hitting set of $co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$. Due to Proposition 3.2.40, $\mathcal{S}$ is a hitting set of $co\text{-NREP}_{max}(\mathcal{K})$ if and only if $(\mathcal{S}, \emptyset)$ is a minimal hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$. By Theorem 3.2.32 this is equivalent to $(\mathcal{S}, \emptyset) \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$. Due to Proposition 3.2.18, this is the case if and only if $\mathcal{S} \in \text{REP}_{min}(\mathcal{K})$. □

We thus see that both Theorem 3.1.12 as well as Theorem 3.2.9 can be inferred from Theorem 3.2.32 by establishing appropriate connections between the hitting sets of the different inconsistency notions. This underlines the fact that the concepts we investigated so far amplify and generalize each other in a natural way. We continue our discussion with some further aspects.

3.3 Preferences and Refinements

Until now, our investigation focused on removing formulas from or adding to a given knowledge base. This approach is rather severe as it does not allow moving from a formula $\alpha \in \mathcal{K}$ to a weaker formula α' instead of just deleting it, i. e., moving to $\mathcal{K} \setminus \{\alpha\}$. The former would yield more fine-grained modifications to our knowledge base $\mathcal{K}$. The goal to keep as much information as possible is already taken into account by moving to a maximal consistent subset $\mathcal{H} \in C_{max}(\mathcal{K})$ rather than to an arbitrary consistent one. It thus appears natural to further enhance our methods by modifying instead of just removing or adding formulas.

Moreover, our previous theorems do not allow to discriminate between formulas. Usually, a knowledge base consists of information from several sources with different levels of reliability. For example, some information might be hard-coded meta-level observations, e.g., "If it is raining, the streets get wet." It does not appear to be beneficial to resolve conflicts by deleting formulas that encode such information.

To illustrate why further improvements of our results are desirable, let us consider the following example:

Example 3.3.1. Assume an agent needs to decide whether to carry an umbrella. In case it is going to rain outside, this is clearly the right choice. However, if the weather is good our agent would carry unnecessary items around, so:

$$bring\_umbrella \leftarrow rain. \qquad \neg bring\_umbrella \leftarrow sunshine.$$

The day before, the weather forecast predicted sunshine all day:

$$forecast(sunshine). \qquad X \leftarrow forecast(X).$$

However, by looking out of the window we observe that it is a rainy day:

$$obs(rain). \qquad X \leftarrow obs(X).$$

Since this is an inconsistent logic program P_{rain}, we face a situation where our agent does not know whether to bring an umbrella or not. Let us now see how to resolve this conflict in a reasonable way. The rules with non-trivial body are meta-level information, so we do not want to remove them. For example, if it is raining, we do not want to miss our umbrella, so we keep the rule "$bring\_umbrella \leftarrow rain.$". So we consider the facts "$forecast(sunshine).$" and "$obs(rain).$". The fact "$obs(rain).$" is due to an observation the agent just made with her own eyes. Hence, this information is reliable without any doubt. The weather forecast, on the other hand, might have predictions wrong, making the rule $r = forecast(sunshine).$ less reliable. We thus move to the consistent logic program

$$\begin{array}{lll} P_{rain} \setminus \{r\} : & bring\_umbrella \leftarrow rain. & \neg bring\_umbrella \leftarrow sunshine. \\ & X \leftarrow forecast(X). & \\ & obs(rain). & X \leftarrow obs(X). \end{array}$$

with answer set $\{obs(rain), rain, bring\_umbrella\}$. However, the problem here is actually caused by the rule $s = X \leftarrow forecast(X).$ which never doubts the weather forecast. Instead of removing the rule r as above, it is probably more reasonable to only believe the weather forecast if no contrary observation is made. Hence, we should weaken the rule s and replace it with $s' = X \leftarrow forecast(X),\ \text{not}\ obs(Y),\ X \neq Y.$ which does not allow us to infer $sunshine$ anymore due to $obs(rain)$. We thus end up with $P'_{rain} = P_{rain} \setminus \{s\} \cup \{s'\}$ given as follows:

$$\begin{array}{lll} P'_{rain} : & bring\_umbrella \leftarrow rain. & \neg bring\_umbrella \leftarrow sunshine. \\ & forecast(sunshine). & X \leftarrow forecast(X),\ \text{not}\ obs(Y),\ X \neq Y. \\ & obs(rain). & X \leftarrow obs(X). \end{array}$$

The program possesses the answer set $\{obs(rain), rain, bring\_umbrella\}$.

In this section, we demonstrate how to tackle situations of this kind. First, we assume that certain information are undisputed because of their reliability, so we want repairs not to allow for removal of certain formulas in $\mathcal{B} \subseteq \mathcal{K}$. Then, we investigate refining instead of deleting and adding formulas. Our goal is again to characterize the minimal modifications in terms of a hitting set duality. This raises the following questions:

- Can repairs that forbid removal of formulas in $\mathcal{B} \subseteq \mathcal{K}$ be characterized in a natural way?
- Given a formula $\alpha \in \mathcal{K}$, what does it in general mean that another formula β is *weaker* or *stronger* than α?
- Can we apply our former results, in particular Theorem 3.2.32?

In particular the answer to the last question is quite appealing: We will see that we do not require novel proofs within this section. Applied to an appropriate setup, Theorem 3.2.32 will provide us with the results we are looking for. This demonstrates the versatility of this theorem, based on an abstract definition of a logic.

3.3.1 Reliable Information

The first situation we discuss is about the existence of formulas $\mathcal{B} \subseteq \mathcal{K}$ which can be interpreted as reliable information. We thus introduce a notion of repairs which insists on $\mathcal{B} \subseteq \mathcal{K}$, i. e., removal of $\alpha \in \mathcal{B}$ yields an inconsistent knowledge base. This ensures that subsets of $\mathcal{K}$ which are accepted as "maximal consistent" contain $\mathcal{B}$.

Definition 3.3.2. For disjoint knowledge bases $\mathcal{K}$ and $\mathcal{G}$, and $\mathcal{B} \subseteq \mathcal{K}$, set

$$\textsc{bi-Rep}(\mathcal{K}, \mathcal{G}, \mathcal{B}) = \{(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Rep}(\mathcal{K}, \mathcal{G}) \mid \mathcal{D} \cap \mathcal{B} = \emptyset\}$$

and analogously

$$\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B}) = \{(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}) \mid \mathcal{D} \cap \mathcal{B} = \emptyset\}.$$

Observe that $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})$ contains exactly the bidirectional repairs we are interested in: They are minimal and do not allow for removal of any formula in $\mathcal{B}$. Our goal is now to characterize $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})$ in terms of a hitting set duality similar to Theorem 3.2.32. More precisely, we can apply this theorem to characterize $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})$.

For this, we start by defining an auxiliary logic. So assume we are given L with $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$. We construct $L_{\mathcal{B}} = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC}_{\mathcal{B}})$ in order to simulate the desired behavior. Thereto, we change $\mathcal{ACC}$ in a way that a knowledge base which does not contain $\mathcal{B}$ is rendered inconsistent. To achieve this, we want to obtain $\mathcal{ACC}_{\mathcal{B}}(\mathcal{K}) \subseteq \mathcal{INC}$ whenever $\mathcal{B} \nsubseteq \mathcal{K}$ (recall Definition 2.5.6). Since we do not know anything about $\mathcal{INC}$, we set $\mathcal{ACC}_{\mathcal{B}}(\mathcal{K}) = \emptyset$ in this case. Otherwise, $\mathcal{ACC}_{\mathcal{B}}(\mathcal{K})$ shall coincide with $\mathcal{ACC}(\mathcal{K})$. Thus, set

$$\mathcal{ACC}_{\mathcal{B}}(\mathcal{K}) := \begin{cases} \emptyset & \text{if } \mathcal{B} \nsubseteq \mathcal{K}, \\ \mathcal{ACC}(\mathcal{K}) & \text{otherwise.} \end{cases} \tag{3.2}$$

This yields our auxiliary logic $L_{\mathcal{B}}$. Note that we have two different logics L and $L_{\mathcal{B}}$ now. Recall our remark after Definition 2.5.6 that any notion in this work is wrt. a given logic, but we omit the superscript L for ease of presentation. For the remainder of this section we need to be precise in order to avoid confusion, so we give the superscripts.

The following lemma shows that the minimal bi-directional repairs wrt. $L_{\mathcal{B}}$ coincide with $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$. Thus, in order to obtain the desired outcome that no $\alpha \in \mathcal{B}$ is deleted, we can move to the auxiliary logic $L_{\mathcal{B}}$ and investigate the bi-directional repairs as before.

Lemma 3.3.3. *If L is a logic, $\mathcal{K}$ and $\mathcal{G}$ are disjoint knowledge bases and $\mathcal{B} \subseteq \mathcal{K}$, then* $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}} = \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$.

Proof. We show $\text{BI-REP}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}} = \text{BI-REP}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$.
"$\subseteq$": Assume $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$. Then $(\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}$ is consistent wrt. $L_\mathcal{B}$. Clearly, if a knowledge base is consistent wrt. $L_\mathcal{B}$, it must also be consistent wrt. L. We thus have $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}(\mathcal{K}, \mathcal{G})^L$. Moreover, consistency wrt. $L_\mathcal{B}$ implies $\mathcal{D} \cap \mathcal{B} = \emptyset$. Hence $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$.
"$\supseteq$": Now let $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$, i. e., $(\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}$ is consistent wrt. L and $\mathcal{D} \cap \mathcal{B} = \emptyset$. We have $\mathcal{B} \subseteq \mathcal{K}$ and thus $\mathcal{B} \subseteq (\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}$ due to $\mathcal{D} \cap \mathcal{B} = \emptyset$. Thus, by definition of $\mathcal{ACC}_\mathcal{B}$,

$$\mathcal{ACC}_\mathcal{B}((\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}) = \mathcal{ACC}((\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}).$$

Since $(\mathcal{K} \setminus \mathcal{D}) \cup \mathcal{A}$ is consistent wrt. L, it is also consistent wrt. $L_\mathcal{B}$. Hence we infer $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$. □

Given this lemma, the reader may already predict our technique to find a duality characterization for $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$: As we know this set coincides with $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$, Theorem 3.2.32 suggests to investigate $\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$. The outcome is formalized in the following lemma.

Lemma 3.3.4. *If L is a logic, $\mathcal{K}$ and $\mathcal{G}$ are disjoint knowledge bases and $\mathcal{B} \subseteq \mathcal{K}$, then* $\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}} = \{(\mathcal{D} \cup \mathcal{B}, \mathcal{A}) \mid (\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^L\}$.

Proof. "$\subseteq$": Assume $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$. Then $(\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ is inconsistent wrt. $L_\mathcal{B}$ for each $(\mathcal{D}', \mathcal{A}')$ with $(\mathcal{D}', \mathcal{A}') \subseteq (\mathcal{D}, \mathcal{A})$. Due to maximality of $(\mathcal{D}, \mathcal{A})$ we see $\mathcal{B} \subseteq \mathcal{D}$. So we may write $(\mathcal{D}, \mathcal{A}) = (\mathcal{D}, \tilde{\mathcal{A}} \cup \mathcal{B})$ with $\tilde{\mathcal{A}} = \mathcal{A} \setminus \mathcal{B}$. Now consider $(\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ with $(\mathcal{D}', \mathcal{A}') \subseteq (\mathcal{D}, \tilde{\mathcal{A}})$. By assumption, this knowledge base is inconsistent wrt. $L_\mathcal{B}$. Due to $\mathcal{A}' \subseteq \tilde{\mathcal{A}}$, we see $\mathcal{B} \subseteq (\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ and hence, $(\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ must be inconsistent wrt. L as well. Thus, $(\mathcal{D}, \tilde{\mathcal{A}}) \in \text{BI-NREP}(\mathcal{K}, \mathcal{G})^L$. Now wee see there is $\mathcal{B}' \subseteq \mathcal{B}$ such that $(\mathcal{D}, \tilde{\mathcal{A}} \cup \mathcal{B}') \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^L$. Hence, $(\mathcal{D}, \mathcal{A})$ is of the form $(\mathcal{D}, (\tilde{\mathcal{A}} \cup \mathcal{B}') \cup \mathcal{B})$ with $(\mathcal{D}, (\tilde{\mathcal{A}} \cup \mathcal{B}')) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^L$.
"$\supseteq$": Now assume $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})^L$. Then $(\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ is inconsistent wrt. L for each $(\mathcal{D}', \mathcal{A}') \subseteq (\mathcal{D}, \mathcal{A})$. In particular, this is also true wrt. $L_\mathcal{B}$. Since this means that $(\mathcal{K} \setminus (\mathcal{D}' \cup \mathcal{B}')) \cup \mathcal{A}'$ is also considered inconsistent wrt. $L_\mathcal{B}$ for each $\mathcal{B}' \subseteq \mathcal{B}$, we see $(\mathcal{D} \cup \mathcal{B}, \mathcal{A}) \in \text{BI-NREP}(\mathcal{K}, \mathcal{G})^{L_\mathcal{B}}$. We infer maximality since consistency wrt. L and $L_\mathcal{B}$ only differs in knowledge bases that do not contain $\mathcal{B}$. □

This motivates the following definition, which is similar to $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})$:

Definition 3.3.5. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\mathcal{B} \subseteq \mathcal{K}$. Set

- $\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G}, \mathcal{B}) := \{(\mathcal{D} \cup \mathcal{B}, \mathcal{A}) \mid (\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})\}$,
- $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G}, \mathcal{B}) := \{(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \mid (\mathcal{K} \setminus \overline{\mathcal{D}}, \mathcal{G} \setminus \overline{\mathcal{A}}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G}, \mathcal{B})\}$.

By applying Theorem 3.2.32 to the logic $L_\mathcal{B}$ we infer the following duality characterization of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})^L$:

Corollary 3.3.6. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\mathcal{B} \subseteq \mathcal{K}$. Then $\mathcal{S}$ is a minimal hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G}, \mathcal{B})$ if and only if $\mathcal{S} \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}, \mathcal{B})$.*

Example 3.3.7. For P and G we found that $\text{BI-REP}_{min}(P, G)$ consists of the following tuples:

$$(\{b \leftarrow a.\}, \emptyset),\ (\{c \leftarrow \text{not } b.\}, \emptyset),\ (\{\neg c \leftarrow \text{not } b.\}, \emptyset),\ (\emptyset, \{b.\}),\ (\emptyset, \{d.,\ b \leftarrow d.\}).$$

Now assume "$b \leftarrow a.$" is considered reliable and thus we do not want to remove this rule. So let $B = \{b \leftarrow a.\}$. The set $\text{BI-REP}_{min}(P, G, B)$ consists of the tuples

$$(\{c \leftarrow \text{not } b.\}, \emptyset), (\{\neg c \leftarrow \text{not } b.\}, \emptyset), (\emptyset, \{b.\}), (\emptyset, \{d.,\ b \leftarrow d.\}).$$

Recall

$$\text{BI-NREP}_{max}(P, G) = \{(\{a \vee b.\}, \{a., b \leftarrow d.\}),\ (\{a \vee b.\}, \{a., d.\})\}.$$

Due to Definition 3.3.5 $\text{BI-NREP}_{max}(P, G, B)$ consists of

$$(\{a \vee b.,\ b \leftarrow a.\}, \{a., b \leftarrow d.\}),\ (\{a \vee b.,\ b \leftarrow a.\}, \{a., d.\}),$$

i. e., $co\text{-BI-NREP}_{max}(P, G, B)$ contains

$$(\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}, \{b.,\ d.\}), (\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}, \{b.\, b \leftarrow d.\}).$$

The reader may verify that the duality claimed in Corollary 3.3.6 holds indeed.

We were thus able to take account of formulas $\mathcal{B} \subseteq \mathcal{K}$ which are not supposed to be removed as they are interpreted as reliable information. As already mentioned, achieving a duality characterization for the arising repair notion does not require novel proofs, but only applying Theorem 3.2.32 to an appropriate auxiliary logic $L_{\mathcal{B}}$.

3.3.2 Modifying Formulas

Let us continue with more fine-grained modifications to knowledge bases, namely weakening and strengthening instead of deleting and adding formulas. For that, we need a general notion for α being a stronger formula than β. Phrasing this within our setting requires some caution as we do not have a notion of a model of a formula. However, we can identify the following property of a "stronger" formula: Whenever it is present in $\mathcal{K}$, then the "weaker" formula should be subsumed. This can be expressed in terms of $\mathcal{ACC}(\mathcal{K})$. Formally, we define the following general entailment relation:

Definition 3.3.8. Let L be a logic. We say α *entails* β, denoted by $\alpha \models_L \beta$, iff for all knowledge bases $\mathcal{K}$, $\alpha \in \mathcal{K}$ implies $\mathcal{ACC}(\mathcal{K}) = \mathcal{ACC}(\mathcal{K} \cup \{\beta\})$.

To model modifications to formulas, we consider mappings w, s (weaker, stronger) of the form $w, s : \mathcal{K} \to \mathcal{WF} \setminus \mathcal{K}$ with $s(\alpha) \models_L \alpha \models_L w(\alpha)$. For technical reasons we assume the sets $w(\mathcal{K})$ and $s(\mathcal{K})$ to be disjoint. Note that both of them are disjoint with $\mathcal{K}$ by definition. We consider the knowledge base $\mathcal{K}$ after applying w and s to two disjoint subsets $\mathcal{H}_w, \mathcal{H}_s \subseteq \mathcal{K}$, i. e., we let

$$\mathcal{K}[w(\mathcal{H}_w), s(\mathcal{H}_s)] = (\mathcal{K} \setminus (\mathcal{H}_w \cup \mathcal{H}_s)) \cup (w(\mathcal{H}_w) \cup s(\mathcal{H}_s)).$$

Definition 3.3.9. Let $\mathcal{K}$ be a knowledge base. If $\mathcal{H}_w, \mathcal{H}_s \subseteq \mathcal{K}$ are disjoint and the knowledge base $\mathcal{K}[w(\mathcal{H}_w), s(\mathcal{H}_s)]$ is consistent, then we call $(\mathcal{H}_w, s(\mathcal{H}_s))$ a *consistency-restoring modification* of $\mathcal{K}$ wrt. w and s. Let $\text{MOD}_{min}(\mathcal{K})$ denote the set of all minimal consistency-restoring modifications of $\mathcal{K}$ wrt. w and s.

Note the intended asymmetry in the tuple $(\mathcal{H}_w, s(\mathcal{H}_s))$, which is chosen to conveniently phrase Lemmata 3.3.10 and 3.3.11 below. Again we want to characterize $\text{MOD}_{min}(\mathcal{K})$ in terms of a hitting set duality. For this, consider $\tilde{\mathcal{K}}$ and $\tilde{\mathcal{G}}$ given as $\tilde{\mathcal{K}} = \mathcal{K} \cup w(\mathcal{K})$, i. e., $\tilde{\mathcal{K}}$ contains in addition the weakened formulas and $\tilde{\mathcal{G}} = s(\mathcal{K})$, i. e., $\tilde{\mathcal{G}}$ consists of the strengthened formulas. Please note that $\mathcal{ACC}(\tilde{\mathcal{K}}) = \mathcal{ACC}(\mathcal{K})$. In particular, if we remove $\alpha \in \mathcal{K}$ from $\tilde{\mathcal{K}}$, then the latter still contains $w(\alpha)$, i. e., $\mathcal{ACC}(\tilde{\mathcal{K}} \setminus \{\alpha\}) = \mathcal{ACC}(\mathcal{K}[w(\{\alpha\}), s(\emptyset)])$. Hence, this corresponds to weakening α. Similarly, adding a formula $s(\alpha) \in \tilde{G}$ to $\tilde{\mathcal{K}}$ corresponds to strengthening α. This almost induces a characterization of $\text{MOD}_{min}(\mathcal{K})$:

Lemma 3.3.10. *If $\mathcal{K}$ is a knowledge base, then*

$$\text{BI-REP}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}) = \text{MOD}_{min}(\mathcal{K}) \cup \{(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}) \mid w(\mathcal{K}) \cap \mathcal{D} \neq \emptyset\}.$$

Proof. We prove $\text{MOD}_{min}(\mathcal{K}) = \{(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}) \mid w(\mathcal{K}) \cap \mathcal{D} = \emptyset\}$.
"$\subseteq$": Assume $(\mathcal{D}, \mathcal{A}) \in \text{MOD}_{min}(\mathcal{K})$. This means $(\mathcal{D}, \mathcal{A})$ is of the form $(\mathcal{H}_w, s(\mathcal{H}_s))$, where $\mathcal{H}_w, \mathcal{H}_s \subseteq \mathcal{K}$ with $\mathcal{H}_w \cap \mathcal{H}_s = \emptyset$, and

$$\mathcal{K}[w(\mathcal{H}_w), s(\mathcal{H}_s)] = (\mathcal{K} \setminus (\mathcal{H}_w \cup \mathcal{H}_s)) \cup (w(\mathcal{H}_w) \cup s(\mathcal{H}_s))$$

is consistent. Consider now $(\tilde{\mathcal{K}} \setminus \mathcal{H}_w) \cup s(\mathcal{H}_s)$. Recall that $\tilde{\mathcal{K}} = \mathcal{K} \cup w(\mathcal{K})$ and thus

$$(\tilde{\mathcal{K}} \setminus \mathcal{H}_w) \cup s(\mathcal{H}_s) = \mathcal{K} \setminus \mathcal{H}_w \cup (w(\mathcal{K}) \cup s(\mathcal{H}_s)).$$

Due to $\alpha \models_L w(\alpha)$ for each $\alpha \in \mathcal{K}$ we can add $w(\alpha)$ to the knowledge base without changing the semantics, as long as α itself is present. Hence, only adding $w(\mathcal{H}_w)$ needs to be taken into account, yielding

$$\mathcal{ACC}(\mathcal{K} \setminus \mathcal{H}_w \cup (w(\mathcal{K}) \cup s(\mathcal{H}_s))) = \mathcal{ACC}(\mathcal{K} \setminus \mathcal{H}_w \cup (w(\mathcal{H}_s) \cup s(\mathcal{H}_s))).$$

Due to $s(\alpha) \models_L \alpha$ for each $\alpha \in \mathcal{H}_s$ removal of $\mathcal{H}_s$ does not change the semantics of the knowledge base as long as $s(\mathcal{H}_s)$ is present, so

$$\mathcal{ACC}(\mathcal{K} \setminus \mathcal{H}_w \cup (w(\mathcal{H}_s) \cup s(\mathcal{H}_s))) = \mathcal{ACC}(\mathcal{K} \setminus (\mathcal{H}_w \cup \mathcal{H}_s) \cup (w(\mathcal{H}_s) \cup s(\mathcal{H}_s))).$$

In summary, we have

$$\mathcal{ACC}((\tilde{\mathcal{K}} \setminus \mathcal{H}_w) \cup s(\mathcal{H}_s)) = \mathcal{ACC}((\mathcal{K} \setminus (\mathcal{H}_w \cup \mathcal{H}_s)) \cup (w(\mathcal{H}_w) \cup s(\mathcal{H}_s))). \tag{3.3}$$

Since the latter is consistent and $\mathcal{H}_w \subseteq \tilde{\mathcal{K}}$ as well as $s(\mathcal{H}_s) \subseteq \tilde{\mathcal{G}}$, we are able to infer that $(\mathcal{H}_w, s(\mathcal{H}_s)) \in \text{BI-REP}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})$. The equation (3.3) also shows that minimality is preserved. Finally, we have $w(\mathcal{K}) \cap \mathcal{H}_w = \emptyset$ since w is a mapping $w : \mathcal{K} \to \mathcal{WF} \setminus \mathcal{K}$. Altogether, $(\mathcal{H}_w, s(\mathcal{H}_s)) = (\mathcal{D}, \mathcal{A}) \in \{(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}) \mid w(\mathcal{K}) \cap \mathcal{D} = \emptyset\}$.
"$\supseteq$": Now assume $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})$ with $w(\mathcal{K}) \cap \mathcal{D} = \emptyset$. Simply set $\mathcal{H}_w = \mathcal{D}$ and $\mathcal{H}_s = s^{-1}(\mathcal{A})$. We now have consistency of

$$(\tilde{\mathcal{K}} \setminus \mathcal{H}_w) \cup s(\mathcal{H}_s)$$

by assumption and since we may infer (3.3) as above, we find consistency of

$$(\mathcal{K} \setminus (\mathcal{H}_w \cup \mathcal{H}_s)) \cup (w(\mathcal{H}_w) \cup s(\mathcal{H}_s))$$

as well. Again minimality is clearly preserved, so $(\mathcal{H}_w, s(\mathcal{H}_s))\text{MOD}_{min}(\mathcal{K})$. □

The above lemma does not yield a characterization of $\textsc{Mod}_{min}(\mathcal{K})$ since one may delete formulas in $w(\mathcal{K})$ in order to render $\tilde{\mathcal{K}}$ consistent, which does not correspond to weakening a formula of the initial knowledge base. Depending on the desired outcome, one could either accept this or apply the technique from above in order to forbid removal of $\mathcal{B} := w(\mathcal{K}) \subseteq \tilde{\mathcal{K}}$. If we define $L_{\mathcal{B}}$ as above, Lemma 3.3.10 yields the desired characterization of $\textsc{Mod}_{min}(\mathcal{K})$.

Lemma 3.3.11. *If $\mathcal{K}$ be a knowledge base, then* $\textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{\mathcal{B}}} = \textsc{Mod}_{min}(\mathcal{K})^L$.

Proof. This is a simple corollary of Lemma 3.3.10 when applying Lemma 3.3.3. We have

$$\textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^L = \textsc{Mod}_{min}(\mathcal{K})^L \cup \{(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^L \mid w(\mathcal{K}) \cap \mathcal{D} \neq \emptyset\}$$

and moreover, $\textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{w(\mathcal{K})}} = \textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}, w(\mathcal{K}))^L$. Hence,

$$\begin{aligned}
\textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{w(\mathcal{K})}} &= \textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}, w(\mathcal{K}))^L \\
&= \{(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^L \mid (\mathcal{D} \cap w(\mathcal{K})) = \emptyset\} \\
&= \textsc{Mod}_{min}(\mathcal{K})^L.
\end{aligned}$$

So $\textsc{bi-Rep}_{min}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{w(\mathcal{K})}}$ captures exactly the tuples in $\textsc{Mod}_{min}(\mathcal{K})^L$. □

By applying Theorem 3.2.32 we find a duality in terms of $\textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{\mathcal{B}}}$, so we are interested in the nature of this set.

Lemma 3.3.12. *Let $\mathcal{K}$ be a knowledge base. Then $\textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{\mathcal{B}}}$ is the collection of all maximal tuples $(\mathcal{D} \cup \mathcal{B}, \mathcal{A}) = (\mathcal{H}_w \cup w(\mathcal{K}), s(\mathcal{H}_s))$ such that the knowledge base $\mathcal{K}[w(\mathcal{H}'_w), s(\mathcal{H}'_s)]$ is inconsistent for all $(\mathcal{H}'_w, \mathcal{H}'_s) \subseteq (\mathcal{H}_w, \mathcal{H}_s)$.*

Proof. By definition, $\textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})$ consists of all maximal tuples $(\mathcal{H}_w, s(\mathcal{H}_s))$ such that $\mathcal{K}[w(\mathcal{H}'_w), s(\mathcal{H}'_s)]$ is inconsistent for all $(\mathcal{H}'_w, \mathcal{H}'_s) \subseteq (\mathcal{H}_w, \mathcal{H}_s)$. By Lemma 3.3.4,

$$\begin{aligned}
\textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{\mathcal{B}}} &= \textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}}, \mathcal{B})^L \\
&= \{(\mathcal{D} \cup \mathcal{B}, \mathcal{A}) \mid (\mathcal{D}, \mathcal{A}) \in \textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^L\}
\end{aligned}$$

so the claim follows. □

Finally, applying Theorem 3.2.32 yields the desired duality.

Corollary 3.3.13. *Let $\mathcal{K}$ be a knowledge base and $\tilde{\mathcal{K}}, \tilde{\mathcal{G}}$ and $\mathcal{B}$ as above. Then $\mathcal{S}$ is a minimal hitting set of co-$\textsc{bi-NRep}_{max}(\tilde{\mathcal{K}}, \tilde{\mathcal{G}})^{L_{\mathcal{B}}}$ iff $\mathcal{S} \in \textsc{Mod}_{min}(\mathcal{K})^L$.*

The following scheme depicts how to utilize the results we obtained in this section in order to repair a given knowledge base in a more fine-grained fashion:

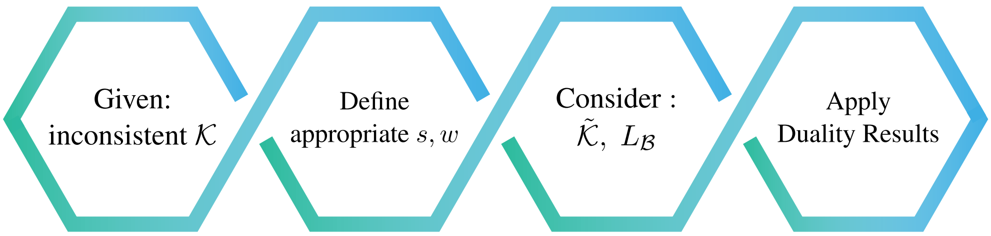


Figure 3.13: Summary of Scetion 3.3: Given an inconsistent knowledge base, we need to define our mappings w and s to refine formulas. We then consider an appropriate auxiliary knowledge base and apply the duality results from before.

3.4 Infinite Knowledge Bases

So far, we were only concerned with *finite* knowledge bases. This appears to be natural since one can expect the knowledge of an agent to be finite or at least representable in a finite way. However finite models are not always sufficient to capture real world applications, mainly due to the reason that an upper bound for the size of our knowledge base is not known. For example a security camera needs to collect data until an unknown point of time in the future. Even though this camera is going to be turned off eventually, we do not know when this will be. It would thus not make sense to fix a size of a knowledge base when maintaining the data collected by this camera.

We thus generalize our previous definition of a knowledge base to an infinite set. For ease of presentation we restrict our investigation to countable knowledge bases.

Definition 3.4.1. Let $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ be a logic. An *infinite knowledge base* $\mathcal{K}$ of L is a countably infinite subset of $\mathcal{WF}$, i. e., there is a bijection $\rho : \mathcal{K} \to \mathbb{N}$.

Our notions we previously defined for finite knowledge bases generalize in the expected way. We want to investigate the sets $C_{max}(\mathcal{K})$, $SI_{min}(\mathcal{K})$ and their connection. For this, we recall that the proof of the duality Theorem 3.1.12 is in principle based on two observations:

- Consistency and inconsistency are complementary properties.
- When considering finite sets, we can always turn a set with a certain property into a minimal or maximal one possessing this property.

Clearly the former does not change in the context of infinite knowledge bases since it is true by definition. Thus, the good news is: If we are given a minimal hitting set $\mathcal{S}$ of $SI_{min}(\mathcal{K})$, the complement $\mathcal{K} \setminus \mathcal{S}$ is maximal consistent as in the finite case. However, the existence of *minimal* (strongly) inconsistent subsets or *minimal* hitting sets of those is no longer guaranteed. For example, given an inconsistent set $\mathcal{H} \subseteq \mathcal{K}$ we do not know whether there is a minimal inconsistent set $\mathcal{H}'$ with $\mathcal{H}' \subseteq \mathcal{H} \subseteq \mathcal{K}$.

Due to the hitting set duality, the existence of maximal consistent subsets is closely related to the existence of minimal (strongly) inconsistent subsets and minimal hitting sets, which leads to the following questions:

- Given a (strongly) inconsistent set $\mathcal{H} \subseteq \mathcal{K}$, is there a minimal one $\mathcal{H}' \subseteq \mathcal{H}$?
- Given a hitting set $\mathcal{S}$ of $SI_{min}(\mathcal{K})$, is there a minimal one $\mathcal{S}' \subseteq \mathcal{S}$?
- Given a consistent set $\mathcal{H} \subseteq \mathcal{K}$, is there a maximal one $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$?
- How are the answers to these questions related?

3.4.1 On the Absence of Minimality and Maximality

Due to the generality of our notion of logics and knowledge bases, it should be no surprise that the answer to the above questions is negative in general. We will now consider several examples to illustrate this. They will also motivate the restrictions which we consider.

Let us focus on inconsistent subsets of an infinite knowledge base $\mathcal{K}$ first. Recall that in the finite case, we consider $SI_{min}(\mathcal{K})$, i. e., the *minimal* strongly inconsistent subsets, and require a *minimal* hitting set of it. The following example illustrates that an infinite inconsistent knowledge base does not necessarily possess a minimal strongly inconsistent subset.

Example 3.4.2. Let $A = \{a_i \mid i \in \mathbb{N}\}$ be a countable set of arguments. We consider the AF represented by the knowledge base $\mathcal{K} = \{(a_i, a_j) \mid j < i, i, j \in \mathbb{N}\}$, so each a_i attacks all arguments with a strictly smaller index:

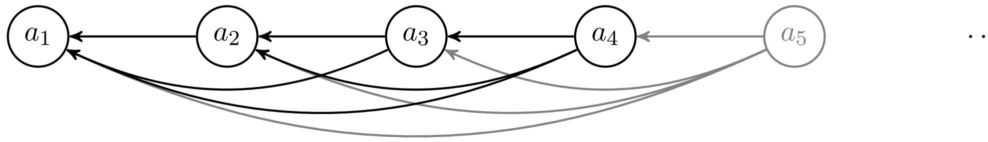


Figure 3.14: The AF from Example 3.4.2.

Observe that $\mathcal{K}$ is inconsistent: Assume $a_i \in E$ for a set $E \subseteq A$ of arguments. If E is conflict-free, then $a_{i+1} \notin E$. Hence, in order for E to be a stable extension, there must be an index $j \geq i + 2$ with $a_j \in E$. Otherwise a_{i+1} is not attacked by E. Since $(a_j, a_i) \in \mathcal{K}$, we see that E with $a_i, a_j \in E$ is not conflict-free and thus no stable extension, so $\mathcal{K}$ is inconsistent. We thus have $\mathcal{K} \in SI(\mathcal{K})$. In particular, $SI(\mathcal{K}) \neq \emptyset$. However, there is no minimal strongly inconsistent subset, i. e., $SI_{min}(\mathcal{K}) = \emptyset$. To see this, let $\mathcal{H} \subseteq \mathcal{K}$. We show that $\mathcal{H} \notin SI_{min}(\mathcal{K})$.

First assume $\mathcal{H}$ is finite. Then there is an index $k \in \mathbb{N}$ such that

$$\mathcal{H} \subseteq \bigcup_{1 \leq i < j \leq k} \{(a_i, a_j)\}.$$

Now consider

$$\mathcal{H}' = \mathcal{H} \cup \bigcup_{1 \leq j \leq k} \{(a_{k+1}, a_j)\},$$

i. e., we add the $(k + 1)$-th argument and all its attacks. Of course, $\mathcal{H}' \subseteq \mathcal{K}$. Moreover, $stb(\mathcal{H}') = \bigcup_{i \geq k+1} a_i$ (recall that we only remove attacks, so each AF contains all arguments), so $\mathcal{H}'$ is consistent. Thus, $\mathcal{H} \notin SI(\mathcal{K})$. In particular, $\mathcal{H} \notin SI_{min}(\mathcal{K})$.

Now assume $\mathcal{H}$ is infinite with $\mathcal{H} \in SI(\mathcal{K})$. Consider an arbitrary argument a_i such that there is a $j < i$ with $(a_i, a_j) \in \mathcal{H}$. We claim that $\mathcal{H} \setminus \{(a_i, a_j)\}$ is strongly inconsistent as well. Assume this is not the case, then there is a consistent set $\mathcal{H}'$ with $\mathcal{H} \setminus \{(a_i, a_j)\} \subseteq \mathcal{H}'$. Assume E is a stable extension of $\mathcal{H}'$. Consider an argument $a_k \in E$ with $k > i$. Intuitively, consistency of $\mathcal{H}'$ does not depend on the finitely many arguments $a_1, \ldots, a_k$, so we may augment $\mathcal{H}'$ with all attacks (a_m, a_n) with $m, n \leq k$. Due to potential conflicts, E is not necessarily a stable extension of this augmented framework. More precisely, this is the case if there is an argument $a_t \in E$ with $t < k$. Formally, the framework represented by

$$\mathcal{H}'' = \mathcal{H}' \cup \bigcup_{1 \leq m < n \leq k} \{(a_n, a_m)\}$$

possesses the stable extension $E'' = E \setminus \{a_i \mid i < k\}$. We have $\mathcal{H} \subseteq \mathcal{H}''$ and $\mathcal{H}''$ is consistent, so $\mathcal{H}$ is not strongly inconsistent, which is a contradiction, so the assumption that $\mathcal{H} \setminus \{(a_i, a_j)\}$ is not strongly inconsistent must be false. Since $\mathcal{H}$ was an arbitrary set in $SI(\mathcal{K})$ and we considered removal of an arbitrary attack, we infer that there is no minimal strongly inconsistent subset of $\mathcal{K}$.

In order to ensure the existence of minimal hitting sets of $SI_{min}(\mathcal{K})$, we explicitly require the existence of sets in $SI_{min}(\mathcal{K})$. However, even if $SI_{min}(\mathcal{K})$ is non-empty, it might be the case that some $\mathcal{H} \in SI(\mathcal{K})$ do not contain a minimal strongly inconsistent subset (this can be seen by a straightforward adjustment to the above example). In this case, we cannot move from $\mathcal{H}$ to a set $\mathcal{H}' \subseteq \mathcal{H}$ with $\mathcal{H}' \in SI_{min}(\mathcal{K})$, which was essential in the proof of Theorem 3.1.12. However, we require minimal sets for our hitting set characterization. This can already be seen for finite knowledge bases:

Example 3.4.3. For $\mathcal{K} = \{a, \neg a, b, c\}$ consider $\mathcal{H} = \{a, \neg a, b\} \in SI(\mathcal{K})$. Clearly, $\mathcal{H}$ is not minimal. We have the minimal hitting set $\mathcal{S} = \{b\}$, but $\mathcal{K} \setminus \mathcal{S}$ is inconsistent.

Of course, one could require the missing property explicitly, i. e., consider only knowledge bases $\mathcal{K}$ where each $\mathcal{H} \subseteq \mathcal{K}$ with $\mathcal{H} \in SI(\mathcal{K})$ contains a minimal strongly inconsistent subset. As it turns out, this is still not sufficient in general, because now we face the same issue regarding the hitting sets. More precisely, given an infinite set $\mathcal{X}$ of sets, i. e., $X_i \in \mathcal{X}$ for $i \in \mathbb{N}$, and given a hitting set $\mathcal{S}$ of $\mathcal{X}$, then there is not necessarily a *minimal* hitting set $\mathcal{S}' \subseteq \mathcal{S}$ of $\mathcal{X}$. To see this, consider the following example.

Example 3.4.4. Consider $\mathcal{X} = \{X_i \mid i \in \mathbb{N}\}$ with $X_i \subseteq \mathbb{Z}$ for each i and let

$$X_i = \{-i\} \cup \{n \in \mathbb{N} \mid n \geq i\}.$$

Observe that $X_i \nsubseteq X_j$ for $i \neq j$, so one could construct an (artificial) logic with an infinite knowledge base $\mathcal{K}$ such that $SI_{min}(\mathcal{K}) = \mathcal{X}$. We want to show that there is a hitting set $\mathcal{S}$ of $\mathcal{X}$ which cannot be turned into a minimal one, namely $\mathcal{S} = \mathbb{N}$. This is of course a hitting set of $\mathcal{X}$, so let $\mathcal{S}' \subseteq \mathcal{S}$ be a hitting set of $\mathcal{X}$ as well. We prove that $\mathcal{S}'$ is not minimal. For this, let $n \in \mathbb{N}$ be an integer with $n \in \mathcal{S}'$ and observe that $\mathcal{S}' \setminus \{n\}$ is also a hitting set of $\mathcal{X}$. Formally, by assumption we have $\mathcal{S}' \cap X_{n+1} \neq \emptyset$, so there must be an integer $m \geq n + 1$ with $m \in \mathcal{S}'$. However, $\{n\} \cap X_i \neq \emptyset$ implies $\{m+1\} \cap X_i \neq \emptyset$, so $\mathcal{S}' \setminus \{n\}$ is a hitting set of $\mathcal{X}$ as well. Since $\mathcal{S}'$ was arbitrary, there is no minimal hitting set of $\mathcal{X}$ which is a subset of $\mathcal{S}$.

It is clear that similar observations can be made for $C_{max}(\mathcal{K})$. For example, even though a knowledge base $\mathcal{K}$ might possess consistent subsets, it is in general possible that $C_{max}(\mathcal{K})$ is empty.

3.4.2 The Minimal Inconsistency And The Maximal Consistency Property

To overcome the issues pointed out during the previous examples, we need to restrict the considered knowledge base. We do this as follows:

Definition 3.4.5 (Minimal inconsistency property)**.** An inconsistent infinite knowledge base $\mathcal{K}$ satisfies the *minimal inconsistency property* if

- given a set $\mathcal{H} \in SI(\mathcal{K})$, there is a set $\mathcal{H}' \subseteq \mathcal{H}$ with $\mathcal{H}' \in SI_{min}(\mathcal{K})$,
- given a hitting set $\mathcal{S}$ of $SI_{min}(\mathcal{K})$, there is a minimal hitting set $\mathcal{S}' \subseteq \mathcal{S}$ of $SI_{min}(\mathcal{K})$.

Given this property, the collection of maximal consistent subsets of an infinite knowledge base is as well-behaving as in the finite case.

Proposition 3.4.6. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base satisfying the minimal inconsistency property. If $\mathcal{H} \subseteq \mathcal{K}$ is a consistent subset of $\mathcal{K}$, then there is a maximal consistent subset $\mathcal{H}' \in C_{max}(\mathcal{K})$ with $\mathcal{H} \subseteq \mathcal{H}'$. In particular, $\mathcal{H}' = \mathcal{K} \setminus \mathcal{S}$ for a minimal hitting set $\mathcal{S}$ of $SI_{min}(\mathcal{K})$.*

Proof. Let $\mathcal{H} \subseteq \mathcal{K}$ be consistent. We show that $\mathcal{S} = \mathcal{K} \setminus \mathcal{H}$ is a hitting set of $SI_{min}(\mathcal{K})$. Assume this is not the case, i. e., there is a set $\mathcal{I} \in SI_{min}(\mathcal{K})$ with $\mathcal{S} \cap \mathcal{I} \neq \emptyset$. Hence, $\mathcal{I} \subseteq \mathcal{K} \setminus \mathcal{S}$ and in particular $\mathcal{I} \subseteq \mathcal{K} \setminus \mathcal{S} = \mathcal{H}$. Since $\mathcal{I}$ is strongly inconsistent, the set $\mathcal{H}$ with $\mathcal{I} \subseteq \mathcal{H}$ is inconsistent, which contradicts our assumption. We thus infer that $\mathcal{S} = \mathcal{K} \setminus \mathcal{H}$ is a hitting set of $SI_{min}(\mathcal{K})$. Due to the minimal inconsistency property, there is a minimal hitting set $\mathcal{S}'$ of $SI_{min}(\mathcal{K})$ with $\mathcal{S}' \subseteq \mathcal{S}$.

It is left to show that $\mathcal{H}' = \mathcal{K} \setminus \mathcal{S}'$ is *maximal* consistent. This can be done similarly to the proof of Theorem 3.1.12: If there is a consistent set $\mathcal{H}''$ with $\mathcal{H}' \subsetneq \mathcal{H}''$, it is easy to show that $\mathcal{K} \setminus \mathcal{H}''$ is a hitting set of $SI_{min}(\mathcal{K})$ as well, which contradicts minimality of $\mathcal{S}'$. Thus, each proper superset of $\mathcal{H}'$ is inconsistent. Furthermore, assume $\mathcal{H}'$ is not consistent. Since each superset of $\mathcal{H}'$ is inconsistent (as seen before), $\mathcal{H}'$ is in particular strongly inconsistent. We utilize the minimal inconsistency property of $\mathcal{K}$ again to infer that there is a minimal strongly inconsistent set $\mathcal{H}''$ with $\mathcal{H}'' \subseteq \mathcal{H}'$. Since $\mathcal{H}'' \subseteq \mathcal{H}' = \mathcal{K} \setminus \mathcal{S}'$, we obtain $\mathcal{H}'' \cap \mathcal{S}' = \emptyset$, so $\mathcal{S}'$ is not a hitting set of $SI_{min}(\mathcal{K})$. This is a contradiction. Therefore, $\mathcal{H}' \in C_{max}(\mathcal{K})$. Since $\mathcal{H}$ was an arbitrary consistent set, the claim follows. □

The minimal inconsistency property ensures that consistent subsets can always be turned into maximal consistent ones. This leads to the question whether the converse holds as well, i. e., can we guarantee the minimal inconsistency property if each consistent subset $\mathcal{H}$ of $\mathcal{K}$ can be turned into a maximal consistent one? The answer to this particular question is negative, but this will not disturb us in any way. For this, recall Theorem 3.1.19, i. e., $\mathcal{S}$ is a minimal hitting set of $coC_{max}(\mathcal{K})$ if and only if $\mathcal{S} \in SI_{min}(\mathcal{K})$. Hence, to ensure that a given set $\mathcal{H} \in SI(\mathcal{K})$ contains a minimal strongly inconsistent set, we need to ensure that a given hitting set $\mathcal{S}$ of $coC_{max}(\mathcal{K})$ contains a minimal hitting set $\mathcal{S}' \subseteq \mathcal{S}$.

Although this cannot be guaranteed in general as seen in Example 3.4.4, the minimal inconsistency property ensures the required structure of hitting sets of $coC_{max}(\mathcal{K})$:

Proposition 3.4.7. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base satisfying the minimal inconsistency property. If $\mathcal{S}$ is a hitting set of $coC_{max}(\mathcal{K})$, then there is a minimal hitting set $\mathcal{S}' \subseteq \mathcal{S}$ of $coC_{max}(\mathcal{K})$. In particular, $\mathcal{S}' \in SI_{min}(\mathcal{K})$.*

Proof. To see this, we prove that $\mathcal{S} \in SI(\mathcal{K})$ if and only if $\mathcal{S}$ is a hitting set of $coC_{max}(\mathcal{K})$.
"$\Leftarrow$": Let $\mathcal{S}$ be a hitting set of $coC_{max}(\mathcal{K})$. Assume for the sake of contradiction that $\mathcal{S}$ is not in $SI(\mathcal{K})$. Then, there is a consistent set $\mathcal{H}$ with $\mathcal{S} \subseteq \mathcal{H}$. According to Proposition 3.4.6, there is a maximal consistent set $\mathcal{H}' \in C_{max}(\mathcal{K})$ with $\mathcal{H} \subseteq \mathcal{H}'$. In particular,

$$\mathcal{S} \cap (\mathcal{K} \setminus \mathcal{H}') \subseteq \mathcal{S} \cap (\mathcal{K} \setminus \mathcal{H}) = \emptyset.$$

Hence, $\mathcal{S}$ is not a hitting set of $coC_{max}(\mathcal{K})$, which is a contradiction.
"$\Rightarrow$": Now let $\mathcal{S} \in SI(\mathcal{K})$. Similarly, assume $\mathcal{S}$ is not a hitting set of $coC_{max}(\mathcal{K})$. Then, there is a set $\mathcal{H} \in coC_{max}(\mathcal{K})$ with $\mathcal{H} \cap \mathcal{S} = \emptyset$. By definition, $\mathcal{K} \setminus \mathcal{H}$ is consistent and since $\mathcal{H} \cap \mathcal{S} = \emptyset$, we have $\mathcal{S} \subseteq \mathcal{K} \setminus \mathcal{H}$ which contradicts strong inconsistency of $\mathcal{S}$. □

The previous result motivates considering a dual counterpart to the minimal inconsistency property, defined as follows:

Definition 3.4.8 (Maximal consistency property)**.** An inconsistent infinite knowledge base $\mathcal{K}$ satisfies the *maximal consistency property* if

- given a consistent set $\mathcal{H} \subseteq \mathcal{K}$, there is a set $\mathcal{H}' \in C_{max}(\mathcal{K})$ with $\mathcal{H} \subseteq \mathcal{H}'$,
- given a hitting set $\mathcal{S}$ of $coC_{max}(\mathcal{K})$, there is a minimal hitting set $\mathcal{S}'$ of $coC_{max}(\mathcal{K})$ with $\mathcal{S}' \subseteq \mathcal{S}$.

Combining Proposition 3.4.6 and Proposition 3.4.7, we get: If $\mathcal{K}$ satisfies the minimal hitting set property, then it satisfies the maximal consistency property as well. As already mentioned, the converse is also true:

Proposition 3.4.9. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base. If $\mathcal{K}$ satisfies the maximal consistency property, then it satisfies the minimal hitting set property as well.*

Proof. Let $\mathcal{H} \in SI(\mathcal{K})$. As seen in the proof of Proposition 3.4.7, this is the case if and only if $\mathcal{H}$ is a hitting set of $coC_{max}(\mathcal{K})$. By assumption, there is a minimal hitting set $\mathcal{H}'$ of $coC_{max}(\mathcal{K})$ with $\mathcal{H}' \subseteq \mathcal{H}$. Hence, $\mathcal{H}' \in SI_{min}(\mathcal{K})$.

Now assume we are given a hitting set $\mathcal{S}$ of $SI_{min}(\mathcal{K})$. Then, $\mathcal{H} = \mathcal{K} \setminus \mathcal{S}$ is consistent. Again by assumption, there is a set $\mathcal{H}' \in C_{max}(\mathcal{K})$ with $\mathcal{H} \subseteq \mathcal{H}'$ and as usual we see that now, $\mathcal{S}' = \mathcal{K} \setminus \mathcal{H}'$ must be a minimal hitting set of $SI_{min}(\mathcal{K})$. □

So, the results of this section can be summarized as follows.

Theorem 3.4.10. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base. Then, $\mathcal{K}$ satisfies the minimal inconsistency property if and only if it satisfies the maximal consistency property. Moreover, $\mathcal{S}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$ and $\mathcal{S}$ is a minimal hitting set of $coC_{max}(\mathcal{K})$ if and only if $\mathcal{S} \in SI_{min}(\mathcal{K})$.*

This result emphasizes the fact that infinite knowledge bases are similar in their spirit to finite ones, except for the lack of maximal resp. minimal sets with certain properties. Whenever we make appropriate assumptions about our knowledge base (for example the minimal inconsistency property), we can work with our hitting set duality as usual. The definition of the minimal inconsistency property is tailored ensures the existence of maximal consistent sets. Interestingly, it also yields an analogous result for the hitting sets of $coC_{max}(\mathcal{K})$. The converse Proposition 3.4.9 emphasizes the close link between the notions we investigate.

3.4.3 Compact Logics

Having established Theorem 3.4.10, a naturally arising questions is which significant examples of logics satisfy the minimal inconsistency property. A rather popular result about inconsistency of propositional knowledge bases is the compactness theorem.

Theorem 3.4.11 (Compactness Theorem, see [94])**.** *If $\mathcal{K}$ is a propositional knowledge base, then $\mathcal{K}$ is consistent iff each finite subsets $\mathcal{H} \subseteq \mathcal{K}$ is.*

One would expect this property to ensure a pleasant behavior of inconsistent sets as well as their hitting sets. Indeed, it is easy to see that propositional knowledge bases satisfy the first item of the minimal inconsistency property: $\mathcal{K}$ is inconsistent iff there is a finite inconsistent subset $\mathcal{H}$. Hence, given an inconsistent subset $\mathcal{H} \subseteq \mathcal{K}$, there is a finite minimal inconsistent subset contained in $\mathcal{H}$.

For the second item of the minimal inconsistency property, we need to take the hitting sets into account. The compactness theorem ensures that each $\mathcal{H} \in SI_{min}(\mathcal{K})$ is finite and hence, $\mathcal{S}$ must be a hitting set of a collection of *finite* sets. In fact, this ensures that any hitting set can be turned into a minimal one. This result does not seem to be obvious and has, to the best of our knowledge, not been stated explicitly before.

Lemma 3.4.12. *Let $\mathcal{X}$ be a set of finite subsets of a countable set X. If $\mathcal{S}$ is a hitting set of $\mathcal{X}$, then there is a minimal hitting set $\mathcal{S}' \subseteq \mathcal{S}$ of $\mathcal{X}$.*

Proof. Note that $\mathcal{X}$ is a countable collection of sets, so we may assume $\mathcal{X}$ is of the form $\mathcal{X} = \{X_i \mid i \in \mathbb{N}\}$ with $X_i \subseteq X$ for each $i \in \mathbb{N}$.

Let $\mathcal{S}$ be a hitting set of $\mathcal{X}$. We construct a minimal hitting set $\mathcal{S}'$ of $\mathcal{X}$ inductively as follows. Let us first consider X_1. Since X_1 is finite, the set $\mathcal{S} \cap X_1$ is finite as well. Consider a maximal set Y_1 of removable elements in $X_1 \cap \mathcal{S}$, i. e., let

$$Y_1 \in \max\{Y \subseteq \mathcal{S} \cap X_1 \mid \mathcal{S} \setminus Y \text{ is a hitting set of } \mathcal{X}\}$$

where at least one such maximal set exists since $\mathcal{S} \cap X_1$ is finite. Moreover, if we are given $Y_1, \ldots, Y_{n-1}$, then we set

$$Y_n \in \max\left\{Y \subseteq \left(\mathcal{S} \setminus \bigcup_{i=1}^{n-1} Y_i\right) \cap X_n \mid \mathcal{S} \setminus \left(\bigcup_{i=1}^{n-1} Y_i \cup Y\right) \text{ is a hitting set of } \mathcal{X}\right\} \quad (3.4)$$

where again at least one such maximal set exists since

$$\left(\mathcal{S} \setminus \bigcup_{i=1}^{n-1} Y_i\right) \cap X_n$$

is finite. Note that the Y_i are pairwise disjoint by definition. We claim that

$$\mathcal{S}' := \mathcal{S} \setminus \left(\bigcup_{i \in \mathbb{N}} Y_i\right)$$

is a minimal hitting set of $\mathcal{X}$. First observe that $\mathcal{S}'$ is indeed a hitting set, since in (3.4) the set Y_n we remove is constrained accordingly.

Now assume $\mathcal{S}'$ is not minimal. Hence, there is an element $x_m \in X$ such that $\mathcal{S}' \setminus \{x_m\}$ is a hitting set of $\mathcal{X}$ as well. Let X_n be such that $x_m \in X_n$. We see that this contradicts the construction of Y_n. More precisely, since $x_m \in \mathcal{S}'$ we have $x_m \notin \bigcup_{i=1}^{n-1} Y_i$ and hence,

$$Y_n \cup \{x_m\} \subseteq \left(\mathcal{S} \setminus \bigcup_{i=1}^{n-1} Y_i\right) \cap X_n$$

holds. Clearly,

$$\mathcal{S} \setminus \left(\bigcup_{i=1}^{n-1} Y_i \cup Y_n \cup \{x_m\}\right)$$

is a hitting set of $\mathcal{X}$and thus,

$$Y_m \notin \max\left\{Y \subseteq \left(\mathcal{S} \setminus \bigcup_{i=1}^{n-1} Y_i\right) \cap X_n \mid \mathcal{S} \setminus \left(\bigcup_{i=1}^{n-1} Y_i \cup Y\right) \text{ is a hitting set of } \mathcal{X}\right\},$$

which is a contradiction. □

Equipped with this lemma, we see that propositional logic satisfies the minimal inconsistency property.

Theorem 3.4.13. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base of the propositional logic L_P. Then $\mathcal{K}$ satisfies both the minimal inconsistency as well as the maximal consistency property.*

Proof. Let $\mathcal{H} \subseteq \mathcal{K}$ be strongly inconsistent. Since propositional logic is monotonic, this simply means that $\mathcal{H}$ is inconsistent. Due to the compactness theorem, $\mathcal{H}$ possesses a finite inconsistent subset $\mathcal{H}' \subseteq \mathcal{H}$. This implies that there is a minimal inconsistent subset $\mathcal{H}'' \subseteq \mathcal{H}'$. This is the first item of the minimal inconsistency property. Now consider $SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$. Since this a collection of finite sets of a countable set, we apply Lemma 3.4.12 in order to obtain the second item of the minimal inconsistency property. Due to Theorem 3.4.10, the maximal consistency property is satisfied as well. □

The reader may observe that the proof of Theorem 3.4.13 does not rely on specific properties of propositional logic, but only on the structural properties that can be inferred from the compactness theorem. So the following is a sufficient condition for a knowledge base to satisfy both the minimal inconsistency as well as the maximal consistency property.

Definition 3.4.14 (The compactness property)**.** An inconsistent infinite knowledge base $\mathcal{K}$ satisfies the *compactness property* if

- a subset $\mathcal{H} \subseteq \mathcal{K}$ is strongly inconsistent iff it possesses a finite subset $\mathcal{H}' \subseteq \mathcal{H}$ with $\mathcal{H}' \in SI(\mathcal{K})$.

The name "compactness property" is motivated by the fact that logics where every knowledge base satisfies the above property are called *compact*. We now have:

Theorem 3.4.15. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base which satisfies the compactness property. Then $\mathcal{K}$ satisfies both the minimal inconsistency as well as the maximal consistency property.*

The proof is similar to the proof of Theorem 3.4.13. Instead of applying the compactness theorem, we require $\mathcal{K}$ to possess the compactness property and hence, we infer the result analogously.

In a similar fashion we see that the following dual property also implies both the minimal inconsistency as well as the maximal consistency property.

Definition 3.4.16 (The co-compactness property)**.** An inconsistent infinite knowledge base $\mathcal{K}$ satisfies the *co-compactness property* if

- for each consistent $\mathcal{H} \subseteq \mathcal{K}$ there is a superset $\mathcal{H} \subseteq \mathcal{H}'$ with $\mathcal{H}' \in C_{max}(\mathcal{K})$ such that $\mathcal{K} \setminus \mathcal{H}'$ is finite.

Now we may proceed as in the above theorem. Given a consistent $\mathcal{H} \subseteq \mathcal{K}$, we can move to a maximal one $\mathcal{H}'$ (first item of the maximal consistency property). Moreover, $\mathcal{K} \setminus \mathcal{H}'$ is finite for each $\mathcal{H}' \in C_{max}(\mathcal{K})$, and hence we can apply Lemma 3.4.12 which yields the second item of the maximal consistency property. Thus:

Theorem 3.4.17. *Let $\mathcal{K}$ be an inconsistent infinite knowledge base which satisfies the co-compactness property. Then, $\mathcal{K}$ satisfies both the minimal inconsistency as well as the maximal consistency property.*

We want to mention that in general, properties like the *minimal inconsistency property* or the *compactness property* are rare, especially when considering non-monotonic logics. This becomes apparent in view of Example 3.4.2: First, this example works for nearly all relevant semantics for AFs in the sense that there is no non-empty extension (see Chapter 5 below). Second, it is straightforward to construct similar examples for ASP or other non-monotonic frameworks like Reiter's default logic [95] via infinite chains of default-negated atoms. However, besides being interesting from a theoretical point of view, the results of this section can be applied to certain classes of knowledge bases. After all, the *compactness property* is inspired by the behavior of propositional logic and so does not appear unnatural. The compactness theorem is also true for e.g. first order or modal logic. Moreover, in Chapter 5 we will see how to apply our results to so-called *finitary* AFs.

3.5 Conclusion and Related Work

In this chapter we studied the relation between inconsistent and consistent subsets of a knowledge base in an abstract setting covering arbitrary logics, including non-monotonic ones. We showed that in the general case, the standard notion of inconsistency does not play the same role as in monotonic reasoning. One of our main results shows that Reiter's duality characterization of maximal consistent subsets of a given knowledge base generalizes to non-monotonic logics. This was achieved by strong inconsistency. We extended our investigation to repairs based on adding formulas to a knowledge base, which is inherently meaningless in the monotonic case and thus a novel approach compared to Reiter's setting. We applied our results to handle situations where formulas are to be modified instead of deleted or added. In general, the results we established demonstrate the following properties of inconsistent subsets in non-monotonic logics:

- In order to obtain the desired duality characterizations, we need to impose monotonic behavior on the inconsistent sets.
- The requirement that all sets within a certain range are supposed to be inconsistent facilitates similar duality results for different other settings:
  - Allowing for adding information to a knowledge base as well; here, the inconsistency notions are $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ and $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$.
  - Modifying formulas; here, the inconsistency notion requires inconsistency for all modifications within a certain range.
- Refining consistency to strong consistency yields analogous duality results. This shows that the "direction" in which we impose monotony on the subsets of $\mathcal{K}$ does not seem to matter from a structural point of view.
- Moving to infinite knowledge bases does not cause any harm besides the usual considerations about the existence of certain sets (in our case minimal resp. maximal sets with desired properties).

We believe that the results of this chapter are not only interesting regarding generalizations of Reiter's hitting set duality, but also point out quite encouraging structural properties of knowledge bases.

In his seminal paper [96], Reiter establishes his duality result within a setting consisting of a system description SD and components $Comp$. Our results allow us thus to capture system descriptions expressed in non-monotonic logics, e.g., [53]. The paper [96] is also concerned about computing hitting sets. In fact, many algorithms and systems for enumerating minimal inconsistent sets –see [8; 78; 79]– build on the duality between minimal inconsistent sets, maximal consistent sets, and their respective hitting sets. For example, [79] takes turns in computing minimal unsatisfiable sets and maximal consistent sets and uses the duality between the two to compute remaining sets of either type. Hitting sets are also utilized in computation of causes and responsibilities of inconsistency in databases [98].

The motivation of [96] for establishing the duality result is *diagnosis*, i. e., the problem of checking whether a system (which is usually formalized in propositional or first-order logic) is faulty and determining the causes for that. Many further approaches consider this problem as well and apply and extend the work of [96]. To name just a few, [100] exploits the duality between minimal inconsistent sets and maximal consistent sets and, similarly as discussed above for the task of enumerating these sets, interleaves construction of these two sets with each other, in order to solve the diagnosis problem. The work [85] casts the problem into SAT and reports on to-date significant performance improvements. Similarly, Marques-Silva et al. (see [83]) solve the diagnosis problem by casting it into a MaxSAT problem and leveraging SAT solvers.

We are not aware of any work extending the well known hitting set duality to non-monotonic logics as we did in this chapter. The concept of maximal consistency in non-monotonic logics, however, is not novel (see for instance [97, Definition 5.3]). Extensions of consistency removal to non-monotonic logics can also be found in the literature. An example in autoepistemic logic has been analyzed in [70]. The closest to our work is probably [55]. Thomas Eiter and colleagues have studied ways of restoring consistency in multi-context systems [35]. They focus on the case where the source of inconsistency can be attributed to the bridge rules of a multi-context system. The paper [56] considers the problem of belief revision in ASP and analyzes, similar to the present chapter, a setting where restoring consistency may be obtained due to additional rules.

Chapter 4

Measuring Inconsistency

So far, we have viewed inconsistency as a binary concept - a knowledge base or a subset of it is either inconsistent or not. Although this is true without any doubt, one might ask how severe conflicts of a given knowledge base are. For example, an agent could be given some information which yields the choice between several inconsistent conclusions he or she might draw. The agent is thus interested in a comparison between the conflicts, i. e., which conclusion represents the “most insignificant” contradictions? This motivates techniques to quantitatively assess the inconsistency of a given knowledge base rather than just deciding whether it is consistent or not.

In order to achieve this goal, so-called *inconsistency measures* have been proposed in the literature, see for instance [60; 66; 107]. Inconsistency measures are functions that aim at comparing the severity of the conflicts in propositional knowledge bases. The basic intuition behind such a measure $\mathcal{I}$ is that large values of $\mathcal{K}$ correspond to more severe inconsistencies.

To illustrate possible applications of inconsistency measures as well as the necessity to take non-monotonic reasoning into account, let us consider the following example.

Example 4.0.1. Assume there is a trial and the judges need to determine who is responsible for a burglary. There is a suspect A who claims he was at home at that time, in particular he was not at the crime scene. We can model this testimony as a simple fact, i. e., we have

$$\neg atCrimeScene(A).$$

However, there is a witness W_1 stating the opposite. Any witness is credible by default.

$$atCrimeScene(X) \leftarrow witness(Y, X), \text{not } \neg credible(Y). \qquad witness(W_1, A).$$

Moreover, the police found the suspect’s fingerprints, which strongly suggests his presence at the crime scene.

$$atCrimeScene(X) \leftarrow fingerprints(X). \qquad fingerprints(A).$$

There is a second suspect B. He was also seen at the crime scene, even by two witnesses W_2 and W_3, but also denies his presence.

$$\neg atCrimeScene(B). \qquad witness(W_2, B). \qquad witness(W_3, B).$$

However, witness W_3 was drunk at that time, so his testimony is to be taken with a grain of salt.

$$\neg credible(W_3).$$

After collecting all the information, the judges face the following contradictions: For A

$$\neg atCrimeScene(A).$$
$$witness(W_1, A).$$
$$atCrimeScene(A) \leftarrow witness(W_1, A), \text{not } \neg credible(W_1).$$
$$fingerprints(A).$$
$$atCrimeScene(A) \leftarrow fingerprints(A).$$

and for B

$$\neg atCrimeScene(B).$$
$$witness(W_2, B).$$
$$atCrimeScene(B) \leftarrow witness(W_2, B), \text{not } \neg credible(W_2).$$

The judges are quite sure that A was the burglar, because A is involved in more conflicts; but without an inconsistency measure they cannot formally justify this intuition.

A simple but popular approach to measure inconsistency is to take the number of minimal inconsistent subsets [67], i. e., to define $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |I_{min}(\mathcal{K})|$. The measure $\mathcal{I}_{\mathsf{MI}}$ already complies with many basic ideas of inconsistency measurement, for example it satisfies *consistency*, i. e., $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = 0$ iff $\mathcal{K}$ is consistent. *Consistency* is one of the most basic properties an inconsistency measure $\mathcal{I}$ should have since if formalizes that $\mathcal{I}$ is capable of distinguishing between consistency and inconsistency. By also taking the size and the relationships of minimal inconsistent subsets into account, a wide variety of different inconsistency measures can be defined by refining $\mathcal{I}_{\mathsf{MI}}$, see [67; 71; 73].

Measuring inconsistency in non-monotonic logics possesses additional challenges which is already apparent when considering the measure $\mathcal{I}_{\mathsf{MI}}(\mathcal{K})$ from above. If $\mathcal{K}$ is a non-monotonic knowledge base, $\mathcal{I}_{\mathsf{MI}}$ is not as meaningful as a knowledge base $\mathcal{K}$ may contain minimal inconsistent subsets even though it is consistent. As the reader may already expect, our notion of strong inconsistency will play a central role in order to overcome this issue.

Research in inconsistency measurement is driven by *rationality postulates*, i. e., desirable properties that should hold for concrete approaches. There is a growing number of rationality postulates for inconsistency measurement but not every postulate is generally accepted, see [27] for a discussion on this topic. The issue of measuring inconsistency in non-monotonic frameworks requires some reconsideration compared to the propositional setting. This becomes apparent when considering the *monotony* postulate which is usually satisfied by inconsistency measures and demands $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$ whenever $\mathcal{K} \subseteq \mathcal{K}'$ holds, i. e., the severity of inconsistency cannot be decreased by adding new information. This rationality postulate is motivated by the observation that novel information cannot resolve conflicts. As already discussed, this is not the case anymore when investigating non-monotonic frameworks. It is thus possible that $\mathcal{K}$ is inconsistent, while $\mathcal{K}'$ is not, so we would expect $\mathcal{I}(\mathcal{K}') < \mathcal{I}(\mathcal{K})$ for any reasonable measure $\mathcal{I}$ in this case.

The goal of this chapter is to contribute to a thorough understanding of inconsistency by extending our investigation with this quantitative approach to analyze inconsistency. We want to consider generalized versions of three measures based on minimal inconsistent sets. In order to assess their behavior, we will develop rationality postulates extending established ones from the literature. Some of the postulates still make sense for a general, possibly non-monotonic logic, but most of them require refinements. We then analyze the measures with respect to the postulates. We will also discuss the severity of inconsistency of a subset $\mathcal{H}$ of a knowledge base $\mathcal{K}$, i. e., how significant are conflicts within $\mathcal{H}$ considering it is part of a knowledge base $\mathcal{K}$? The end of this chapter is devoted to considerations which are specific for the concrete framework ASP. This allows for more concrete approaches and demonstrates how to tailor measures and postulates for particular logics.

4.1 Measures for Non-monotonic Logics

We now introduce the inconsistency measures we are going to consider throughout most of this chapter. Assume an arbitrary but fixed logic L. In classical inconsistency measurement, minimal inconsistent subsets of a knowledge base play an important role since they can be seen as the "atomic conflicts" within $\mathcal{K}$. A rather simple but still popular approach to measure inconsistency is thus taking the value $|I_{min}(\mathcal{K})|$. The notion of strong inconsistency facilitates the following generalization of this measure to non-monotonic logics.

Definition 4.1.1. Define $\mathcal{I}_{\mathsf{MSI}} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0}$ via $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) = |SI_{min}(\mathcal{K})|$.

One drawback of this approach is that the size of a set $\mathcal{H} \in SI_{min}(\mathcal{K})$ is not taken into account. Usually, a minimal inconsistent subsets is considered more severe the smaller it is, i. e., the fewer formulas are required in order to yield a contradiction. A famous example to illustrate this is the so-called *lottery paradox*:

Example 4.1.2. Assume there is a lottery with n tickets. Consider atoms $t_1, \ldots, t_n$ with the intuitive meaning that t_i is true iff the i-th ticket wins. Assume the lottery is fair and exactly one ticket wins. We thus have $t_1 \dot{\vee} \ldots \dot{\vee} t_n$. However, considering an individual ticket t_i it appears reasonable to assume that it loses, so we have $\neg t_1, \ldots, \neg t_n$. We thus obtain the inconsistent knowledge base $\mathcal{K}_n = \{t_1 \dot{\vee} \ldots \dot{\vee} t_n, \neg t_1, \ldots, \neg t_n\}$. Now consider a lottery where n is quite small, e.g., $n = 1$. In this case, the assumption that t_1 loses is not as reasonable anymore given that at least one ticket wins. However, the bigger n is, the more reasonable this assumption becomes, e.g., $n = 10^6$ yields a negligible chance for each t_i to win. Hence, even though both $\mathcal{K}_1$ and $\mathcal{K}_{10^6}$ are inconsistent, the latter appears quite reasonable while the former is hard to take seriously.

So commonly, the bigger a minimal inconsistent set is, the less severe the conflict is viewed. This is obviously ignored by $\mathcal{I}_{\mathsf{MSI}}$. For example, $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}_n) = 1$ for any n for the knowledge base $\mathcal{K}_n$ from the lottery paradox. In [67] a measure is proposed taking this into account. Making use of strong inconsistency, we obtain the following measure:

Definition 4.1.3. Define $\mathcal{I}_{\mathsf{MSI^C}} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0}$ via $\mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K}) = \sum_{\mathcal{H} \in SI_{min}(\mathcal{K})} \frac{1}{|\mathcal{H}|}$.

Instead of counting the number of sets in $SI_{min}(\mathcal{K})$, one could also consider the amount of *formulas* which are considered problematic. Based on a measure in [61], we have the following, quite simple approach:

Definition 4.1.4. Define $\mathcal{I}_{\mathsf{p}} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0}$ via $\mathcal{I}_{\mathsf{p}}(\mathcal{K}) = |\bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}|$.

Note that there are further measures based on minimal inconsistent sets, see [67; 71; 73]. An investigation of generalizations of those is left for future work.

Example 4.1.5. Consider our running examples from the previous chapter, i. e., the propositional knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$ with

$$SI_{min}(\mathcal{K}) = \{\{a,\ a \to b,\ \neg b\},\ \{c,\ \neg c\}\},$$

the logic program

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

with

$$SI_{min}(P) = \{\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}\}$$

and the argumentation framework over $A = \{a, b, c\}$ represented by the knowledge base $R = \{(a, b), (b, c), (c, c)\}$ with

$$SI_{min}(R) = \{\{(a, b), (c, c)\}\}.$$

The inconsistency measures from above assign the following values:

	$\mathcal{I}_{\mathsf{MSI}}$	$\mathcal{I}_{\mathsf{MSI^C}}$	$\mathcal{I}_{\mathsf{p}}$
$\mathcal{K}$	2	5/6	5
P	1	1/3	3
R	1	1/2	2

We observe that $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ differ for P and R even though both possess one minimal strongly inconsistent subset.

Now assume a situation where we are given some inconsistent knowledge bases $\mathcal{K}_1, \ldots, \mathcal{K}_n$ and need to assess the severity of their conflicts. To achieve this goal, it seems reasonable to choose an appropriate inconsistency measure $\mathcal{I}$, compare the values $\mathcal{I}(\mathcal{K}_1), \ldots, \mathcal{I}(\mathcal{K}_n)$ and then prefer the knowledge base $\mathcal{K}_i$ with the lowest inconsistency degree:

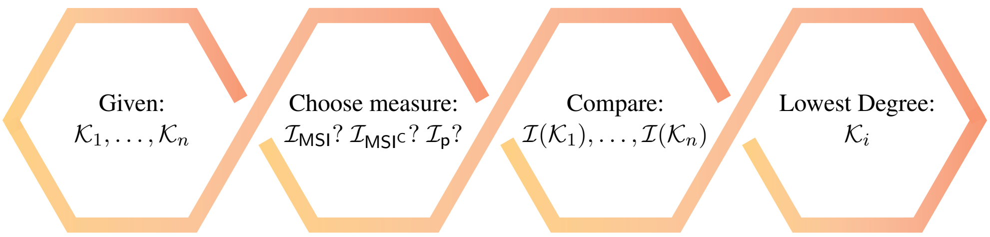


Figure 4.1: Finding "the best" knowledge base via inconsistency measures

Considering this approach we observe that the crucial point is finding the suitable inconsistency measure. In order to investigate and compare the behavior of inconsistency measures, researchers have thus proposed various rationality postulates (see for instance Thimm:2016e). We want to develop similar tools for inconsistency measurement in non-monotonic logics.

4.2 Rationality Postulates for General Logics

We are going to revisit rationality postulates for inconsistency measures from the literature and phrase them within the context of an arbitrary, possibly non-monotonic, logic. We will start by considering the four postulates that a *basic inconsistency measure* should have due to [68]. We will then continue our investigation with a collection of other postulates that can be lifted to our general setting.If not stated otherwise, we assume an arbitrary but fixed logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ and an inconsistency measure $\mathcal{I} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0}$ for the remainder of this section.

4.2.1 Basic Postulates

The most basic (and undisputed) property that an inconsistency measure should have is the ability to distinguish between consistency and inconsistency, i. e., $\mathcal{I}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent. Undoubtedly this makes sense in non-monotonic frameworks as well.

Consistency For any knowledge base $\mathcal{K} \subseteq \mathcal{WF}$, $\mathcal{I}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent.

In contrast to non-monotonic logics, the *consistency* postulates implies $\mathcal{I}(\mathcal{H}) = 0$ for each $\mathcal{H} \subseteq \mathcal{K}$ if the underlying logic is monotonic. This follows immediately from Lemma 2.5.18.

Proposition 4.2.1. *If L is monotonic, $\mathcal{K}$ is consistent and $\mathcal{I}$ satisfies consistency, then we also have $\mathcal{I}(\mathcal{H}) = 0$ for each $\mathcal{H} \subseteq \mathcal{K}$.*

Since we do not see any reason to doubt the *consistency* postulate for our setting, let us continue with the second one, which is *monotony*. *Monotony* is a fairly accepted postulate which formalizes the intuition that moving from a *monotonic* knowledge base to a superset should not decrease the inconsistency degree since adding information to a knowledge base does not resolve conflicts.

Monotony If $\mathcal{K}$ and $\mathcal{K}'$ are knowledge bases, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

In non-monotonic frameworks, *monotony* does not make sense because additional information might resolve some conflicts or even render $\mathcal{K}$ consistent. We should thus only expect a monotonic behavior if $\mathcal{K}'$ does not resolve conflicts occurring in $\mathcal{K}$. More precisely, if $\mathcal{H} \subseteq \mathcal{K}$ is strongly inconsistent, i. e., $\mathcal{H} \in SI(\mathcal{K})$, then there should be no subset $\mathcal{H}' \subseteq \mathcal{K}'$ such that $\mathcal{H} \cup \mathcal{H}'$ is consistent. Otherwise, if $\mathcal{H} \cup \mathcal{H}'$ is consistent, then it is unclear whether $\mathcal{H}$ even contributes to inconsistency of $\mathcal{K} \cup \mathcal{K}'$, which makes a comparison between $\mathcal{I}(\mathcal{K})$ and $\mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ questionable. To illustrate this, consider the following example:

Example 4.2.2. Recall our logic program

$$
\begin{array}{lll}
P: & a \vee b. & a \leftarrow b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b.
\end{array}
$$

Consider $P' = \{b.,\ d.,\ \neg d.\}$ containing "$b.$", which resolves the conflict within P: The subprogram $H \in SI_{min}(P)$ with $H = \{c \leftarrow \text{not } b., \neg c \leftarrow \text{not } b., a \leftarrow b.\}$ is not strongly $(P \cup P')$-inconsistent due to the consistent superprogram $H \cup H'$ with $H' = \{b.\} \subseteq P'$:

$$
H \cup H' : \quad c \leftarrow \text{not } b. \quad \neg c \leftarrow \text{not } b. \quad a \leftarrow b. \quad b.
$$

In particular, $P \cup \{b.\}$ is consistent as well, but P' involves the conflict "$d.$" vs. "$\neg d.$". We have $SI_{min}(P \cup P') = \{\{d., \neg d.\}\}$ which only represents the conflict within P'. So the comparison between inconsistency of P and $P \cup P'$ does not seem to make sense.

We thus need to adjust the *monotony* postulate in order to obtain a meaningful one for non-monotonic frameworks. For this, recall the first time we faced a similar issue, namely when considering strong inconsistency as a refinement of the ordinary counterpart in monotonic logics. The idea was to turn the following property into an axiom:

- If $\mathcal{H} \subseteq \mathcal{K}$ is inconsistent, then the same is true for each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$.

This yields the following property for all knowledge bases $\mathcal{K}$ and $\mathcal{K}'$ which is significant regarding the *monotony* postulate:

- If $\mathcal{H} \subseteq \mathcal{K}$ is inconsistent, and $\mathcal{H}' \subseteq \mathcal{K}'$, then $\mathcal{H} \cup \mathcal{H}'$ is inconsistent as well.

The above property formally states that no subset of $\mathcal{K}'$ is capable of resolving any conflict within $\mathcal{K}$. Given a non-monotonic logic, this is not necessarily the case anymore. However, we can restrict the postulate to cases where $\mathcal{K}'$ satisfies this property. As inconsistent subsets do not play a central role in non-monotonic reasoning (see Chapter 3 above), we may replace "inconsistent" with "strongly inconsistent", which yields:

- If $\mathcal{H} \subseteq \mathcal{K}$ is strongly $\mathcal{K}$-inconsistent, and $\mathcal{H}' \subseteq \mathcal{K}'$, then $\mathcal{H} \cup \mathcal{H}'$ is strongly $(\mathcal{K} \cup \mathcal{K}')$-inconsistent.

It is now easy to see that restricting to $\mathcal{H}' = \emptyset$ suffices. More precisely, the above property is equivalent to the following one:

- If $\mathcal{H} \subseteq \mathcal{K}$ is strongly $\mathcal{K}$-inconsistent, then $\mathcal{H}$ is strongly $(\mathcal{K} \cup \mathcal{K}')$-inconsistent as well.

If this is the case, then moving from $\mathcal{K}$ to $\mathcal{K} \cup \mathcal{K}'$ is comparable to the same situation in a monotonic logic. This motivates the following:

Definition 4.2.3. Let $\mathcal{K}$ and $\mathcal{K}'$ be knowledge bases. We say that $\mathcal{K}'$ *preserves conflicts of* $\mathcal{K}$ if $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{K}')$ for any $\mathcal{H} \in SI(\mathcal{K})$.

We observe that this property transfers to *minimal* strongly inconsistent subsets as well:

Proposition 4.2.4. *If $\mathcal{K}$ and $\mathcal{K}'$ are knowledge bases and $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$, then $SI_{min}(\mathcal{K}) \subseteq SI_{min}(\mathcal{K} \cup \mathcal{K}')$.*

Proof. Let $\mathcal{H} \in SI_{min}(\mathcal{K})$. Since $\mathcal{K}'$ preserves conflicts, $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{K}')$. Now assume there is a set $\mathcal{H}' \subsetneq \mathcal{H}$ with $\mathcal{H}' \in SI(\mathcal{K} \cup \mathcal{K}')$. Since $\mathcal{H}' \subseteq \mathcal{H}$, we have $\mathcal{H}' \in SI(\mathcal{K})$, yielding a contradiction as $\mathcal{H}$ was assumed to be minimal. Hence $\mathcal{H} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$. □

Let us now consider our running examples again to see the definition at work. We first start with the monotonic propositional knowledge base from before. As the reader may already expect, monotony renders the property of preserving conflicts trivial.

Example 4.2.5. Recall $\mathcal{K} = \{a, a \to b, \neg b, c, \neg c\}$. Any propositional knowledge base $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$ due to monotony of the logic, so we have $\{c, \neg c\} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$.

More generally, the following statement can be easily inferred from Lemma 2.5.18.

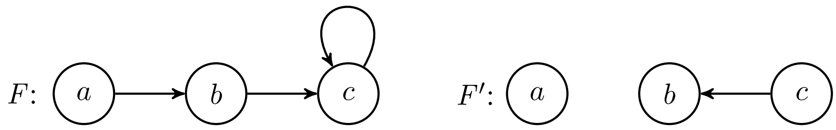
Proposition 4.2.6. *Let L be monotonic and $\mathcal{K}$ and $\mathcal{K}'$ be two knowledge bases of L. Then $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$.*

Example 4.2.7. Recall the logic program

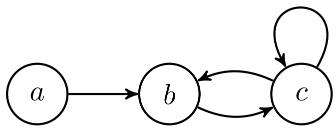
$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

The program $P' = \{b.,\ d.,\ \neg d.\}$ from Example 4.2.2 does not preserve conflicts of P as we already noted above. More precisely, we saw that $H = \{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}$ is not strongly $(P \cup P')$-inconsistent since P' contains the fact "$b.$".

Example 4.2.8. Now consider the running example AF $F = (A, R)$ with $A = \{a, b, c\}$ represented by the knowledge base $R = \{(a, b), (b, c), (c, c)\}$. A quite simple choice of a knowledge base R' which preserves conflicts of R is $R' = \{(c, b)\}$, which induces the AF $F' = (A, R')$:



We see that $SI_{min}(R \cup R') = \{\{(a, b),\ (c, c)\}\} = SI_{min}(R)$. The AF represented by $R \cup R'$ is the following:



The notion of preserving conflicts motivates our postulate *strong monotony*, a non-monotonic counterpart to *monotony* from above. The difference is that we require $\mathcal{K}'$ not not resolve conflicts within $\mathcal{K}$. This restriction is not completely novel, since it is inherently given for monotonic logics as we discussed above.

Strong Monotony If $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

It should nevertheless be clear that *strong monotony* is a weaker postulate than *monotony* due to the additional premise. After all, resolving conflicts via adding information is an important feature for non-monotonic logics which is excluded for the postulate.

We now turn to the *free formula independence* postulate [68]. Intuitively, a formula α of a knowledge base $\mathcal{K}$ is *free* in $\mathcal{K}$ if it does not cause any conflicts. A free formula should thus not be "blamed" for inconsistency of $\mathcal{K}$ and hence not change the inconsistency degree, which is formalized by the postulate *free formula independence*. We consider two ways to generalize this notion to non-monotonic logics as we believe both to be reasonable.

Consider a monotonic logic. Formally, a free formula $\alpha \in \mathcal{K}$ is one that does not occur in a minimal inconsistent subset $\mathcal{H} \in SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$.

Definition 4.2.9. Let $\mathcal{K}$ be a monotonic knowledge base. A formula $\alpha \in \mathcal{K}$ is called *free* if

$$\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H}. \tag{4.1}$$

Denote by $\mathit{Free}(\mathcal{K})$ the set of all free formulas of $\mathcal{K}$.

This notion matches the intuition that α is not responsible for any conflict within $\mathcal{K}$. Whenever there is an inconsistent subset $\mathcal{H} \subseteq \mathcal{K}$, then $\mathcal{H} \setminus \{\alpha\}$ is inconsistent as well, so α does not cause any harm.

Example 4.2.10. Recall that the propositional knowledge base $\mathcal{K} = \{a,\, a \to b,\, \neg b,\, c,\, \neg c\}$ possesses $I_{min}(\mathcal{K}) = \{\{a,\, a \to b,\, \neg b\},\, \{c,\, \neg c\}\}$. Hence it does not contain any free formula since every formula occurs in a minimal inconsistent subset.

In order to generalize this notion, let us first take a look at an alternative way to define free formulas. Assume we are given a monotonic logic. Recall Reiter's hitting set duality, i. e., $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$ iff $\mathcal{S}$ is a hitting set of $I_{min}(\mathcal{K})$. This strong connection between minimal inconsistent and maximal consistent subsets facilitates a definition of free formulas via the maximal consistent subsets of $\mathcal{K}$. Indeed,

$$\mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H} \tag{4.2}$$

holds and hence, a formula α is free iff it occurs in every maximal consistent subset of $\mathcal{K}$. We note that the intuition is the same: Since α can be added to any subset $\mathcal{H} \subseteq \mathcal{K}$ without introducing inconsistency, it occurs in any $\mathcal{H} \in C_{max}(\mathcal{K})$. For the purpose of our generalization to non-monotonic logics, we observe that (4.2) is a corollary of Reiter's hitting set duality which is generalized to non-monotonic logics in Theorem 3.1.12. We thus expect a similar result when replacing $I_{min}(\mathcal{K})$ with $SI_{min}(\mathcal{K})$. Indeed:

Corollary 4.2.11. *Let $\mathcal{K}$ be a knowledge base. Then*

$$\mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}.$$

Proof. "$\subseteq$": Let $\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}$. Hence, α does not occur in any minimal hitting set of $SI_{min}(\mathcal{K})$ and thus, due to Theorem 3.1.12, it occurs in all maximal consistent sets $\mathcal{H} \in C_{max}(\mathcal{K})$.
"$\supseteq$": Now assume $\alpha \notin \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}$, i. e., $\alpha \in \mathcal{H}$ for a minimal strongly inconsistent set $\mathcal{H} \in SI_{min}(\mathcal{K})$. Hence, $\mathcal{H} \setminus \{\alpha\} \notin SI_{min}(\mathcal{K})$ and thus, there is a maximal consistent set $\mathcal{H}'$ with $\mathcal{H} \setminus \{\alpha\} \subseteq \mathcal{H}'$. This means $\alpha \notin \mathcal{H}'$ because otherwise, $\mathcal{H}'$ would contain a strongly inconsistent set. It follows that $\alpha \notin \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}$. □

Corollary 4.2.11 suggests a very natural and smooth generalization of free formulas, which lifts the intuitive as well as the formal meaning with respect to both aspects: A free formula α does not introduce conflicts ("$\alpha \notin \mathcal{H}$ for each $\mathcal{H} \in SI_{min}(\mathcal{K})$") where we note that the relevant conflicts of a non-monotonic knowledge base are the minimal strongly inconsistent subsets; and α can be added to any subset of $\mathcal{K}$ without introducing inconsistency ("$\alpha \in \mathcal{H}$ for each $\mathcal{H} \in C_{max}(\mathcal{K})$"). So we define:

Definition 4.2.12. Let $\mathcal{K}$ be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *free wrt. strong inconsistency* (or *SI-free* or simply *free* if there is no risk of confusion) if

$$\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}.$$

Denote by $\mathit{Free}_{SI}(\mathcal{K})$ the set of all SI-free formulas of $\mathcal{K}$.

Example 4.2.13. Consider our program P again:

$$P: \qquad a \vee b. \qquad a \leftarrow b.$$
$$c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

As already discussed, $SI_{min}(P) = \{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}$. We thus obtain $Free_{SI}(P) = \{a \vee b.\}$. To see Corollary 4.2.11at work we recall

$$\begin{aligned} C_{max}(P) = \{&\{a \vee b.,\ a \leftarrow b.,\ c \leftarrow \text{not } b.\},\\ &\{a \vee b.,\ a \leftarrow b.,\ \neg c \leftarrow \text{not } b.\},\\ &\{a \vee b.,\ c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}\}. \end{aligned}$$

Since "$a \vee b.$" is the only formula occurring in all maximal consistent sets, we obtain $\bigcap_{H \in C_{max}(P)} H = \{a \vee b.\}$.

Example 4.2.14. For the AF represented by $R = \{(a,b),(b,c),(c,c)\}$ we obtain the set $Free_{SI}(R) = \{(b,c)\}$.

As expected, $Free_{SI}(\mathcal{K})$ generalizes $Free(\mathcal{K})$ to non-monotonic logics in the sense that they coincide in the monotonic case.

Proposition 4.2.15. *Let $\mathcal{K}$ be a monotonic knowledge base. Then, $Free(\mathcal{K}) = Free_{SI}(\mathcal{K})$.*

Proof. Due to Proposition 3.1.11, Item 2, we have $I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$. In particular,

$$Free(\mathcal{K}) = \mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H} = \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = Free_{SI}(\mathcal{K}),$$

which proves our claim. □

Finally, let us mention that $Free_{SI}(\mathcal{K})$ can also be defined without explicitly mentioning minimality

Proposition 4.2.16. *Let $\mathcal{K}$ be a knowledge base. If $\alpha \in \mathcal{K}$, then $\alpha \in Free_{SI}(\mathcal{K})$ iff*

$$\forall \mathcal{H} \subseteq \mathcal{K}:\ \mathcal{H} \notin SI(\mathcal{K}) \Rightarrow \mathcal{H} \cup \{\alpha\} \notin SI(\mathcal{K}). \tag{4.3}$$

Proof. The implication "$\Leftarrow$" is trivial, so we show "$\Rightarrow$": Assume (4.3) is wrong, i. e., there is a set $\mathcal{H} \notin SI(\mathcal{K})$ with $\mathcal{H} \cup \{\alpha\} \in SI(\mathcal{K})$. Then $\mathcal{H} \cup \{\alpha\}$ contains a minimal strongly inconsistent set $\mathcal{H}'$. Observe that $\alpha \in \mathcal{H}'$, because otherwise, $\mathcal{H}'$ has a consistent superset as it is the case for $\mathcal{H}$. Since $\mathcal{H}'$ is minimal, it holds that $\mathcal{H}' \setminus \{\alpha\} \notin SI_{min}(\mathcal{K})$, but $\mathcal{H}' \in SI_{min}(\mathcal{K})$ and thus, $\alpha \notin Free_{SI}(\mathcal{K})$. □

The similarities between $Free(\mathcal{K})$ and $Free_{SI}(\mathcal{K})$ motivate a rationality postulate similar in spirit to *free formula independence*, which requires $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$ for $\alpha \in Free(\mathcal{K})$. Analogously, one might expect an SI-free formula $\alpha \in \mathcal{K}$ not to increase the inconsistency degree of $\mathcal{K}$ since it does not introduce strongly inconsistent subsets. It could still resolve conflicts, motivating the following postulates, similar to *free formula dilution* from [87]:

SI-Free If $\alpha \in Free_{SI}(\mathcal{K})$, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$.

However, SI-free formulas are not as well-behaving as free formulas in monotonic logics. The issue we need to take into account is that ordinary inconsistency of a subset $\mathcal{H} \subseteq \mathcal{K}$ merely depends on $\mathcal{H}$, where strong inconsistency is a property $\mathcal{H}$ has with respect to the whole knowledge base $\mathcal{K}$. Removing an SI-free formula $\alpha \in \mathcal{K}$ might thus change the structure of $SI_{min}(\mathcal{K})$ in an unexpected way. To see this, let us consider the following example.

Example 4.2.17. Let P be the following logic program:

$$P: \ a \leftarrow \text{not } a,\, b. \qquad a \leftarrow \text{not } c. \qquad d \leftarrow \text{not } d. \qquad b. \quad c. \quad d.$$

The reader may verify that

$$SI_{min}(P) = \{\{a \leftarrow \text{not } a,\, b.,\ b.,\ c.\}\}.$$

In particular,

$$r_1 = d.$$
$$r_2 = a \leftarrow \text{not } c.$$

are in $Free_{SI}(P)$. However, removal of r_1 turns $\{d \leftarrow \text{not } d.\}$ into a strongly inconsistent subset, while removal of r_2 renders "c." irrelevant regarding the conflicts of P, so

$$SI_{min}(P \setminus \{r_1, r_2\}) = \{\{a \leftarrow \text{not } a,\, b.,\ b.\},\ \{d \leftarrow \text{not } d.\}\}.$$

We make the following observations:

- the conflict "$d \leftarrow \text{not } d$." suddenly occurs,
- the conflict $\{a \leftarrow \text{not } a,\, b.,\ b.,\ c.\}$ does not rely on "c." anymore since the option to infer a is removed,
- the number of minimal conflicts increased.

It is thus hard to predict what happens when an SI-free formula is removed from a given knowledge base. In particular, we do not have $SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\})$ for each $\alpha \in Free_{SI}(\mathcal{K})$ which means that not even $\mathcal{I}_{\mathsf{MSI}}$ –a measure based on minimal strongly inconsistent subsets– satisfies the *SI-free* postulate (cf. Section 4.3 below). Another observation regarding $Free_{SI}(\mathcal{K})$ is relevant: Although free formulas are not supposed to participate in minimal conflicts, the set $Free_{SI}(\mathcal{K})$ itself is in general not consistent.

Example 4.2.18. Consider the logic program

$$P: \qquad a. \qquad \neg a. \qquad \leftarrow \text{not } a, \text{not } \neg a.$$

We see that $SI_{min}(P) = \{\{a.,\ \neg a.\}\}$ and hence, $Free_{SI}(P) = \{\leftarrow \text{not } a, \text{not } \neg a.\}$ is an inconsistent program.

The previous considerations suggest that this notion depends heavily on the particular knowledge base. We will thus continue by introducing a stronger notion.

For this, we consider an alternative to the definition of free formulas in a way that they "do not induce strong inconsistency". Let us have a look at the monotonic case again. Since a free formula α does not induce inconsistency, one can see that α satisfies

$$\forall \mathcal{H} \subseteq \mathcal{K} : \mathcal{H} \in C(\mathcal{K}) \Rightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}). \tag{4.4}$$

In a monotonic framework, (4.4) formalizes that α is irrelevant regarding conflicts of $\mathcal{K}$ as it cannot turn a consistent set $\mathcal{H} \subseteq \mathcal{K}$ in an inconsistent one. In a non-monotonic logic, α could resolve conflicts, so we need to strengthen the condition:

$$\forall \mathcal{H} \subseteq \mathcal{K} : \mathcal{H} \in C(\mathcal{K}) \Leftrightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}). \tag{4.5}$$

This motivates the following definition.

Definition 4.2.19. Let $\mathcal{K}$ be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *neutral* if it satisfies

$$\forall \mathcal{H} \subseteq \mathcal{K} : \mathcal{H} \in C(\mathcal{K}) \Leftrightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}).$$

The neutral formulas in $\mathcal{K}$ are denoted by $Ntr(\mathcal{K})$.

It is easy to see that both notions coincide for monotonic logics.

Proposition 4.2.20. *If $\mathcal{K}$ is monotonic, then $Ntr(\mathcal{K}) = Free(\mathcal{K})$.*

Proof. This is clear due to Lemma 2.5.18. □

Note that in general, Ntr is a stronger notion than $Free_{SI}$.

Proposition 4.2.21. *If $\mathcal{K}$ is a knowledge base, then $Ntr(\mathcal{K}) \subseteq Free_{SI}(\mathcal{K})$*

Proof. Let $\alpha \in Ntr(\mathcal{K})$. Due to (4.4), it can be added to any set $\mathcal{H} \subseteq \mathcal{K}$ without introducing inconsistency. Hence, $\alpha \in \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H} = Free_{SI}(\mathcal{K})$. □

We note that in contrast to SI-free formulas, the neutral ones do not make use of strong inconsistency. Even though the hitting set duality from Theorem 3.1.12 suggests to utilize this notion, the neutral formulas are still quite well-behaving. The reason is that neutral formulas do not depend as much on the structure of the knowledge base and vice versa, do not influence $\mathcal{K}$ and in particular the structure of $SI_{min}(\mathcal{K})$.

Proposition 4.2.22. *Let $\mathcal{K}$ be a knowledge base and let $\alpha \in Ntr(\mathcal{K})$. Then,*

$$SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\}).$$

Proof. By definition of $Ntr(\mathcal{K})$ we have $\mathcal{H} \in SI(\mathcal{K})$ if and only if $\mathcal{H} \in SI(\mathcal{K} \setminus \{\alpha\})$ for any set $\mathcal{H} \subseteq \mathcal{K} \setminus \{\alpha\}$. Hence, the claim follows since no set $\mathcal{H} \in SI_{min}(\mathcal{K})$ contains α. □

Moreover, in contrast to $Free_{SI}(\mathcal{K})$, the neutral formulas always form consistent subsets of a knowledge base, as long as the empty knowledge base is considered consistent. Without proof, we state the following obvious fact:

Proposition 4.2.23. *If L is a logic such that $\emptyset$ is consistent and $\mathcal{K}$ a knowledge base of L, then $Ntr(\mathcal{K})$ is consistent.*

As before, we expect a neutral formula $\alpha \in \mathcal{K}$ not to increase the inconsistency degree of $\mathcal{K}$ since it does not induce inconsistency to any subset $\mathcal{H} \subseteq \mathcal{K}$. By definition, resolving conflicts is impossible as well, motivating:

Independence If $\alpha \in Ntr(\mathcal{K})$, then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$.

As *neutral* is a quite strong property, our *independence* postulate seems to be rather basic. However, note that *independence* is a generalization of *free formula independence* [68] which is not free from criticism. In [27], for example, it has been noted that the knowledge base $\mathcal{K} = \{a \wedge c, b \wedge \neg c\}$ should be considered less problematic than $\mathcal{K} \cup \{\neg a \vee \neg b\}$ even though the additional formula is free. We want to emphasize that the same concerns apply to *independence*. A weaker version can be found in [103], where a formula $\alpha \in \mathcal{K}$ is called *safe* if the atoms in α do not occur elsewhere in $\mathcal{K} \setminus \{\alpha\}$. The corresponding postulate *weak independence* is similar to *free formula independence*. However, this requirement is hard to phrase for an arbitrary logic L as in Definition 2.5.1. This suggests that weaker versions of *independence* should be tailored for a specific framework like the postulate *safe-rule independence* for answer set programming (see Section 4.5).

The final rationality postulate belonging here is *dominance* [68]. In the propositional setting, *dominance* requires that for two formulas α and β such that α is satisfiable and $\alpha \models \beta$, then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}(\mathcal{K} \cup \{\beta\})$ should hold. The postulate formalizes that α carries more information and is hence more likely to be involved in conflicts than β. Of the postulates we considered so far, it is probably the most disputed one (see [27; 72]). One of the most notable problems with this postulate was pointed out in [46]. The authors state the following result for a propositional knowledge base $\mathcal{K}$: If $\mathcal{I}$ satisfies *monotony* and *dominance*, then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \{\beta\})$ if $\alpha \models \beta$ for a satisfiable formula $\alpha \in \mathcal{K}$. They continue with describing the following case: Say $\mathcal{K} = \{\alpha_1, \dots, \alpha_n\}$ where each α_i is satisfiable and let $\mathcal{K}' = \{\beta_1, \dots, \beta_n\}$ where $\beta_i \equiv \alpha_i$ for each i. Then, *monotony* and *dominance* already imply $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$, so copying all conflicts does not change the inconsistency degree. They thus propose a refined version *dominance'* of *dominance*, where $\alpha, \beta \notin \mathcal{K}$ is additionally required.

For our setting of a non-monotonic logic, distinguishing between *dominance* and *dominance'* does not make a significant difference, and we do not need to be too concerned about the objections from [27; 72]. The reason is simply that there is a much more fundamental issue with this postulate here. Since adding information to a knowledge base may not only induce, but also potentially resolve conflicts, the intuition does not hold anymore: There is simply no reason why α, which carries more information than β, should be considered more problematic in general. We thus believe there is no meaningful generalization of *dominance* for non-monotonic logics.

Let us summarize how we obtained our refined rationality postulates so far. We did not need to adjust *consistency*. To *monotony* we added a premise based on the notion of *preserving conflicts*, a property which is inherently given in monotonic logics. We considered two refinements of *free formulas*, called *SI-free* and *neutral* and the corresponding postulates. We pointed our why *dominance* does not appear to have a natural counterpart in our setting. This concludes our discussion on the four postulates that are required for *basic inconsistency measures* [68]:

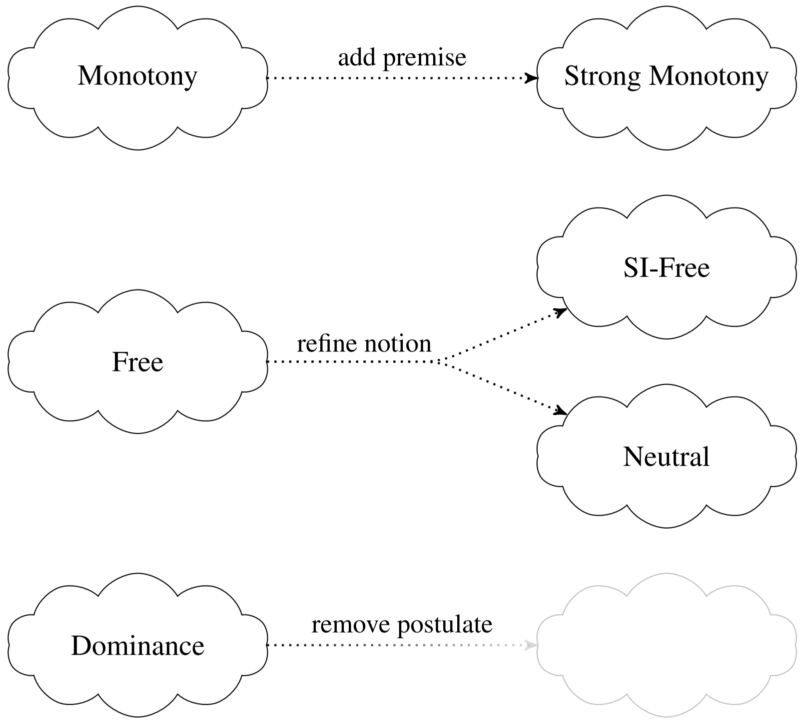


Figure 4.2: Making rationality postulates work for non-monotonic logics

4.2.2 Extended Postulates

Many concrete approaches to inconsistency measurement depend on the syntax of a knowledge base. The most common example is the difference between the conjunction $\{a \wedge b\}$ and two formulas $\{a, b\}$. To illustrate this issue, let us recall the lottery paradox from above.

Example 4.2.24. We considered the knowledge base $\mathcal{K}_n = \{t_1 \dot{\vee} \ldots \dot{\vee} t_n, \neg t_1, \ldots, \neg t_n\}$ and argued that the inconsistency degree of $\mathcal{K}_n$ should be lower the bigger n is. This was due to the number of formulas required in order to obtain a contradiction. However, if we express $\mathcal{K}_n$ as the two formulas $\mathcal{K}'_n = \{t_1 \vee \ldots \vee t_n, \neg t_1 \wedge \ldots \wedge \neg t_n\}$, then the single minimal inconsistent set of $\mathcal{K}'$ contains two formulas, independent of n.

One could now argue that even when considering $\mathcal{K}'_n$, the number n of tickets still effects the *size* of the formulas within $\mathcal{K}'_n$; but then again, taking the size of a formula into account raises some other issues: It enforces distinguishing equivalent formulas depending on how they are written down. There are thus rationality postulates in the literature which are concerned about the behavior of inconsistency measures when dealing with equivalent formulas resp. equivalent knowledge bases. Of course, it is desirable that a measure $\mathcal{I}$ is robust wrt. the syntax of $\mathcal{K}$.

The postulate *adjunction invariance* [27] formalizes the idea that there should be no difference between $\{a \wedge b\}$ and $\{a, b\}$, i. e., $\mathcal{I}(\mathcal{K} \cup \{a \wedge b\}) = \mathcal{I}(\mathcal{K} \cup \{a, b\})$. There are more postulates considering situations where (parts of) semantically equivalent knowledge bases are compared, see [103].

In non-monotonic frameworks, a notion of equivalence of the form "$\mathcal{K}$ has the same models as $\mathcal{K}'$" is too weak as conclusions can be withdrawn due to non-monotony. This observation has led to the development of strong equivalence, as already discussed in Section 3.1.3. Strong equivalence plays a similar role in non-monotonic frameworks as (normal) equivalence in monotonic frameworks. In particular, it allows for *modularisation* of knowledge bases. If a subset $\mathcal{H}$ of a knowledge base $\mathcal{K}$ is strongly equivalent to a set $\mathcal{H}'$ then $\mathcal{H}$ can be replaced in $\mathcal{K}$ by $\mathcal{H}'$ without changing the inferences one can draw from $\mathcal{K}$. This also means that $\mathcal{H}$ and $\mathcal{H}'$ should be interchangeably when it comes to the inconsistency they contribute to $\mathcal{K}$. By generalizing this idea to the whole knowledge base, we obtain that strongly equivalent knowledge bases should have the same degree of inconsistency.

Strong Equivalence If $\mathcal{K} \equiv_s \mathcal{K}'$, then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$.

However, whether or not this is desirable depends on the framework under consideration. In many cases, this postulate does not make sense. For example, in monotonic logics we have $\mathcal{K} \equiv_s \mathcal{K}'$ for any two inconsistent knowledge bases $\mathcal{K}$ and $\mathcal{K}'$, thus satisfying *strong equivalence* contradicts the idea of *quantitatively* assessing the inconsistency of a knowledge base. In ASP it still allows to distinguish between, e.g., $P = \{a.\ \neg a.\}$ and $P' = \{a \leftarrow \text{not } b.\ \neg a \leftarrow \text{not } b.\}$ as they are both inconsistent, but not strongly equivalent.

The issue with *strong equivalence* is quite straightforward: Consideration of the whole knowledge base is not fine-grained enough. One should look at the single formulas within $\mathcal{K}$ instead. This allows to compare equivalent and in particular *consistent* parts of a knowledge base. The technique we utilize is similar to [103] for the postulate "irrelevance of syntax". For our setting we define:

Definition 4.2.25. Let $\mathcal{K}$ and $\mathcal{K}'$ be two knowledge bases. We call $\mathcal{K}$ and $\mathcal{K}'$ *formula-wise strongly equivalent*, denoted by $\mathcal{K} \equiv_\alpha \mathcal{K}'$, if there is a bijection $\rho : \mathcal{K} \to \mathcal{K}'$ such that $\{\alpha\} \equiv_s \{\rho(\alpha)\}$ holds for all $\alpha \in \mathcal{K}$.

Equipped with this notion we may phrase a refinement of *strong equivalence*. Instead of requiring $\mathcal{K} \equiv_s \mathcal{K}'$, we consider two formula-wise strongly equivalent knowledge bases which yields a more meaningful rationality postulate. We thus obtain the following generalization of *irrelevance of syntax* [103] (FW=formula-wise):

FW-Strong Equivalence If $\mathcal{K} \equiv_\alpha \mathcal{K}'$, then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$.

In contrast to *strong equivalence*, the postulate comes with a quite strong premise. To illustrate this, let us mention that $\mathcal{K} \equiv_\alpha \mathcal{K}'$ induces the same property for *any* subset of $\mathcal{K}$ and $\mathcal{K}'$.

Proposition 4.2.26. *If $\mathcal{K}$ and $\mathcal{K}'$ are formula-wise strongly equivalent, then there is a bijection $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ such that $\mathcal{H} \equiv_s \tilde{\rho}(\mathcal{H})$ for any $\mathcal{H} \subseteq \mathcal{K}$. In particular, $|\mathcal{H}| = |\tilde{\rho}(\mathcal{H})|$.*

Proof. By assumption, there is a bijection $\rho : \mathcal{K} \to \mathcal{K}'$ with $\{\alpha\} \equiv_s \{\rho(\alpha)\}$ for all $\alpha \in \mathcal{K}$. So let $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ be the mapping with

$$\tilde{\rho}(\mathcal{H}) := \bigcup_{\alpha \in \mathcal{H}} \{\rho(\alpha)\}.$$

Then the claim follows by induction from

$$\gamma \equiv_s \gamma' \wedge \delta \equiv_s \delta' \Rightarrow \{\gamma, \delta\} \equiv_s \{\gamma', \delta'\},$$

which is easy to see. □

A further refinement of this notion is to consider the *replacement* of a formula α with a strongly equivalent formula α'. Note that this postulate is similar to *exchange* from [27].

Strong Equivalent Replacement If $\{\alpha\} \equiv_s \{\alpha'\}$ and $\alpha \notin \mathcal{K}$ as well as $\alpha' \notin \mathcal{K}$, then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) = \mathcal{I}(\mathcal{K} \cup \{\alpha'\})$.

To conclude this discussion on extended postulates, let us consider two final ones which are concerned about modularisation of $\mathcal{K}$. The first one is *separability* [68] which has a straightforward representation in our general context.

Separability If $SI_{min}(\mathcal{K} \cup \mathcal{K}') = SI_{min}(\mathcal{K}) \cup SI_{min}(\mathcal{K}')$ and $SI_{min}(\mathcal{K}) \cap SI_{min}(\mathcal{K}') = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$.

In other words, if the conflicts of two knowledge bases $\mathcal{K}$ and $\mathcal{K}'$ are independent, the inconsistency value of their union should decompose as the sum of the individual values.

Finally, we end our investigation with a generalization of *monotony*, namely *supper-additivity* [102]. It states that $\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ should hold whenever $\mathcal{K}$ and $\mathcal{K}'$ are disjoint. As for *strong monotony*, we need to take into account that adding information might resolve conflicts in non-monotonic frameworks. Therefore, we add the additional condition of conflict preservation to our version of *super-additivity*.

Strong Super-Additivity If $\mathcal{K}'$ and $\mathcal{K}$ preserve each other's conflicts and $\mathcal{K} \cap \mathcal{K}' = \emptyset$, then $\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

4.3 Analysis

As already mentioned, we are going to discuss the behavior of the measures with respect to the introduced postulates. For postulates that are not satisfied by a particular measure in general, we give counterexamples within the logic L_{ASP}. We also briefly discuss relations between the measures in terms of order compatibility [61] and their relation to the inconsistency graph [32].

4.3.1 Compliance with Rationality Postulates

In general, we obtain the following result on the compliance of our measures with the rationality postulates, see also the table below.

Proposition 4.3.1. *The measures* $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$ *and* $\mathcal{I}_{\mathsf{p}}$ *satisfy* consistency, strong monotony, independence, FW-strong equivalence, strong equivalent replacement, *and* strong super-additivity. *The measures* $\mathcal{I}_{\mathsf{MSI}}$ *and* $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$ *also satisfy* separability.

Proof. Consistency: Since $SI_{min}(\mathcal{K}) = \emptyset$ if and only if $\mathcal{K}$ is consistent (see Proposition 3.1.11), $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *consistency*.

Strong monotony: Let $\mathcal{K}$ and $\mathcal{K}'$ be knowledge bases such that $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$. Let $\mathcal{H} \in SI_{min}(\mathcal{K})$. By Proposition 4.2.4, we have $\mathcal{H} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$. Hence, we see $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{MSI}}(\mathcal{K} \cup \mathcal{K}')$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}(\mathcal{K} \cup \mathcal{K}')$ and $\mathcal{I}_{\mathsf{p}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{p}}(\mathcal{K} \cup \mathcal{K}')$ follow straightforwardly.

Independence: Let $\alpha \in Ntr(\mathcal{K})$. Then we have $SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\})$ according to Proposition 4.2.22. It follows that $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *independence*.

FW-strong equivalence: Let $\mathcal{K}$ and $\mathcal{K}'$ be such that $\mathcal{K} \equiv_\alpha \mathcal{K}'$. Proposition 4.2.26 implies that there is a bijection $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ such that $\mathcal{H} \equiv_s \tilde{\rho}(\mathcal{H})$ for any $\mathcal{H} \subseteq \mathcal{K}$. Furthermore, observe that if $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent then any $\mathcal{H}'$ with $\mathcal{H}' \equiv_s \mathcal{H}$ is strongly $\mathcal{K} \setminus \mathcal{H} \cup \mathcal{H}'$-inconsistent. It follows that $\mathcal{H} \in SI_{min}(\mathcal{K})$ if and only if $\tilde{\rho}(\mathcal{H}) \in SI_{min}(\mathcal{K}')$. Since $|\mathcal{H}| = |\tilde{\rho}(\mathcal{H})|$ is also guaranteed in Proposition 4.2.26, $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *FW-strong equivalence*.

Strong equivalent replacement: Similar.

Strong super-additivity: Let $\mathcal{K}'$ and $\mathcal{K}$ preserve each other's conflicts and $\mathcal{K} \cap \mathcal{K}' = \emptyset$. Proposition 4.2.4 implies $SI_{min}(\mathcal{K}) \cup SI_{min}(\mathcal{K}') \subseteq SI_{min}(\mathcal{K} \cup \mathcal{K}')$. Since $\mathcal{K} \cap \mathcal{K}' = \emptyset$ yields $SI_{min}(\mathcal{K}) \cap SI_{min}(\mathcal{K}') = \emptyset$ we see that the measures satisfy *strong super-additivity*.

Separability: Straightforward for $\mathcal{I}_{\mathsf{MSI}}$ and $\mathcal{I}_{\mathsf{MSI^C}}$. □

As already mentioned in Section 4.2, *SI-free* is not satisfied by any of the measures.

Example 4.3.2. Consider the program P given as follows:

$$P:\ a \leftarrow \text{not } a, b. \qquad a \leftarrow \text{not } c, \text{not } d. \qquad b. \quad c. \quad d.$$

We have $r = a \leftarrow \text{not } c, \text{not } d. \in \mathit{Free}_{SI}(P)$: the rule "$a \leftarrow$ not a, b." combined with the fact "b." is responsible for P being inconsistent and r cannot restore consistency as long as "c." or 'd." are present. Hence, $SI_{min}(P)$ consists of $\{a \leftarrow \text{not } a, b., b., c.\}$ and $\{a \leftarrow \text{not } a, b., b., d.\}$, i.e., $\mathcal{I}_{\mathsf{MSI}}(P) = 2$, $\mathcal{I}_{\mathsf{MSI^C}}(P) = \frac{2}{3}$ and $\mathcal{I}_{\mathsf{p}}(P) = 4$. However $SI_{min}(P \setminus \{r\}) = \{a \leftarrow \text{not } a, b., b.\}$, i.e., $\mathcal{I}_{\mathsf{MSI}}(P \setminus \{r\}) = 1$, $\mathcal{I}_{\mathsf{MSI^C}}(P \setminus \{r\}) = \frac{1}{2}$ and $\mathcal{I}_{\mathsf{p}}(P \setminus \{r\}) = 2$.

Example 4.3.3. Consider the programs P and P' given via

$$P:\ a. \quad \neg a. \qquad P':\ a. \quad \neg a. \quad a \leftarrow \neg a. \quad \neg a \leftarrow a.$$

It is easy to see that $P \equiv_s P'$ as the inconsistency in both programs cannot be repaired in any extension of them. However, we have that $\mathcal{I}(P_1) \neq \mathcal{I}(P_2)$ for all $\mathcal{I} \in \{\mathcal{I}_{\mathsf{MSI}}, \mathcal{I}_{\mathsf{MSI^C}}, \mathcal{I}_{\mathsf{p}}\}$ thus showing that *strong equivalence* is violated by all three measures.

A counterexample for *strong eqiuvalence* is easy to find since $\mathcal{K} \equiv_s \mathcal{K}'$ for any two inconsistent propoistional knowledge bases. For a counterexample of *separability* wrt. $\mathcal{I}_{\mathsf{p}}$ see [106] (already in the propositional case).

Observe that for those postulates that are generalizations of classical ones -i. e., *consistency*, *strong monotony*, *independence*, *strong super-additivity*, and *separability*- the compliance of our three measures generalizes their compliance with the corresponding postulates in the classical case, cf. [106]. The results are summarized in the following table:

	$\mathcal{I}_{\mathsf{MSI}}$	$\mathcal{I}_{\mathsf{MSI^C}}$	$\mathcal{I}_{\mathsf{p}}$
Consistency	✓	✓	✓
Strong Monotony	✓	✓	✓
SI-Free	✗	✗	✗
Independence	✓	✓	✓
Strong Equivalence	✗	✗	✗
FW-Strong Equivalence	✓	✓	✓
Strong Equivalent Replacement	✓	✓	✓
Separability	✓	✓	✗
Strong Super-Additivity	✓	✓	✓

4.3.2 Further Aspects

We investigate the measures we introduced wrt. two further aspects from the literature. First, we recall the notion of an IG measure from [32]. Afterwards, we discuss *order comparability* of our measures. Both aspects can be seen as straight generalizations from the propositional setting.

IG measures. In [32] the notion of the *inconsistency graph* is utilized in order to classify inconsistency measures. Let us briefly recall the required notions.

Definition 4.3.4. An *inconsistency graph* for a monotonic knowledge base $\mathcal{K}$ is a bipartite graph $\mathsf{IG}(\mathcal{K}) = (U, V, E)$ such that there are bijections $f_U : U \to \bigcup I_{min}(\mathcal{K})$ as well as $f_V : V \to I_{min}(\mathcal{K})$ with $E = \{\{u, v\} \mid f_U(u) \in f_V(v)\}$.

Then, the class of so called IG measures is defined: A measure $\mathcal{I}$ is IG if it can be written as $\mathcal{I}(\mathcal{K}) = f(\mathsf{IG}(\mathcal{K}))$ for a mapping f assigning non-negative real values to inconsistency graphs.

Example 4.3.5. Consider the knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$. Observe that

$$I_{min}(\mathcal{K}) = \{\{a,\ a \to b,\ \neg b\}, \{c,\ \neg c\}\}.$$

The graph $\mathsf{IG}(\mathcal{K}) = (U, V, E)$ is depicted in Figure 4.3. It is easy to see that the measure $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |I_{min}(\mathcal{K})|$ is an IG measure since $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |V|$.

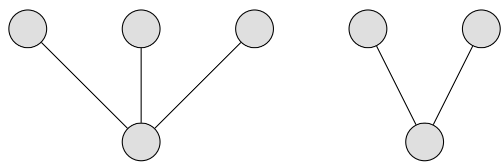


Figure 4.3: The IG graph of $\mathcal{K}$ from Example 4.3.5.

It is quite clear that the *inconsistency graph* is not appropriate for non-monotonic logics. We thus consider the straightforward refinement we need.

Definition 4.3.6. A *strong inconsistency graph* for an arbitrary knowledge base $\mathcal{K}$ is a bipartite graph $\mathsf{SIG}(\mathcal{K}) = (U, V, E)$ such that there are bijections $f_U : U \to \bigcup SI_{min}(\mathcal{K})$ and $f_V : V \to SI_{min}(\mathcal{K})$ with $E = \{\{u, v\} \mid f_U(u) \in f_V(v)\}$.

We define SIG measures in the canonical way. As a corollary of Proposition 1 in [32], we see:

Proposition 4.3.7. *The measures $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ are SIG measures.*

The paper [32] is mostly concerned about classifying inconsistency measures. For the measures considered in the present section, this classification is quite obvious: All of them are SIG measures. However, it might be more insightful (and more challenging) when it comes to inconsistency measures which are specifically tailored for certain logics. We believe this is a promising research direction for future work.

Order compatibility. In order to compare inconsistency measures we can use the notion of order compatibility cf. [61]. We say that two inconsistency measures $\mathcal{I}_1$ and $\mathcal{I}_2$ are *order-compatible* if $\mathcal{I}_1(\mathcal{K}) \leq \mathcal{I}_1(\mathcal{K}')$ iff $\mathcal{I}_2(\mathcal{K}) \leq \mathcal{I}_2(\mathcal{K}')$ for all knowledge bases $\mathcal{K}, \mathcal{K}'$. So $\mathcal{I}_1$ and $\mathcal{I}_2$ induce the same ranking on knowledge bases without necessarily assigning the same inconsistency values. As a corollary of the corresponding result from [61] for the propositional case, we obtain:

Proposition 4.3.8. *The measures* $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ *and* $\mathcal{I}_{\mathsf{p}}$ *are pairwise not order-compatible.*

Hence, all measures are incompatible and provide different points of view on inconsistency.

4.4 Measuring Inconsistent Subsets

We continue our investigation with a situation where we are not restricted to a knowledge base $\mathcal{K}$ and subsets of it, but have also additional information available. As already noted in Section 3.2, additional information do not only contribute to inconsistency of a knowledge base $\mathcal{K}$, but are also worth investigating when it comes to finding repairs. This leads to more varied situations that we also want to address in the context of measuring inconsistency. Similarly to Section 3.2, let us assume we are given a knowledge base $\mathcal{K}$ as well as a disjoint knowledge base $\mathcal{G}$. In the present section, we do not interpret $\mathcal{G}$ as a set of potential additional assumptions. We will instead assume $\mathcal{K} \cup \mathcal{G}$ is our whole knowledge base and thus, $\mathcal{K}$ is not an isolated one, but seen as a subset of $\mathcal{K} \cup \mathcal{G}$. This should be taken into account when assessing the conflicts within $\mathcal{K}$ as the following examples illustrate.

Example 4.4.1. Consider the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$. Since there is no way to infer a or c, the program P is inconsistent and, e.g., the measure $\mathcal{I}_{\mathsf{MSI}}$ assigns 1 to it. However, interpreted as part of the program $P \cup G$ with

$$\begin{array}{llll} P \cup G: & a \leftarrow \text{not } b. & c \leftarrow \text{not } d. & \leftarrow \text{not } a, \text{not } c. \\ & b \leftarrow \text{not } a. & d \leftarrow \text{not } c. & \end{array}$$

the program P simply constrains the answer sets rather than causing inconsistency. Hence, although $\mathcal{I}_{\mathsf{MSI}}(P) = 1$ seems reasonable on its own, it does not appear to make sense when considering $P \cup G$.

Example 4.4.2. Now consider the program $P \cup G$ given as follows:

$$P \cup G: \; a \vee b. \qquad \leftarrow \text{not } a. \qquad \leftarrow \text{not } b.$$

Inconsistency of $P \cup G$ stems from the two constraints "$\leftarrow$ not a." and "$\leftarrow$ not b.". As answer sets are required to be minimal models, it is not possible to satisfy both constraints simultaneously. The subset $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ obviously consists of two conflicts and this intuition is confirmed by the observation that $\mathcal{I}_{\mathsf{MSI}}(P) = 2$. However, given the disjunctive rule "$a \vee b$.", this is peculiar since there is actually only one conflict which cannot be resolved (either a or b is missing). This is confirmed by the observation $\mathcal{I}_{\mathsf{MSI}}(P \cup G) = 1$.

A simple solution simulates the concept of strong inconsistency. Recall that a subset of a knowledge base is strongly inconsistent if it contains conflicts that cannot be resolved (within $\mathcal{K}$). We can proceed similarly here and take all supersets of $\mathcal{K}$ within $\mathcal{K} \cup \mathcal{G}$ into account, looking for the smallest possible inconsistency degree. As this approach depends on a given measure $\mathcal{I}$, we obtain the following notion.

Definition 4.4.3. Let $\mathcal{I} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0}$ be an inconsistency measure and $\mathcal{K} \cup \mathcal{G}$ a knowledge base. We call

$$\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) := \min_{\mathcal{G}' \subseteq \mathcal{G}} \mathcal{I}(\mathcal{K} \cup \mathcal{G}') \tag{4.6}$$

the *value of* $\mathcal{I}(\mathcal{K})$ with respect to the context $\mathcal{K} \cup \mathcal{G}$.

This approach is quite well-behaving for the two examples we considered before.

Example 4.4.4. For the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$ with

$$\begin{array}{llll} P \cup G: & a \leftarrow \text{not } b. & c \leftarrow \text{not } d. & \leftarrow \text{not } a, \text{not } c. \\ & b \leftarrow \text{not } a. & d \leftarrow \text{not } c. & \end{array}$$

we immediately see $\mathsf{C}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 0$ caused by consistency of $P \cup G$.

Example 4.4.5. Consider again $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ where $P \cup G$ is

$$P \cup G: \quad a \vee b. \qquad \leftarrow \text{not } a. \qquad \leftarrow \text{not } b.$$

Clearly, there is only one strongly inconsistent subset of $P \cup G$, namely P. We thus see $\mathsf{C}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 1$.

Let us collect some properties of $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$, depending on the given measure $\mathcal{I}$. We see that some desirable properties of $\mathcal{I}$ transfer to $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$.

Proposition 4.4.6. *Let $\mathcal{K}, \mathcal{K}'$ and $\mathcal{G}$ be a knowledge bases.*

(a) If $\mathcal{I}$ satisfies consistency*, then $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ if and only if $\mathcal{K} \notin SI(\mathcal{K} \cup \mathcal{G})$,*

(b) if $\mathcal{I}$ satisfies independence *and $\alpha \in Ntr(\mathcal{K} \cup \mathcal{G})$, then $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = \mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K} \setminus \{\alpha\})$,*

(c) if $\mathcal{I}$ satisfies strong equivalence *and $\mathcal{K} \equiv_s \mathcal{K}'$, then $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = \mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}')$.*

Proof. Let $\mathcal{K}, \mathcal{K}'$ and $\mathcal{G}$ be knowledge bases.
(a): Assume the measure $\mathcal{I}$ satisfies the *consistency* postulate. If $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ then there is $\mathcal{K} \cup \mathcal{G}' \subseteq \mathcal{K} \cup \mathcal{G}$ such that $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = 0$. Since $\mathcal{I}$ satisfies *consistency*, $\mathcal{K} \cup \mathcal{G}'$ is consistent and therefore $\mathcal{K}$ is not strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. On the other hand, if $\mathcal{K} \notin SI(\mathcal{K} \cup \mathcal{G})$ then there is $\mathcal{G}' \subseteq \mathcal{G}$ such that $\mathcal{K} \cup \mathcal{G}'$ is consistent and $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = 0$. Then $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ as well.
(b): Let $\mathcal{I}$ satisfy *independence*. For $\alpha \in Ntr(\mathcal{K} \cup \mathcal{G})$ we have $\alpha \in Ntr(\mathcal{H})$ for any subset $\mathcal{H}$ of $\mathcal{K} \cup \mathcal{G}$. Now the claim follows immediately.
(c): Let $\mathcal{I}$ satisfy *strong equivalence* and let $\mathcal{K} \equiv_s \mathcal{K}'$. It follows that $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$ and, as $\mathcal{K} \equiv_s \mathcal{K}'$ implies $\mathcal{K} \cup \mathcal{G}' \equiv_s \mathcal{K}' \cup \mathcal{G}'$ for all $\mathcal{G}' \subseteq \mathcal{G}$, $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = \mathcal{I}(\mathcal{K}' \cup \mathcal{G}')$ and by this the claim. ☐

While the expression (4.6) simply takes the minimum over all possible supersets, it would be rather appealing to utilize a general framework for measuring the information which is added to a knowledge base. The idea of measuring information is not novel. In [61] an *information measure* is defined as a mapping $\mathcal{J}$ assigning non-negative real numbers to propositional knowledge bases and satisfying

- if $\mathcal{K} = \emptyset$, then $\mathcal{J}(\mathcal{K}) = 0$,
- if $\mathcal{K} \subseteq \mathcal{K}'$ and $\mathcal{K}'$ is consistent, then $\mathcal{J}(\mathcal{K}) \leq \mathcal{J}(\mathcal{K}')$,
- if $\mathcal{K}$ is consistent and at least one formula $\alpha \in \mathcal{K}$ is not a tautology, then $\mathcal{J}(\mathcal{K}) > 0$.

Equipped with an appropriate technique to measure information in non-monotonic logics, one could utilize $\mathcal{J}(\mathcal{A})$ (where $\mathcal{A} \subseteq \mathcal{G}$) to measure the information added to $\mathcal{K}$ and then consider $\mathcal{I}(\mathcal{K} \cup \mathcal{A})$ to measure inconsistency of the remaining conflicts within $\mathcal{K}$. This approach induces inconsistency measures based on, e.g., the expression

$$\min_{\mathcal{A} \subseteq \mathcal{G}} \mathcal{J}(\mathcal{A}) + \mathcal{I}(\mathcal{K} \cup \mathcal{A}). \tag{4.7}$$

Although we leave a thorough investigation of this issue for future work, we want to mention that (4.6) is a special case of this approach, utilizing the trivial function $\mathcal{J} \equiv 0$.[1] In the subsequent Section 4.5 on measuring inconsistency in ASP we will consider a measure $\mathcal{I}_{\pm}$ which can be interpreted as an expression of this form, where $\mathcal{J}$ counts the number of facts we add to a given program.

In order to calculate (4.6), we require a given inconsistency measure $\mathcal{I}$. We also want to consider a novel approach here. Recall the measures introduced in Section 4.1. They are based on (the number of) strongly inconsistent subsets of a knowledge base $\mathcal{K}$. The notion of strong inconsistency facilitated the previously considered generalization of inconsistency measures from the literature. The results of the previous chapter suggest that bidirectional non-repairs are worth investigating when given a knowledge base $\mathcal{K}$ as a subset of $\mathcal{K} \cup \mathcal{G}$. This motivates considering the following measures of the type

$$\mathcal{I}_{\mathcal{G}} : 2^{\mathcal{WF}} \to \mathbb{R}_{\geq 0} \text{ with } \mathcal{K} \mapsto \mathcal{I}_{\mathcal{G}}(\mathcal{K}).$$

Definition 4.4.7. Given disjoint knowledge bases $\mathcal{K}$ and $\mathcal{G}$, define

- $\mathcal{I}_{\mathrm{NR},\mathcal{G}}$ via

$$\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = |co\text{-}\textsc{bi}\text{-}\mathrm{N}\textsc{rep}_{max}(\mathcal{K},\mathcal{G})| = |\textsc{bi}\text{-}\mathrm{N}\textsc{rep}_{max}(\mathcal{K},\mathcal{G})|,$$

- $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}$ via

$$\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = \sum_{\substack{(\mathcal{D},\mathcal{A}) \in \\ co\text{-}\textsc{bi}\text{-}\mathrm{N}\textsc{rep}_{max}(\mathcal{K},\mathcal{G})}} \frac{1}{|\mathcal{D} \cup \mathcal{A}|}.$$

We observe the similarities to the measures $\mathcal{I}_{\mathsf{MSI}}$ and $\mathcal{I}_{\mathsf{MSI^C}}$. Those given in Definition 4.4.7 are similar in their spirit, replacing $SI_{min}(\mathcal{K})$ with $co\text{-}\textsc{bi}\text{-}\mathrm{N}\textsc{rep}_{max}(\mathcal{K},\mathcal{G})$. We want to emphasize that the resulting measures are rather pessimistic when it comes to assessing $\mathcal{K}$ as subset of $\mathcal{K} \cup \mathcal{G}$. To illustrate this, let us consider $\mathcal{I}_{\mathrm{NR},\mathcal{G}}$ applied to the previous examples.

[1] More precisely, $\mathcal{J} \equiv 0$ is no information measure according to [61] since the third condition is violated

Example 4.4.8. Recall the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$ with

$$P \cup G: \; a \leftarrow \text{not } b. \qquad c \leftarrow \text{not } d. \qquad \leftarrow \text{not } a, \text{not } c.$$
$$b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c.$$

Even though $P \cup G$ itself is consistent, there is a non-repairing subset G' of G, namely

$$G': \; b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c.$$

We note that

$$P \cup G': \; b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c. \qquad \leftarrow \text{not } a, \text{not } c.$$

is inconsistent. We thus see $\text{BI-NREP}_{max}(P, G) = \{(\emptyset, G')\}$, yielding $\mathcal{I}_{\text{NR},G}(P) = 1$. Moreover, $co\text{-BI-NREP}_{max}(P, G) = \{(P, G \setminus G')\} = \{(P, \{a \leftarrow \text{not } b., \; c \leftarrow \text{not } d.\})\}$. Hence, we have $\mathcal{I}_{\text{NR}^c,G}(P) = 1/3$.

So these measures punish knowledge bases $\mathcal{K}$ and $\mathcal{G}$ for each maximal bidirectional non-repair. This can be seen as a counterpart to $\mathsf{C}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$ which rewards $\mathcal{K}$ for any possibility $\mathcal{G}$ possesses to resolve a conflict. In this sense, one also could interpret $\mathcal{I}_{\text{NR},\mathcal{G}}$ and $\mathcal{I}_{\text{NR}^c,\mathcal{G}}$ as measures for the quality of the repair options provided by $\mathcal{G}$. To see the two approaches at work, let us consider the following example:

Example 4.4.9. Let P and G be the following programs:

$$P: \quad a. \quad \leftarrow \text{not } b. \qquad G: \quad \neg a. \quad b.$$

We see that $\mathsf{C}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 0$ since P has the consistent superset $P \cup \{b\}$. However, there is also one bidirectional non-repair, namely $(\{a.\}, \{\neg a.\})$, thus we find $\mathcal{I}_{\text{NR},G}(P) = 1$.

We also want to mention a quite special feature of the measure $\mathcal{I}_{\text{NR}^c,\mathcal{G}}$. Recall the motivation for defining $\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}$ in contrast to $\mathcal{I}_{\mathsf{MSI}}$, namely taking the size of a set $\mathcal{H} \in I_{min}(\mathcal{K})$ into account. The measure $\mathcal{I}_{\text{NR}^c,\mathcal{G}}$ attains larger values the bigger the sets in $\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ are. This is similar in spirit to $\mathcal{I}_{\mathsf{MSI}}$, but not quite the same since punishing the size of a tuple $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ means punishing the fact that there is a large number of formulas which are not capable of providing a repair.

Example 4.4.10. Recall $P = \{\leftarrow \text{not } a., \; \leftarrow \text{not } b.\}$ and $G = \{a \vee b.\}$, i. e.,

$$P \cup G: \; a \vee b. \qquad \leftarrow \text{not } a. \qquad \leftarrow \text{not } b.$$

As $P \cup G$ is inconsistent, we see $\text{BI-NREP}_{max}(P, G) = \{(\emptyset, G)\}$, i. e., $\mathcal{I}_{\text{NR},G}(P) = 1$.

Let us now compare the measures from Definition 4.4.7 to those from Section 4.1. We note that $\mathcal{I}_{\text{NR},\mathcal{G}}(\mathcal{K}))$ attains larger values in general. This is a corollary of Proposition 3.2.25.

Corollary 4.4.11. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then, $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\text{NR},\mathcal{G}}(\mathcal{K})$.*

Proof. We have $\mathcal{I}_{\text{NR},\mathcal{G}}(\mathcal{K}) = |\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})|$. As we know from Proposition 3.2.25, if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is an $\mathcal{A} \subseteq \mathcal{G}$ with $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$. In particular, $|SI_{min}(\mathcal{K})| \leq |\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})|$, i. e., $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\text{NR},\mathcal{G}}(\mathcal{K})$. □

We observe that $\mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K}) \leq \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K})$ does not hold in general as one can already see from Example 4.4.8. Here we had $\mathcal{I}_{\mathrm{NR}^c,G}(P) = 1/3$ where $\mathcal{I}_{\mathsf{MSI^C}}(P) = 1$. We want to emphasize that a comparison between $\mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K})$ does not appear to be very meaningful. This is because both measures depend positively on the number of undesired sets, but negatively on their size.

However we note that the expected outcome is obtained whenever $\mathcal{G}$ is empty. This is a corollary of Proposition 3.2.24.

Corollary 4.4.12. *Let $\mathcal{K}$ be a knowledge base and let $\mathcal{G} = \emptyset$. Then, $\mathcal{I}_{\mathrm{NR},\emptyset}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\emptyset}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K})$.*

Proof. Due to Proposition 3.2.24, $(\mathcal{D}, \emptyset) \in$ BI-NREP$_{max}(\mathcal{K}, \emptyset)$ iff $\mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$. Equivalently, $(\mathcal{H}, \emptyset) \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \emptyset)$ iff $\mathcal{H} \in SI_{min}(\mathcal{K})$. This proves both equations. □

We make an analogous observation whenever the underlying logic is monotonic.

Corollary 4.4.13. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let the underlying logic be monotonic. Then, $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K})$.*

Proof. Due to Proposition 3.2.23, $(\mathcal{D}, \mathcal{G}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{K} \setminus \mathcal{D} \in I_{min}(\mathcal{K})$. Equivalently, $(\mathcal{H}, \emptyset) \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{H} \in SI_{min}(\mathcal{K})$. This proves both equations. □

Let us now collect some properties of the two measures we just introduced. As before, they are adapted rationality postulates from Section 4.2. Note the differences and similarities to Proposition 4.4.6.

Proposition 4.4.14. *Given two disjoint knowledge bases $\mathcal{K}$ and $\mathcal{G}$, the measures $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\cdot)$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\cdot)$ satisfy*

(a) $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent,

(b) if $\alpha \in Ntr(\mathcal{K} \cup \mathcal{G})$, then $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathcal{G}}(\mathcal{K} \setminus \{\alpha\}) = \mathcal{I}_{\mathcal{G}\setminus\{\alpha\}}(\mathcal{K})$,

(c) if $\mathcal{G} \equiv_\alpha \mathcal{G}'$, then $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathcal{G}'}(\mathcal{K})$.

Proof. (a): Observe that $\mathcal{K} \setminus \mathcal{D}$ can never be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, whenever $\mathcal{K}$ itself is consistent. So BI-NREP$_{max}(\mathcal{K}, \mathcal{G}) = \emptyset$, i. e., $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = 0$. If $\mathcal{K}$ is inconsistent, then BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ contains at least one tuple since BI-NREP$(\mathcal{K}, \mathcal{G})$ contains at least $(\emptyset, \emptyset)$. So $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}), \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) > 0$.

(b): Clear, similar to the proof of Proposition 4.3.1.

(c): Clear, similar to the proof of Proposition 4.3.1. □

This finishes our discussion on the measures based on bidirectional non-repairs. In contrast to the measure (4.6) defined in Definition 4.4.3, the measures from Definition 4.4.7 are based on bidirectional non-repairs and thus do not require a given measure $\mathcal{I}$. One drawback of the latter is that inconsistencies which stem from defaults ("$a \leftarrow$ not a.") are not distinguished from hard-coded ones ("a." vs. "$\neg a$."). This can be achieved by measures defined via (4.7), when making appropriate choices for $\mathcal{I}$ and $\mathcal{J}$.

4.5 Measuring Inconsistency in Answer Set Programming

As we already mentioned, some inconsistency measures or rationality postulates are hard to phrase for a general notion of a logic. In this section, we will briefly demonstrate how to obtain specific notions, tailored for a given framework. Since most of our examples within this chapter were logic programs, it is natural to consider ASP as our framework. In this section, our underlying logic is L_{ASP} which we may abbreviate by $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$.

The above postulate *strong monotony* provides a meaningful refinement of *monotony* for arbitrary logics. For ASP, we may consider more concrete approaches. Let us start with the most basic case: If a program P (over atoms in A) does not contain default negation "not", i. e., $neg(r) = \emptyset$ for each $r \in P$, then P can only be inconsistent if it possesses $Lit(A)$ as the only answer set. Clearly, conflicts of this kind cannot be resolved. More precisely, we make the following observation.

Proposition 4.5.1. *If P is a logic program such that $neg(r) = \emptyset$ for each rule $r \in P$, then $SI_{min}(P) = I_{min}(P)$. Moreover, $SI_{min}(P) \subseteq SI_{min}(P \cup P')$ for any porgram P'.*

Proof. The first statement is clear. To prove $SI_{min}(P) \subseteq SI_{min}(P \cup P')$, we first observe that the reduct P^M equals P for any set M of literals. Hence for any $H' \subseteq P'$, the reduct $(P \cup H')^M = P^M \cup (H')^M$ contains P. Since $Lit(A)$ is the only model of P, the same is true for $P \cup (H')^M = (P \cup H')^M$. Hence, $P \cup H'$ is inconsistent. Since any superset of P has the form $P \cup H'$ for $H' \subseteq P'$, the claim follows. □

A simple class of rules that cannot resolve conflicts are constraints. Recall that a constraint is a rule of the form

$$r: \quad a \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n, \text{not } a.$$

where a does not occur elsewhere in P. Since they prune away answer sets of a program, we have:

Proposition 4.5.2. *If P is a logic program, then $SI_{min}(P \setminus \{r\}) \subseteq SI_{min}(P)$ for any constraint $r \in P$.*

Proof. A set $H \subseteq P$ is inconsistent iff it does not possess any consistent answer set. If this is the case, then the same is true for $H \cup \{r\}$ if r is a constraint. □

Finally, let us consider *splitting* of a logic program [80]. Splitting is an important concept in non-monotonic reasoning which allows for a modularization of the considered knowledge base. Due to non-monotonic interactions between formulas, this is usually not given. The structural properties that come along with splitting can also be utilized to analyze inconsistency. Before we formally introduce splitting, recall that each logic program P consists of rules r of the form (2.1), i. e.,

$$r: \quad l_0 \vee ... \vee l_k \leftarrow l_{k+1}, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n.$$

Definition 4.5.3. Let P be a logic program. A set U of literals is a *splitting set* for P, if $\{l_0, \ldots, l_k\} \cap U \neq \emptyset$ implies $\{l_0, \ldots, l_n\} \subseteq U$ for each rule $r \in P$. For a splitting set U, let the bottom program $bot_U(P)$ be the set of all rules $r \in P$ with $\{l_0, \ldots, l_n\} \subseteq U$.

In [80] it has been pointed out that answer sets of a program P can be computed in a stepwise fashion when given a splitting.

Theorem 4.5.4 (Splitting Theorem, see [80]). *Let U be a splitting set for a program P. Every answer set M of P is of the form $M = X \cup Y$ with an answer set X of $bot_U(P)$ and a set Y of literals.*

Given the splitting theorem, it is straightforward to infer the following result:

Proposition 4.5.5. *Let U be a splitting set of P. If $bot_U(P)$ is inconsistent, then so is P. In particular, $SI_{min}(bot_U(P)) \subseteq SI_{min}(P)$.*

The above considerations motivate the following rationality postulates which can be interpreted as possible refinements of *strong monotony* for the framework ASP.

CLP-Monotony If P does not contain default negation "not", then $\mathcal{I}(P) \leq \mathcal{I}(P \cup P')$ for any program P'.

Con-Monotony If P is a logic program and $r \in P$ is a constraint, then $\mathcal{I}(P \backslash \{r\}) \leq \mathcal{I}(P)$.

Split-Monotony If U is a splitting set for P, then $\mathcal{I}(bot_U(P)) \leq \mathcal{I}(P)$.

Indeed, all of them describe situations where the additional rules preserve conflicts of the program on the left hand side. Formally, we have:

Proposition 4.5.6. *If a measure $\mathcal{I}$ satisfies* strong monotony*, it satisfies* CLP-monotony*,* split-monotony *and* con-monotony.

Proof. This was found in Propositions 4.5.1, 4.5.2 and 4.5.5 □

However, this does not mean that considering the postulate *strong monotony* suffices in order to understand all situations where an inconsistency measure for a logic program behaves monotonically. In fact, all three postulates *CLP-monotony*, *split-monotony* and *con-monotony* can be satisfied simultaneously while *strong monotony* is not. To see this, we consider a quite simple inconsistency measure $\mathcal{I}_{\pm}$, which operates on the language of a logic program. More precisely, $\mathcal{I}_{\pm}$ aims at measuring the effort needed to turn an inconsistent program into a consistent one. To this end, it quantifies the number of modifications in terms of deleting and adding rules, necessary in order to restore consistency.

Definition 4.5.7. Define $\mathcal{I}_{\pm} : \mathcal{WF} \to \mathbb{R}_{\geq 0}$ via

$$\mathcal{I}_{\pm}(P) = \min\{|A| + |D| \mid A, D \subseteq \mathcal{WF} \text{ st. } (P \cup A) \setminus D \text{ is consistent}\}$$

Let us see how the measure operates on our running example program.

Example 4.5.8. Consider again

$$\begin{array}{llll} P: & a \vee b. & & a \leftarrow b. \\ & c \leftarrow \text{not } b. & & \neg c \leftarrow \text{not } b. \end{array}$$

Since removing the rule "$a \leftarrow b.$" yields a consistent program, we may set $D = \{a \leftarrow b.\}$ in Definition 4.5.7 and hence, $\mathcal{I}_{\pm}(P) \leq 1$. Since P is inconsistent, $\mathcal{I}_{\pm}(P) \geq 1$. Thus $\mathcal{I}_{\pm}(P) = 1$. The same observation can be made due to consistency of $P \cup \{b.\}$, i. e., we may set $A = \{b.\}$ in Definition 4.5.7.

Let us now prove that the above postulates are indeed satisfied by $\mathcal{I}_{\pm}$. For this, it will be helpful to investigate the behavior of the measure first: The definition of $\mathcal{I}_{\pm}$ allows the addition of any rule in order to restore consistency, but in fact it is sufficient to consider only addition of facts instead of general rules. First, we show that adding rules with disjunction in their head is not necessary, which is intuitive since ASP already requires minimality.

Proposition 4.5.9. *Let P be an inconsistent program. If r is a rule such that $P \cup \{r\}$ is consistent, then there is a literal $a \in head(r)$ such that $P \cup \{a \leftarrow body(r).\}$ is consistent.*

Proof. Let M be a consistent answer set of $P \cup \{r\}$. Since P is inconsistent, M cannot be an answer set of P. Thus the rule must be applicable, i. e., $body\left(\{r\}^M\right) \subseteq M$ and $\{r\}^M \neq \emptyset$ because otherwise one could delete the rule while maintaining M as answer set. It follows that $head(r) \cap M \neq \emptyset$ since M is a model of $(P \cup \{r\})^M$. Let $a \in head(r) \cap M$. We show that M is an answer set of $P \cup \{a \leftarrow body(r).\}$ as well. By definition, M is a minimal model of $(P \cup \{r\})^M$. Since $a \in M$, M is a model of $(P \cup \{a \leftarrow body(r).\})^M$ as well. Now assume M is not a minimal model and let $M' \subsetneq M$ be a model of $(P \cup \{a \leftarrow body(r).\})^M$. Since $a \in head(r)$, this implies that M' is also a model of $(P \cup \{r\})^M$. Since M was assumed to be an answer set of $(P \cup \{r\})^M$, this yields a contradiction. □

Now we are ready to show that adding facts is sufficient.

Proposition 4.5.10. *Let P be an inconsistent program. If r is a rule such that $P \cup \{r\}$ is consistent, then there is a literal $a \in head(r)$ such that $P \cup \{a.\}$ is also consistent.*

Proof. Using Proposition 4.5.9, we assume that $head(r)$ contains only one literal a. As in the proof of Proposition 4.5.9, we see that $body\left(\{r\}^M\right) \subseteq M$ and $\{r\}^M \neq \emptyset$. Since M is a model of P^M, we obtain $a \in M$. However, this means M is an answer set of $P \cup \{a.\}$. □

We mentioned in Section 4.4 that $\mathcal{I}_{\pm}$ can be written as an expression of the form (4.7) where $\mathcal{J}$ counts the number of facts added to P. For this, we set

$$\mathcal{I}_{-}(P) := \min\{|D| \mid D \subseteq \mathcal{WF} \text{ st. } P \setminus D \text{ is consistent}\}.$$

If we let $\mathcal{WF}^0 \subseteq \mathcal{WF}$ be the set of facts in $\mathcal{WF}$, then $\mathcal{J}$ is a mapping $\mathcal{J} : \mathcal{WF}^0 \to \mathbb{R}_{\geq 0}$ with $\mathcal{J}(A) := |A|$. Due to Proposition 4.5.10, $\mathcal{I}_{\pm}$ is of the form

$$\mathcal{I}_{\pm}(P) = \min_{A \subseteq \mathcal{WF}^0} \mathcal{J}(A) + \mathcal{I}_{-}(P \cup A).$$

Therefore, measures of the form (4.7) can indeed be defined in a natural way. Let us now show that $\mathcal{I}_{\pm}$ satisfies the three postulates we introduced above.

Proposition 4.5.11. *$\mathcal{I}_{\pm}$ satisfies* CLP-monotony, split-monotony *and* con-monotony.

Proof. CLP-monotony: Let P be a logic program with $neg(r) = \emptyset$ for each $r \in P$. Let P' be an arbitrary logic program. Assume $\mathcal{I}_{\pm}(P \cup P') = k$. Let $(P \cup P' \cup A) \setminus D$ is consistent with $|A| + |D| = k$. As seen in Proposition 4.5.1, $P \setminus D$ must be consistent as well since adding rules cannot restore consistency. Hence, $\mathcal{I}_{\pm}(P) \leq |D| \leq |A| + |D| = k$.

Con-monotony: If r is a constraint and $(P \cup \{r\} \cup A) \setminus D$ is consistent, then the program $(P \cup A) \setminus D$ is consistent as well.

Split-monotony: Let P be a program and U a splitting set. Let $\mathcal{I}_{\pm}(P) = k$ and assume $(P \cup A) \setminus D$ is consistent with $|A| + |D| = k$. We use Proposition 4.5.10 to assume that A is a set of facts. Thus, we observe that U is a splitting set of $(P \cup A) \setminus D$ as well. In particular, if we let A_U be the subset of A such that $r \in A_U$ if and only if $head(r) \subseteq U$, then $(bot_U(P)) \cup A_U) \setminus D$ is the corresponding bottom program. Now let M be a consistent answer set of $(P \cup A) \setminus D$. Due to Theorem 4.5.4, there is a subset $X \subseteq M$ such that X is an answer set of $(bot_U(P)) \cup A_U) \setminus D$. As a subset of the consistent set M of literals, X is consistent. Therefore, $(bot_U(P)) \cup A_U) \setminus D$ is consistent. Hence

$$\mathcal{I}_{\pm}(bot_U(P)) \leq |D| + |A_U| \leq |D| + |A| = k,$$

which finishes our proof. □

As already mentioned, despite satisfying the postulates *CLP-monotony*, *split-monotony* and *con-monotony*, $\mathcal{I}_{\pm}$ does not satisfy *strong monotony*. To see this, let us consider the following example.

Example 4.5.12. Let P and P' be the programs

$$\begin{array}{llll} P: & \leftarrow \text{not } a. & \leftarrow \text{not } b. & \leftarrow \text{not } c. \\ P': & a \leftarrow d. & b \leftarrow d. & c \leftarrow d. \end{array}$$

It is easy to see that $SI_{min}(P)$ consists of the three unsatisfied constraints. Moreover, as d cannot be entailed, P' preserves conflicts of P. Yet, $\mathcal{I}_{\pm}(P) = 3$, while $\mathcal{I}_{\pm}(P \cup P') = 1$.

This example shall illustrate that the notion of preserving conflicts might be too strong in some cases. The fact that $\mathcal{I}_{\pm}$ satisfies the three weak versions of *monotony* mentioned above confirms the intuition that $\mathcal{I}_{\pm}$ behaves quite "monotonic" as long as the additional information does not resolve conflicts. This is not surprising as $\mathcal{I}_{\pm}$ counts the number of modifications that are required on the level of formulas to restore consistency. Moving from P to $P \cup P'$ weakens the severity of the inconsistency, since a single additional rule suffices to satisfy all constraints. However, these are considerations on the level of the language of the given programs and thus hard to capture within the general notion of a logic.

We want to consider two more rationality postulates for ASP. Both of them make explicit use of the language of a given program. After our discussion regarding neutral formulas in Section 4.2.1, we already mentioned the notion of *safe* formulas [103]: A consistent formula α is *safe* wrt. a knowledge base $\mathcal{K}$ if α and $\mathcal{K}$ do not share any atoms. Therefore adding α to $\mathcal{K}$ will never introduce inconsistency. The corresponding postulate *safe-formula independence* requires $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \{\alpha\})$ whenever α is safe wrt. $\mathcal{K}$.

Since rules are directed, we can be more liberal in defining a corresponding notion of safeness for ASP and do not require entirely disjoint languages for r and P. There are two different ways in which a single rule r can influence the consistency of a program P. First, the literals in the head of r may interact with the program and thus cause or resolve inconsistency. To exclude this kind of interaction, we require that the atoms in the head of r are disjoint from the atoms occurring in P. The second and more subtle way of causing a contradiction is if r is self-contradictory in the sense that a literal is derivable if and only if it is not derivable. The simplest example of such a rule is "$a \leftarrow$ not a."; when added to a program P that does not contain the atom a, it causes inconsistency. To avoid this, we require that the literals in the head of r do not occur default-negated in the body of r.

This leads to the following definition. For this, we let $At(P)$ and $At(head(r))$ be the sets of atoms occurring in P and the head of r, respectively.

Definition 4.5.13. Let P be a disjunctive logic program and r a disjunctive rule. The rule r is called *safe* wrt. P if $At(head(r)) \cap At(P) = \emptyset$ and $head(r) \cap neg(r) = \emptyset$.

The corresponding rationality postulate is the following:

Safe-Rule Independence If r is safe wrt. P, then $\mathcal{I}(P) = \mathcal{I}(P \cup \{r\})$.

The last postulate is concerned with situations where the program can be split into two parts over disjoint sets of atoms. Intuitively, parts of a program which do not share any vocabulary elements with the rest of the program should be assessed separately wrt. inconsistency.

Language Separability If $At(P) \cap At(P') = \emptyset$, then $\mathcal{I}(P \cup P') = \mathcal{I}(P) + \mathcal{I}(P')$.

It is easy to see that our measure $\mathcal{I}_{\pm}$ satisfies both *safe-rule independence* as well as *language separability*. Let us briefly summarize our approach to obtain rationality postulates for ASP. Instead of considering the general *strong monotony* we considered special cases to find suitable premises for potential postulates, e.g., the added rule is a *constraint*. By considering the language of a program we were able to refine the *safe formula independence* postulate and even weaken the premise since we are only concerned about the head of the added rule. Moreover, we gave a version of *separability* which is based on the language instead of the (strongly) inconsistent subsets:

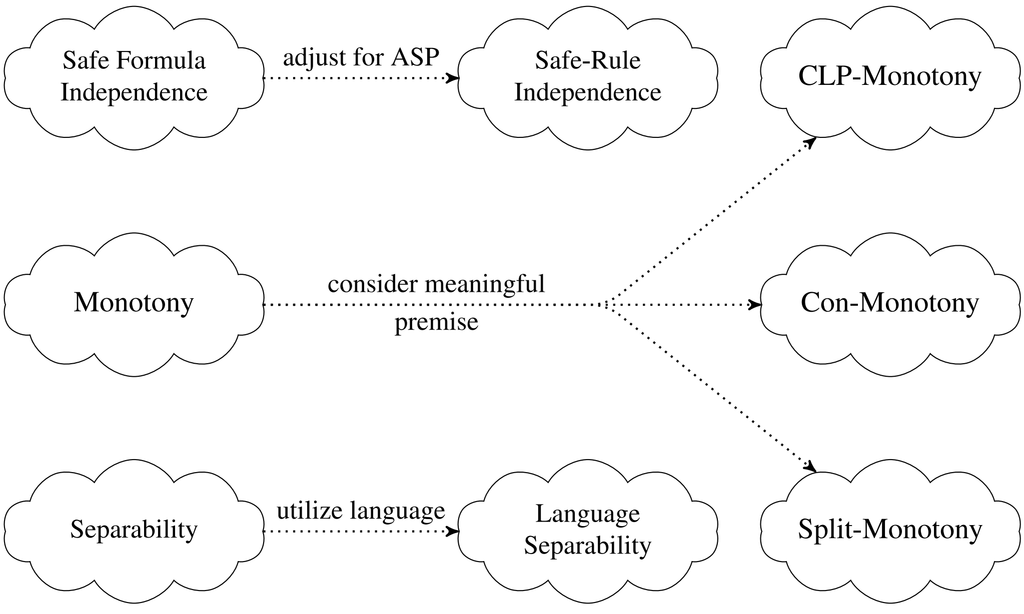


Figure 4.4: Obtaining rationality postulates for ASP

To conclude our discussion on measuring inconsistency in ASP, let us demonstrate how to utilize the specifics of ASP semantics in order to tailor inconsistency measures for this framework. The following measure compares the distance between certain sets of literals. Distance-based measures have also been used in the setting of propositional logic, see for instance [62]. Our measure makes use of the reduct P^M of a program P wrt. a given set M of literals. Recall that the reduct is defined via

$$P^M = \{head(r) \leftarrow pos(r) \mid head(r) \leftarrow pos(r), neg(r) \in P, neg(r) \cap M = \emptyset\}.$$

Recall (in Example 2.3.7) we demonstrated that our running example logic program

$$\begin{array}{lll} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

does not have an answer set. We considered $\{a\}$, $\{b\}$ and $\{a, b\}$ with reducts

$$\begin{array}{lllll} P^{\{a\}}: & a \vee b. & a \leftarrow b. & c. & \neg c. \\ P^{\{b\}}: & a \vee b. & a \leftarrow b. & & \\ P^{\{a,b\}}: & a \vee b. & a \leftarrow b. & & \end{array}$$

Neither of them is an answer set of P, for different reasons:

- The reduct wrt. $\{a\}$ is an inconsistent program; the conflict is "$c.$" vs. "$\neg c.$".
- The reduct wrt. $\{b\}$ is consistent; but $\{b\}$ is not a model.
- The reduct wrt. $\{a, b\}$ is consistent and $\{a, b\}$ is a model of it; the conflict stems from the minimality requirement of the ASP semantics.

One could thus argue that $\{b\}$ and $\{a, b\}$ are "better" candidates than $\{a\}$. To formalize this intuition, we need to specify what it means for a set of literals to be a "good" or a "bad" candidate. Since an answer set M of P is required to be an answer set of the reduct P^M, we could measure how close M to the answer sets of P^M is. Then the best candidate M determines the inconsistency degree of P. To compare M to the answer sets of P^M we utilize the symmetric difference. This yields:

Definition 4.5.14. For two sets M and M' of literals define the symmetric difference $M \Delta M'$ between M and M' as usual via $M \Delta M' = (M \cup M') \setminus (M \cap M')$. Define $\mathcal{I}_{sd} : \mathcal{WF} \to \overline{\mathbb{R}}_{\geq 0}$ via

$$\mathcal{I}_{sd}(P) = \min \left\{ |M \Delta M'| \mid M' \in \mathcal{ACC}(P^M), M, M' \text{ consistent} \right\}. \tag{4.8}$$

where $\min \emptyset := \infty$.

Example 4.5.15. We are aware that the program

$$\begin{array}{lll} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

is inconsistent. One can see in (4.8) that $\mathcal{I}_{sd}(P) \geq 1$ whenever P is inconsistent. Now consider the candidate $M = \{a, b\}$. We have

$$\begin{array}{lll} P^{\{a,b\}}: & a \vee b. & a \leftarrow b. \end{array}$$

with answer set $M' = \{a\}$. This yields the symmetric difference $M \Delta M' = \{b\}$ and hence, $\mathcal{I}_{sd}(P) \leq 1$, i. e., $\mathcal{I}_{sd}(P) = 1$.

The case $\mathcal{I}_{sd}(P) = \infty$ occurs whenever the given program allows to infer complementary literals without applying default-negated literals.

In order to emphasize that $\mathcal{I}_{sd}$ is indeed well-behaving, let us mention that it satisfies the rationality postulates we introduced within this section, as it was the case for $\mathcal{I}_{\pm}$.

Proposition 4.5.16. *$\mathcal{I}_{sd}$ satisfies* CLP-monotony*,* split-monotony*,* con-monotony*,* safe-rule independence*, and* language separability.

Proof. We only prove *split-monotony* since the other postulates are straightforward. Let P be a disjunctive logic program, U a splitting set and $bot_U(P)$ the corresponding bottom program. The case $\mathcal{I}_{sd}(P) = \infty$ is clear. We consider $\mathcal{I}_{sd}(P) = k < \infty$. Let M be a consistent set of literals with $|M \Delta M'| = k$ for a consistent set M' with $M' \in \mathcal{ACC}(P^M)$. Note that U is also a splitting set of P^M with $(bot_U(P))^M$ as the corresponding bottom part. So according to Theorem 4.5.4, M' is of the form $X' \cup Y'$ with

$$X' \in \mathcal{ACC}\left((bot_U(P))^M\right).$$

We can w. l. o. g. assume $X' \cap Y' = \emptyset$. Now consider $X = M \cap U$ and $Y = M \setminus X$. Then similarly to X' and Y' we have $X \cup Y = M$ and $X \cap Y = \emptyset$. We obtain

$$(bot_U(P))^M = (bot_U(P))^X$$

by construction of the reduct and since U is a splitting set. Thus

$$X' \in \mathcal{ACC}\left((bot_U(P))^X\right).$$

The last step argues that the constructed sets are disjoint: We have $X, X' \subseteq U$ as well as $Y \cap U = Y' \cap U = \emptyset$. So $X \cap Y' = X' \cap Y = \emptyset$. Furthermore, $X \cap Y = X' \cap Y' = \emptyset$ was already mentioned. Thus, we can calculate

$$X \Delta X' \leq X \cup Y \Delta X' \cup Y' = M \Delta M' = k.$$

To summarize, we found two consistent sets X, X' of literals with $X' \in \mathcal{ACC}\left((bot_U(P))^X\right)$ and $X \Delta X' \leq k$. Hence, $\mathcal{I}_{sd}(bot_U(P)) \leq k$. □

Clearly, rationality postulates as well as inconsistency measures that are tailored for a specific framework can take more sophisticated situations into account. In Figure 4.1 it was not quite clear how to properly select an inconsistency measure for a given application. The analysis we performed during this chapter suggests to first select some general rationality postulates the measure should satisfy. The remaining measures can then be refined by some postulates specific for the framework under consideration:

Figure 4.5: Finding the best inconsistency measure

We hope this section convinced the reader that considering specific frameworks and utilizing their properties in order to investigate inconsistency is worth the effort. Although the generality of the measures in Section 4.1 as well as the rationality postulates in Section 4.2 is appealing, it is clearly not possible to cover all conceivable aspects within this general setting. This discussion on ASP shall demonstrate how the investigation can be enhanced, covering general as well as ASP-specific aspects.

4.6 Conclusion and Related Work

In this chapter, we made first steps towards measuring inconsistency in a general, possibly non-monotonic framework by revisiting rationality postulates for propositional logic and adjusting them for our setting. Utilizing those postulates, we examined the behavior of measures based on minimal strongly inconsistent subsets, a generalization of minimal inconsistent subsets to non-monotonic frameworks. Our results show that the measures are indeed well behaving as they satisfy desired rationality postulates. Moreover, we pointed out that a thorough understanding of inconsistency in non-monotonic logics requires consideration of supersets of a given knowledge base as well. We discussed this issue with measures motivated by duality characterizations for this setting which we found in Section 3.2. We also demonstrated the importance of covering specific properties of particular frameworks by investigating postulates and measures for ASP.

Inconsistency measurement in non-classical frameworks has been addressed in some limited fashion before [4; 42; 45; 92; 103]. The latter paper studies disagreement in argumentation graphs, a notion slightly different from inconsistency. It will nevertheless be interesting to see whether postulates for disagreement are applicable to inconsistency as well. Moreover, the paper [68] considers the following problem: Given an inconsistent knowledge base $\mathcal{K}$, how much of the "blame" should be assigned to a particular formula $\alpha \in \mathcal{K}$? For this setting, minimal inconsistent subsets of a knowledge base are a rather useful tool. This facilitates a smooth generalization within our setting due to our notion of strong inconsistency. We believe that consideration of inconsistency values for non-monotonic logics is an interesting and quite promising research direction. However, the notion of strong inconsistency might not be appropriate in all cases as it depends on the structure of the whole knowledge base. In addition to that, for non-monotonic logics, an investigation of the "blame" of a formula also requires some kind of "reward" for resolving conflicts. This is no issue in monotonic logics and thus requires novel considerations. We leave an investigation of inconsistency values for non-monotonic logics for future work.

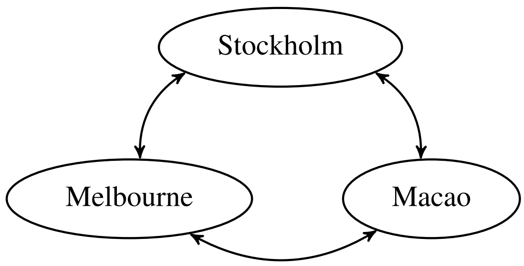
Chapter 5

Inconsistency in Abstract Argumentation

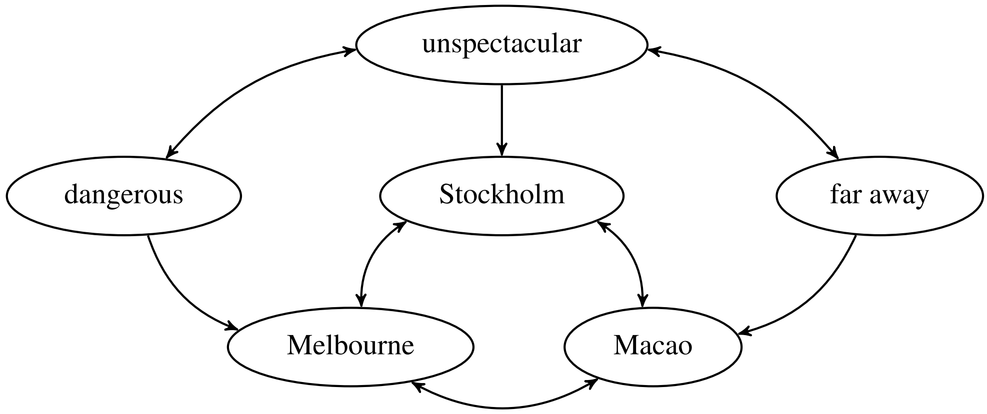
In this chapter, we consider an AF as an agent's knowledge base and the associated extensions correspond to her beliefs (cf. [44; 47; 88] for similar approaches). We focus on inconsistency in abstract argumentation frameworks, covering various semantics and credulous as well as skeptical reasoning. This extends our previous investigation of AFs where we focused on stable semantics and credulous reasoning in the sense that existence of at least one stable extension was sufficient to render an AF consistent.

The starting point of our study is a semantical defect of an agent's AF which prevents her from drawing any plausible conclusion in the sense that no argument is accepted. Our aim is to obtain an agent which is able to act. Therefore we want to know what are minimal diagnoses of the given knowledge base, i.e., which parts are causing the semantical defect. We focus on both attacks as well as arguments. The knowledge about these diagnoses may make it easier to decide what to do next. For instance, a certain minimal diagnosis may consist of arguments which are somehow out of date or not as significant in comparison to the others. Consequently, one may tend to discard these arguments. To illustrate a situation like this, let us consider the following example.

Example 5.0.1. Assume an agent is planning her vacation. The agent's preferred travel destinations are Macao, Stockholm and Melbourne. She only wants to visit one of them:



The agent is aware of the many poisonous animals in Australia and hence believes Melbourne is quite dangerous. Macao is very far away. On the other hand, she visits Europe quite often and thus finds Stockholm less spectacular than the other two options. She deems being at a dangerous place or pretty far away as spectacular since it is unusual. There is no relation between the distance and potential risks since dangerous places can be found anywhere in the world.



Although this AF possesses stable extensions, our agent is not satisfied since when it comes to reasoning about travel destinations, she is quite skeptical. However, there is no skeptically accepted argument. Our agent is thus unable to formally decide which city to visit. As a possible solution, she considers grounded semantics. Each framework has a unique grounded extension, but in case of the Agent's AF this is the empty set. Other semantics induce similar problems. She would thus be interested in techniques to modify this AF in a reasonable way until it possesses accepted arguments.

The main aim of this chapter is to study such semantical defects with regard to the following naturally arising questions:

- *Diagnosis* – Which sets of arguments are causing the collapse?
- *Properties* – Do diagnoses always exist? Are there certain preferred diagnoses? How computationally costly is it to verify a candidate diagnosis?
- *Computation* – How to compute one or even all diagnoses? Can we apply results from Chapter 3?
- *Repair* – How to use this information to obtain an agent which is indeed able to act?

Our study focuses on modifying a given AF by removing a certain set of arguments or attacks.

5.1 Background in Abstract Argumentation

As already mentioned, we do not restrict the investigation to stable semantics. We will thus extend the notions we introduced in Section 2.4 regarding AFs.

Recall that an abstract argumentation framework is a directed graph $F = (A, R)$ where nodes in A represent arguments and the relation R models *attacks*, i. e., for $a, b \in A$, if $(a, b) \in R$ we say that a attacks b. We say that F is *self-controversial* if each argument attacks itself. If not stated otherwise, we restrict ourselves to non-empty finite AFs (cf. [20; 21] for a treatment of unrestricted AFs). Formally, we introduce an infinite reference set $\mathcal{U}$, so-called *universe of arguments* and require for any possible AF, $A \subseteq \mathcal{U}$. The collection of all possible AFs is abbreviated by $\mathcal{F}$.

For a set E we use E^+ for $\{b \mid (a, b) \in R, a \in E\}$ and define $E^{\oplus} = E \cup E^+$. In case we need to be specific about the AF under consideration, we use the more informative notation $E_F^{\oplus}$. A further essential notion in argumentation is *defense*. More precisely, an argument b is *defended by* a set A if each attacker of b is counter-attacked by some $a \in A$. Then, the *characteristic function* of the AF F is a mapping $\Gamma_F : 2^A \rightarrow 2^A$ defined as $\Gamma_F(E) = \{a \in A \mid a \text{ is defended by } E\}$.

An *extension-based semantics* $\sigma : \mathcal{F} \rightarrow 2^{2^{\mathcal{U}}}$ is a function which assigns to any AF $F = (A, R)$ a set of sets of arguments $\sigma(F) \subseteq 2^A$. Each one of them, so-called σ-*extension*, is considered to be acceptable with respect to F. Besides conflict-free and admissible sets (abbreviated cf and ad) and the stable semantics we already introduced, we consider semi-stable, complete, preferred, grounded, ideal and eager semantics (abbreviated ss, co, pr, gr, il and eg respectively). Recent overviews can be found in [9; 10].

Definition 5.1.1. Let $F = (A, R)$ be an AF and $E \subseteq A$.

1. $E \in cf(F)$ iff there are no $a, b \in E$ satisfying $(a, b) \in R$,
2. $E \in ad(F)$ iff $E \in cf(F)$ and E defends all its elements,
3. $E \in ss(F)$ iff $E \in ad(F)$ and there is no $\mathcal{I} \in ad(F)$ satisfying $E^{\oplus} \subset \mathcal{I}^{\oplus}$,
4. $E \in co(F)$ iff $E \in ad(F)$ and for any $a \in A$ defended by E, $a \in E$ (equivalently, $E \in cf(F)$ and $\Gamma(E) = E$),
5. $E \in pr(F)$ iff $E \in co(F)$ and there is no $\mathcal{I} \in co(F)$ satisfying $E \subset \mathcal{I}$ (equivalently, $E \in cf(F)$ and $\Gamma(E) = E$ and $\subseteq$-maximal wrt. the conjunction of both properties),
6. $E \in gr(F)$ iff $E \in co(F)$ and there is no $\mathcal{I} \in co(F)$ satisfying $\mathcal{I} \subset E$ (equivalently, $E \in cf(F)$ and $\Gamma(E) = E$ and $\subseteq$-minimal wrt. the conjunction of both properties),
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.

We say that 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*. All semantics apart from stable are universally defined. In addition, grounded, ideal and eager semantics are examples of uniquely defined semantics. Stable semantics may *collapse*, i.e., there are AFs F, st. $stb(F) = \emptyset$. For two semantics σ and τ we write $\sigma \subseteq \tau$ if for any AF F, $\sigma(F) \subseteq \tau(F)$. For instance, it is well-known that $stb \subseteq ss \subseteq pr \subseteq co \subseteq ad \subseteq cf$.

In this chapter are interested in situations where a given AF $F = (A, R)$ does not possess accepted arguments. To make the notion of acceptance precise, we utilize the usual two alternative reasoning modes, namely *credulous* and *skeptical acceptance*. Note that we require $\sigma(F)$ to be non-empty for skeptical reasoning in order to avoid the (for our purpose) unintended situation that every argument is skeptically accepted due to technical reasons. In our setting it makes sense to define the expression $\bigcap \emptyset$ as the empty set since the (skeptical or credulous) acceptance of an argument a should imply the existence of at least one set containing a.

Definition 5.1.2. Let σ be any semantics, $F = (A, R)$ an AF and $a \in A$ an argument. We say that a is

- *credulously accepted wrt.* σ if $a \in \bigcup \sigma(F)$,
- *skeptically accepted wrt.* σ if $a \in \bigcap \sigma(F)$ and $\sigma(F) \neq \emptyset$.

As already mentioned, our motivation for a concept of *inconsistency* is a semantical defect of an agent's AF which prevents her from drawing any plausible conclusion in the sense that nothing is accepted. This is clearly not only relevant for credulous reasoning (at least one extension is non-empty), but also skeptical reasoning (there are undisputed arguments). Since we require $\sigma(F) \neq \emptyset$ for skeptical acceptance, we naturally obtain the following notion of inconsistent argumentation frameworks.

Definition 5.1.3. Let σ be any semantics and $F = (A, R)$ an AF. We say that F is

- *inconsistent wrt. credulous reasoning and* σ if $\bigcup \sigma(F) = \emptyset$,
- *inconsistent wrt. skeptical reasoning and* σ if $\bigcap \sigma(F) = \emptyset$ where we let $\bigcap \emptyset = \emptyset$.

We omit the specifications "wrt. credulous reasoning" and/or "wrt. σ" whenever the reasoning mode, the semantics or both are implicitly clear or do not matter.

We want to mention that these notions of consistency and inconsistency can be captured by our Definition 2.5.1 of a logic. This is true in general and not only for AFs: Given any logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ we may consider

$$\mathcal{ACC}_{cred}(\mathcal{K}) = \bigcup_{\mathcal{B} \in \mathcal{ACC}(\mathcal{K})} \mathcal{B} \qquad \text{or} \qquad \mathcal{ACC}_{skep}(\mathcal{K}) = \bigcap_{\mathcal{B} \in \mathcal{ACC}(\mathcal{K})} \mathcal{B}$$

with appropriately refined $\mathcal{INC}$ to model credulous or skeptical acceptance of at least one element.

Having established this background including our notion of inconsistent AFs we are now ready to tackle the problem of *repairing* AFs by moving to an appropriate subframework. In the following section, we will discuss different notions of repairs, whether they exists and connections between them.

5.2 On the Existence of Repairs

Clearly, before computing potential repairs one may wonder what types of repairs exist and whether there are minimal diagnoses at all. In this section we provide the formal notions and results wrt. this problem and in particular, we give an affirmative answer for nearly all considerable cases. We also investigate the relationship between different repairs. Unfortunately, the existence of a least repair is not guaranteed in general which calls for a duality characterization of all minimal repairs. This issue will be addressed in the subsequent section.

5.2.1 Notions for Repairs

Our repair approach involves moving to subgraphs of a given AF. Let us start by introducing the required notions and formalizing the concepts of diagnoses and repairs. Consider an AF

$F = (A, R)$. For a given set $\mathcal{S} \subseteq A$ of arguments we use $F_{\mathcal{S}}$ as a shorthand for the restriction of F to the set $A \setminus \mathcal{S}$, i. e.,

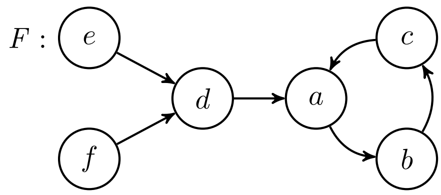
$$F|_{A\setminus\mathcal{S}} := \big(A|_{A\setminus\mathcal{S}}, R|_{A\setminus\mathcal{S}\cap A\setminus\mathcal{S}}\big) = (A \setminus \mathcal{S}, \{(a, b) \in R \mid a, b \in A \setminus \mathcal{S}\}) .$$

In other words, for $\mathcal{S} \subseteq A$, $F_{\mathcal{S}}$ is the subframework of F induced by the removal of arguments in $\mathcal{S}$. Analogously, for a given set $\mathcal{S} \subseteq R$ of attacks we use $F_{\mathcal{S}}$ as a shorthand for $(A, R \setminus \mathcal{S})$. We will also sometimes abuse notation and write $F \cup \{a\}$ and $F \cup \{(a, b)\}$ instead of $F \cup (\{a\}, \emptyset)$ and $F \cup (\{a, b\}, \{(a, b)\})$, respectively.

Let us consider an extended version of our running example AF F.

Example 5.2.1. Consider the AF $F = (A, R)$ with

$$A = \{a, b, c, d, e, f\} \qquad R = \{(a, b),\ (b, c),\ (c, a),\ (d, a),\ (e, d),\ (f, d)\}.$$



Let $\mathcal{S} = \{d\} \subseteq A$. Then

$$\begin{aligned} F_{\mathcal{S}} &= (A \setminus \{d\},\ \{(a, b) \in R \mid a, b \in A \setminus \{d\}\}) \\ &= \big(\{a, b, c, e, f\},\ \{(a, b), (b, c), (c, a)\}\big). \end{aligned}$$

Let $\mathcal{S}' = \{(a, b)\} \subseteq R$. Then

$$\begin{aligned} F_{\mathcal{S}'} &= (A, R \setminus \{(a, b)\}) \\ &= \big(\{a, b, c, d, e, f\},\ \{(b, c), (c, a),\ (d, a),\ (e, d),\ (f, d)\}\big). \end{aligned}$$

The AFs $F_{\{d\}}$ and $F_{\{(a,b)\}}$ are depicted in Figure 5.1 (1) and (2), respectively:

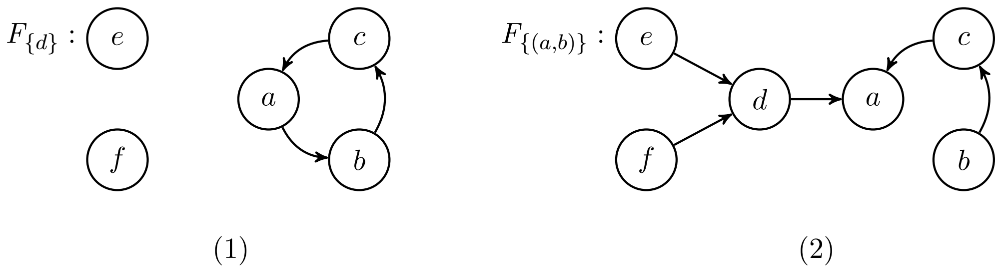


Figure 5.1: AFs $F_{\{d\}}$ and $F_{\{(a,b)\}}$ discussed in Example 5.2.1

Following the usual notions of repairs and diagnoses of knowledge bases we define for our abstract argumentation setting:

Definition 5.2.2. Let $F = (A, R)$ be an AF and σ any semantics. We call $\mathcal{S} \subseteq A$ ($\mathcal{S} \subseteq R$) an *argument-based* (*attack-based*) *σ-cred-diagnosis* of F if $F_\mathcal{S}$ is consistent wrt. credulous reasoning. Moreover, we call the AF $F_\mathcal{S}$ an *argument-based* (*attack-based*) *σ-cred-repair* of F. As usual we use the terms *minimal* and *least* for $\subseteq$-minimal or $\subseteq$-least σ-diagnosis as well as the associated σ-repairs. We define (minimal, least) *σ-skep-diagnoses* and *σ-skep-repairs* analogously.

If clear from context or unimportant we drop the considered semantics and/or reasoning mode.

Recall that during the previous chapters, we considered AFs only with stable semantics. In Example 2.5.5 we modeled a given AF $F = (A, R)$ as a knowledge base $\mathcal{K} = R$ and left A implicit. Then a maximal consistent subset $\mathcal{H}$ of $\mathcal{K}$ was a set of attacks which yielded an AF $F = (A, \mathcal{H})$ with at least one stable extension. This is a special case of Definition 5.2.2 since it corresponds to an attack-based stb-*cred*-repair of F. More precisely, if we let $\mathcal{S} = \mathcal{K} \setminus \mathcal{H}$, then $\mathcal{S}$ is an attack-based stb-*cred*-diagnosis and $F_\mathcal{S}$ an attack-based stb-*cred*-repair.

Example 5.2.3. Consider again F and $\mathcal{S}$ as well as $\mathcal{S}'$ from Example 5.2.1, i. e., $F = (A, R)$ with

$$A = \{a, b, c, d, e, f\} \qquad R = \{(a, b),\ (b, c),\ (c, a),\ (d, a),\ (e, d),\ (f, d)\}$$

and $\mathcal{S}, \mathcal{S}'$ with $\mathcal{S} = \{d\}$ and $\mathcal{S}' = \{(a, b)\}$. Let $\sigma = stb$. We see that $F_\mathcal{S}$ possesses no stable extension due to the (still existing) odd loop. Thus, $\mathcal{S}$ is no stb-diagnosis of F. However, $\mathcal{S}'$ is a stb-diagnosis, since $F_{\mathcal{S}'}$ possesses the unique stable extension $\{a, b, e, f\}$. So, $F_{\mathcal{S}'}$ is a stb-repair. Since only one attack is removed, it is quite easy to see that $\mathcal{S}'$ is even a minimal diagnosis.

Our setting is depicted in Figure 5.2. Given an inconsistent AF $F = (A, R)$, there are several choices we can make when repairing it:

- Which semantics are we interested in?
- Which reasoning mode is appropriate?
- Should we remove arguments or attacks?

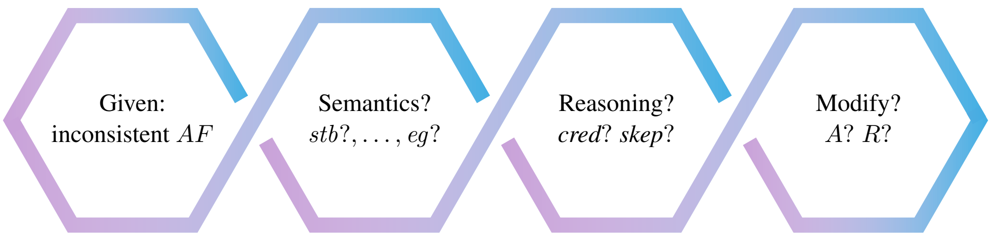


Figure 5.2: Repairing an AF: Choices

5.2.2 Relations between Credulous and skeptical Reasoning Mode

We start with some general relations between credulous and skeptical diagnoses. The following theorem applies to any semantics. It states that minimal credulous diagnoses can be found as subsets of skeptical diagnoses.

Theorem 5.2.4. *Let $F = (A, R)$ be an AF and σ any semantics. If $\mathcal{S}$ is a skep-σ-diagnosis of F, then there is a minimal cred-σ-diagnosis $\mathcal{S}'$ of F st. $\mathcal{S}' \subseteq \mathcal{S}$.*

Proof. Let $\mathcal{S}$ be a skep-σ-diagnosis of F. This means, $\bigcap \sigma(F_{\mathcal{S}}) \neq \emptyset$. Consequently, $\sigma(F_{\mathcal{S}}) \neq \emptyset$ and therefore $\bigcup \sigma(F_{\mathcal{S}}) \neq \emptyset$. Thus, $\mathcal{S}$ is a cred-σ-diagnosis of F. Moreover, by finiteness of $\mathcal{S}$ we deduce the existence of a minimal cred-σ-diagnosis $\mathcal{S}'$ of F with $\mathcal{S}' \subseteq \mathcal{S}$ concluding the proof. □

Observe that the above theorem holds for both argument-based as well as attack-based diagnoses.

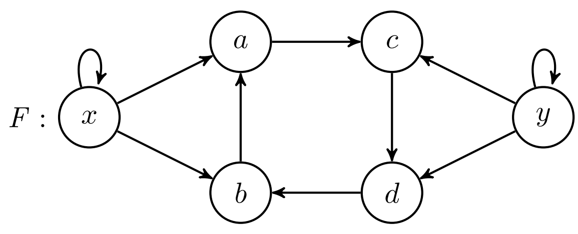
Vice Versa, skeptical diagnoses can be found as supersets of credulous ones. We want to mention two issues. First, in contrast to the assertion before, the proof of Theorem 5.2.5 requires semantics specific properties and thus, does not hold for any argumentation semantics. Secondly, it is not quite clear whether even minimality can be shown. More precisely, the situation differs depending on the semantics and whether argument-based or attack-based diagnoses are considered. Let us start with the existence of an arbitrary repair:

Theorem 5.2.5. *Let $F = (A, R)$ be an AF and $\sigma \in \{stb, ss, co, pr, gr, il, eg\}$. If $\mathcal{S}$ is a cred-σ-diagnosis of F then there is a skep-σ-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$.*

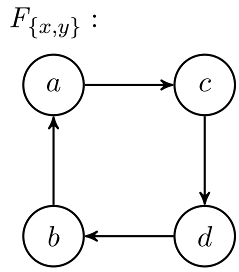
Proof. We give two different proofs depending on whether $\mathcal{S}$ is argument- or attack-based.
Argument-based: Assume the set $\mathcal{S} \subseteq A$ is a cred-σ-diagnosis of F. Then there is an argument $a \in \bigcup \sigma(F_{\mathcal{S}}) \neq \emptyset$. Since a σ-extension is conflict-free, $(a, a) \notin F_{\mathcal{S}}$. So define $\mathcal{S}' = (A \setminus \{a\}, \emptyset)$ yielding $F_{\mathcal{S}'} = (\{a\}, \emptyset)$. Thus, we obtain $\sigma(F_{\mathcal{S}'}) = \{\{a\}\}$ for any semantics σ. Hence, $\bigcap \sigma(F_{\mathcal{S}'}) = \{a\}$ proving that $\mathcal{S}'$ is a skep-σ-diagnosis of F.
Attack-based: This is trivial because $\mathcal{S}' = R$ yields an AF with no attacks, so $\sigma(F_{\mathcal{S}'}) = \{A\}$ for any considered semantics σ. Moreover, $\mathcal{S} \subseteq \mathcal{S}'$ for any attack-based diagnosis $\mathcal{S}$. □

Now we turn to minimality, i. e., the following problem: Assume we are given an AF F, a semantics $\sigma \in \{stb, ss, co, pr, gr, il, eg\}$ and a *minimal* cred-σ-diagnosis $\mathcal{S}$ of F, is there a *minimal* skep-σ-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$? Let us first mention the trivial cases $\sigma \in \{gr, eg, il\}$ where the reasoning modes coincide as there is exactly one extension. Clearly, minimality is given as noted in Corollary 5.2.13. For the other semantics the answer differs depending on the typ of diagnoses under consideration. So let us start with argument-based ones. Here we have the following counterexample for $\sigma \in \{ss, co, pr\}$.

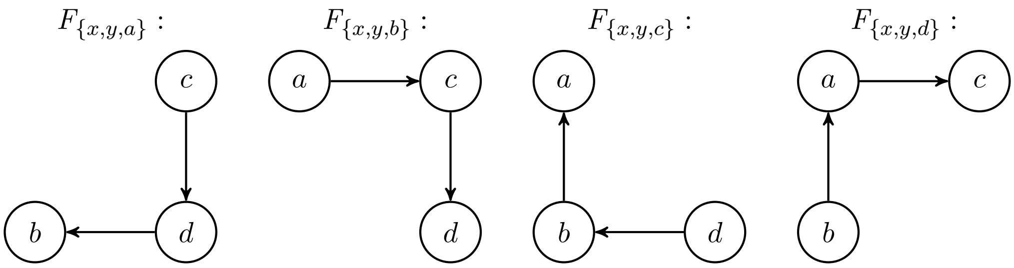
Example 5.2.6. Consider the following AF $F = (A, R)$:



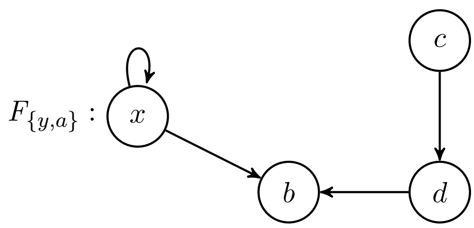
Observe the structure of this AF: We have an even loop consisting of arguments a, b, c and d which is disturbed by two dummy arguments x and y. Let $\sigma \in \{ss, pr, co\}$. Clearly, there is no way to defend anything from x and y and thus $\emptyset$ is the only σ-extension. Hence, F is inconsistent wrt. σ. Now consider the AF $F_{\{x\}}$, i. e., the argument x is removed. We still have $\emptyset$ as the only extension: The two candidates $\{a, d\}$ and $\{b, c\}$ still lack defense against y. However, removal of x and y yields the following AF $F_{\{x,y\}}$ possessing the two non-empty extensions $\{a, d\}$ and $\{b, c\}$:



We thus found the minimal σ-$cred$ diagnosis $\{x, y\}$. In order to extend this diagnosis to a σ-$skep$ diagnosis, we need to remove a, b, c or d as well, so we have σ-$skep$ diagnoses $\{x, y, a\}$, $\{x, y, b\}$, $\{x, y, c\}$ and $\{x, y, d\}$. The corresponding repairs as depicted below:



However, there is no *minimal* σ-$skep$ diagnosis among them. For example, in $F_{\{x,y,a\}}$ the argument a is removed in order to render c skeptically accepted. This does not depend on x, so there is no harm in moving to $F_{\{y,a\}}$ instead; c is still skeptically accepted:



Due to symmetry, we see in addition the minimal σ-$skep$-diagnoses $\{y, c\}$, $\{x, b\}$ and $\{x, d\}$. There is thus no minimal σ-$skep$-diagnosis $\mathcal{S}'$ with $\{x, y\} \subseteq \mathcal{S}'$.

For stable semantics, this is an open problem. We conjecture that minimality can be guaranteed, but did not find a proof so far.

Conjecture 5.2.7. *Let F be an AF. If $\mathcal{S}$ is a minimal argument-based stb-$cred$-diagnosis of F, then there is a minimal stb-$skep$-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$.*

Let us now turn to attack-based diagnoses. They are more fine-grained since removing a single attack is just removing an attack, whereas removing an argument yields an arbitrary amount of removed attacks. We can answer the question regarding minimality affirmatively for preferred, stable and semi-stable semantics. We have the following, even stronger result.

The proof of the theorem below illustrates how precise attack-based diagnoses operate: Given $(a,b) \in \mathcal{S}$ for a minimal σ-diagnosis ($\sigma \in \{ss, pr, stb\}$), then b is guaranteed to be skeptically accepted.

Theorem 5.2.8. *Let $F = (A, R)$ be an AF and let $\sigma \in \{ss, pr, stb\}$. Assume F is inconsistent wrt. σ and credulous reasoning. If $\mathcal{S} \subseteq R$ is a minimal σ-$cred$-diagnosis of F, then $\mathcal{S}$ is a minimal σ-$skep$-diagnosis as well.*

Proof. Let $(a,b) \in \mathcal{S}$. By assumption, $F_{\mathcal{S}} \cup \{(a,b)\} = F_{\mathcal{S}\setminus\{(a,b)\}}$ is inconsistent wrt. σ and credulous reasoning and $F_{\mathcal{S}}$ is consistent. By definition this means there is a non-empty extension $\emptyset \neq E \in \sigma(F_{\mathcal{S}})$. We claim that b is skeptically accepted in $F_{\mathcal{S}}$.

Stable: Let E be a stable extension of $F_{\mathcal{S}}$. By definition, $E^{\oplus}_{F_{\mathcal{S}}} = A$ and E is conflict-free in $F_{\mathcal{S}}$. Assume for the sake of contradiction $b \notin E$. We claim that in this case, E must be a stable extension of $F_{\mathcal{S}} \cup \{(a,b)\}$ as well:

- Since E was conflict-free in $F_{\mathcal{S}}$ and $b \notin E$, E is also conflict-free in $F_{\mathcal{S}} \cup \{(a,b)\}$,
- due to $E^{\oplus}_{F_{\mathcal{S}}} = A$ we immediately infer $A = E^{\oplus}_{F_{\mathcal{S}}} \subseteq E^{\oplus}_{F_{\mathcal{S}}\cup\{(a,b)\}}$ and since the other inclusion is clear, we obtain $A = E^{\oplus}_{F_{\mathcal{S}}\cup\{(a,b)\}}$.

The two items above are the properties a stable extension requires. Now, since E is a stable extension of $F_{\mathcal{S}} \cup \{(a,b)\}$, we see that $\mathcal{S}$ is not a *minimal* stb-$cred$-diagnosis of F contradicting our assumption. We thus conclude $b \in E$. Since E was an arbitrary σ-extension of $F_{\mathcal{S}}$, b is skeptically accepted. We thus infer that $\mathcal{S}$ is even a stb-$skep$-diagnosis. Minimality will be discussed below.

Preferred: Now let E be a non-empty preferred extension of $F_{\mathcal{S}}$. Hence $E = \Gamma_{F_{\mathcal{S}}}(E)$ with $E \neq \emptyset$. Again assume $b \notin E$. In this case, b is not defended by E, otherwise we had $b \in \Gamma_{F_{\mathcal{S}}}(E)$. This means, there is an argument $c \in A$ with $(c,b) \in R\setminus\mathcal{S}$ (hence, $c \neq a$) with either $c \in E$ or c is not attacked by E. Now consider $F_{\mathcal{S}} \cup \{(a,b)\}$. Since $b \notin E$, E is still conflict-free. Moreover, there is the argument c as above, so we have $b \notin \Gamma_{F_{\mathcal{S}}\cup\{(a,b)\}}(E)$ as well. This means the additional attack (a,b) is irrelevant for the characteristic function Γ applied to E, i. e.,

$$E = \Gamma_{F_{\mathcal{S}}}(E) = \Gamma_{F_{\mathcal{S}}\cup\{(a,b)\}}(E)$$

and hence, $E \neq \emptyset$ is a complete extension of $F_{\mathcal{S}} \cup \{(a,b)\}$. Thus, $F_{\mathcal{S}} \cup \{(a,b)\}$ is consistent wrt. complete semantics and credulous reasoning, implying it is consistent wrt. preferred semantics and credulous reasoning. As above, this contradicts minimality of $\mathcal{S}$. We thus conclude $b \in E$ for any *non-empty* preferred extension E. Since there is at least one non-empty preferred extension, $\emptyset \notin pr(F_{\mathcal{S}})$ and we thus see again that b is skeptically accepted.

Semi-Stable: As before, assume we are given an non-empty semi-stable extension E of $F_{\mathcal{S}}$ with $b \notin E$. We may argue as above since E is also a preferred extension of $F_{\mathcal{S}}$. We thus infer that E is a complete extension of $F_{\mathcal{S}} \cup \{(a,b)\}$. This is the same contradiction as before, implying $b \in E$. Again, E was an arbitrary non-empty extension and $\emptyset \notin ss(F_{\mathcal{S}})$, so b is skeptically accepted.

Minimality: In all three cases we observed that $\mathcal{S}$ is a σ-$cred$-diagnosis as well. As a final remark we note that $\mathcal{S}$ must be *minimal* since $\mathcal{S}$ was assumed to be minimal for credulous reasoning already. More precisely a proper subset $\mathcal{S}' \subsetneq \mathcal{S}$ cannot be a σ-$skep$-diagnosis which can be seen as in the proof of Theorem 5.2.4. □

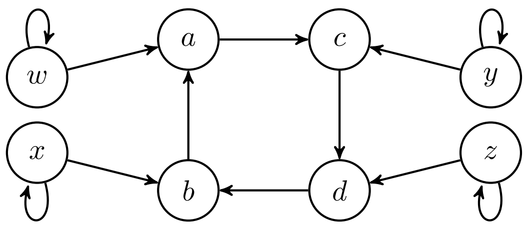
Observe that inconsistency of F wrt. credulous reasoning was a premise of the above theorem. In case F possesses a non-empty σ-extension, it is clear that a minimal σ-$skep$-diagnosis can be found as a superset of the single minimal σ-$cred$-diagnosis $\emptyset$. So we summarize:

Corollary 5.2.9. *Let F be an AF and let $\sigma \in \{ss, pr, stb\}$. If $\mathcal{S}$ is a minimal attack-based σ-$cred$-diagnosis of F, then there is a minimal σ-$skep$-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$.*

Proof. If F is consistent wrt. σ and credulous reasoning, then $\mathcal{S}$ must be empty, so the claim is trivial. Otherwise, if $\mathcal{S}$ is a minimal attack-based σ-$cred$-diagnosis of F, then due to Theorem 5.2.8 we may set $\mathcal{S}' = \mathcal{S}$, i. e., we can even guarantee equality. □

When considering complete semantics, we do *not* have minimality in Theorem 5.2.5 in general. At a first glance, this might be surprising considering the affirmative answer for preferred semantics. However, in the proof of Theorem 5.2.8 we could exclude $\emptyset$ as a possible preferred extension once we found an arbitrary non-empty fixed point of Γ. This does not work for complete semantics. Hence, we find the following counterexample:

Example 5.2.10. Consider the following AF $F = (A, R)$:

F :


Clearly, $\emptyset$ is the only complete extension. The reader may verify that $\mathcal{S} = \{(w, a), (z, d)\}$ is a minimal *co-cred*-diagnosis of F, yielding the complete extension $\{a, d\}$. Since $\emptyset$ is a complete extension of $F_\mathcal{S}$ as well, this is no *co-skep*-diagnosis of F. One may check that $\mathcal{S}$ cannot be extended to a *minimal* skeptical diagnosis: A minimal *co-skep*-diagnosis must ensure that at least one argument is unattacked. Thus, they are given as $\{(w, w)\}, \ldots, \{(z, z)\}$ and $\{(w, a), (b, a)\}, \ldots, \{(z, d), (c, d)\}$. So there is no minimal *co-skep*-diagnoses $\mathcal{S}'$ with $\mathcal{S} \subseteq \mathcal{S}'$.

Finally, we show two helpful, but not unexpected relations between different semantics and their reasoning modes.

Theorem 5.2.11. *Let $F = (A, R)$ be an AF. and let σ and τ, be two semantics with $\sigma \subseteq \tau$ and σ is universally defined.*

- *If $\mathcal{S} \subseteq F$ is a cred-σ-diagnosis of F, then there is a minimal cred-τ-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S}' \subseteq \mathcal{S}$.*
- *If $\mathcal{S} \subseteq F$ is a skep-τ-diagnosis of F, then there is a minimal skep-σ-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S}' \subseteq \mathcal{S}$.*

Proof. We prove the second item only. Let $\mathcal{S}$ be a $skep$-τ-diagnosis of F. This means, $\bigcap \tau(F_\mathcal{S}) \neq \emptyset$. Since $\sigma \subseteq \tau$ is assumed we deduce $\emptyset \neq \bigcap \tau(F_\mathcal{S}) \subseteq \bigcap \sigma(F_\mathcal{S})$. Since σ is universally defined we have $\sigma(F_\mathcal{S}) \neq \emptyset$ which implies that $\mathcal{S}$ is a $skep$-σ-diagnosis of F. Moreover, by finiteness of $\mathcal{S}$ we deduce the existence of a minimal $skep$-σ-diagnosis $\mathcal{S}'$ of F with $\mathcal{S}' \subseteq \mathcal{S}$ concluding the proof. □

5.2.3 Uniquely Defined Semantics

We now focus on uniquely defined semantics, i. e., we have $|\sigma(F)| = 1$ for any $F \in \mathcal{F}$. Considering the semantics we investigate in this chapter this means $\sigma \in \{gr, eg, il\}$.

Note that in case of uniquely defined semantics we have that any (minimal) skeptical diagnosis is a (minimal) credulous one and vice versa.

Lemma 5.2.12. *If F is an AF and $\sigma \in \{gr, eg, il\}$, then $\mathcal{S}$ is a cred-σ-diagnosis of F iff it is a skep-σ-diagnosis of F.*

Proof. If $|\sigma(F)| = 1$, then $\bigcap \sigma(F) = \bigcup \sigma(F)$. □

This implies in particular that minimality in Theorem 5.2.5 can be shown.

Corollary 5.2.13. *Given an AF F and a semantics $\sigma \in \{gr, eg, il\}$. If $\mathcal{S}$ is a minimal cred-σ-diagnosis of F, then there is a minimal skep-σ-diagnosis $\mathcal{S}'$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$.*

Proof. Set $\mathcal{S}' = \mathcal{S}$ and apply Lemma 5.2.12. □

We proceed with grounded semantics since these results will play a central role for all other semantics considered in this chapter. Dung originally defined the grounded extension of an AF $F = (A, R)$ as the $\subseteq$-least fixed point of the characteristic function $\Gamma_F : 2^A \to 2^A$ with $E \mapsto \{a \in A \mid a \text{ is defended by } E\}$. Moreover, he showed that this definition coincides with the $\subseteq$-least complete extension [49, Theorem 25] as introduced in Definition 5.1.1. Since Γ_F is shown to be $\subseteq$-monotonic we may compute the unique grounded extension G stepwise, i.e., applying Γ_F iteratively starting from the empty set. More precisely, $G = \bigcup_{i=1}^{|A|} \Gamma_F^i(\emptyset)$ (cf. [21, Section 3.2]). Consequently, an AF possesses a non-empty grounded extension if and only if there exists at least one unattacked argument. This renders argument-based diagnoses weaker in some cases since there is no way to remove a single argument. It is thus clear that a self-controversial AF does not possess an argument-based diagnosis. More precisely, we find the following:

Fact 5.2.14. Let $F = (A, R)$ be an AF. Let $\sigma = gr$.

- There exists a minimal argument-based gr-repair for F iff F is not self-controversial.
- There exists a minimal attack-based gr-repair for F.

Proof. For the first item, assume $F = (A, R)$ and $a \in A$ does not attack itself. Then, $\mathcal{S} = A \setminus \{a\}$ is a gr-diagnosis of F. Due to finiteness, we find a minimal one $\mathcal{S}'$ with $\mathcal{S}' \subseteq \mathcal{S}$. For the second item observe that $\mathcal{S} = R$ is a diagnosis. □

In addition to Fact 5.2.14, we observe that diagnoses for ideal and eager semantics can be found as subsets of a grounded diagnosis. The intuitive reason is the fact that ideal semantics accepts more arguments than grounded semantics and eager semantics is even more credulous than ideal semantics.

Lemma 5.2.15. *Let $F = (A, R)$ be an AF. Let $\sigma \in \{il, eg\}$. If $\mathcal{S}'$ is a gr-diagnosis of F, then there is a minimal σ-diagnosis $\mathcal{S}$ of F, st. $\mathcal{S} \subseteq \mathcal{S}'$.*

Proof. As already mentioned, $\mathcal{S}'$ is a gr-diagnosis of F if and only if $F_{\mathcal{S}'}$ contains an unattacked argument, say a. In this case we see that a also occurs in the unique ideal as well as eager extension. Hence, $\mathcal{S}'$ is a σ-diagnosis for $\sigma \in \{il, eg\}$. Due to finiteness, there is a minimal σ-diagnosis $\mathcal{S}$ with $\mathcal{S} \subseteq \mathcal{S}'$. □

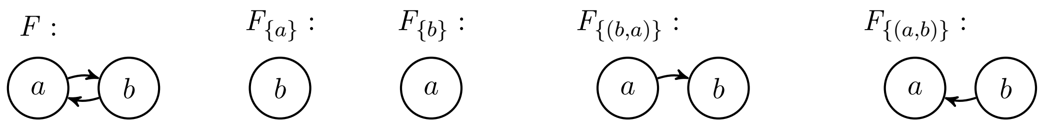
The subsequent theorem claims the existence of minimal σ-diagnoses for the considered uniquely defined semantics (recall that we do not need to distinguish between credulous and skeptical reasoning here). Due to the above fact, we find the claim for grounded semantics. Equipped with a grounded diagnosis, we find the other two by applying Lemma 5.2.15. Moreover, the restriction to finite AFs even gives us the existence of minimal ones.

Theorem 5.2.16. *Let $\sigma \in \{gr, eg, il\}$. Let F be an AF.*

- *There exists a minimal argument-based σ-repair for F iff F is not self-controversial.*
- *There exists a minimal attack-based σ-repair for F.*

Proof. For $\sigma = gr$, this is Fact 5.2.14. For $\sigma \in \{eg, il\}$, this can be inferred from Lemma 5.2.15 after applying Fact 5.2.14. □

Example 5.2.17. The following simple framework F demonstrates that least σ-repairs do not necessarily exist. For $\sigma \in \{gr, eg, il\}$ we have $\sigma(F) = \{\emptyset\}$, i. e., nothing is credulously or skeptically accepted.



Observe that all four depicted given diagnoses $\{a\}$, $\{b\}$, $\{(a, b)\}$ and $\{(b, a)\}$ are minimal. This example thus illustrates that a least repair does not necessarily exist. This is true for both argument-based as well as attack-based diagnoses.

This finishes our discussion on uniquely defined semantics. In the subsequent section, we turn to universally defined semantics.

5.2.4 Universally Defined Semantics

Let us consider now semantics which provide us with at least one acceptable position. The following lemma shows that for these semantics minimal credulous as well as skeptical diagnoses are guaranteed, whenever there is a grounded diagnosis.

Lemma 5.2.18. *Let $\sigma \in \{ss, pr, co\}$. For any AF $F = (A, R)$ there exists a minimal σ-diagnosis $\mathcal{S}$, whenever there exists a gr-diagnosis $\mathcal{S}'$ of F. Moreover, even $\mathcal{S} \subseteq \mathcal{S}'$ can be guaranteed.*

Proof. Let $\sigma \in \{ss, pr, co\}$ and $\mathcal{S}'$ a gr-diagnosis of F. Hence, $gr(F_{\mathcal{S}'}) = \{G\}$ with $G \neq \emptyset$. Since G is the $\subseteq$-least fixpoint of $\Gamma_{F_{\mathcal{S}'}}$ we deduce $G \subseteq C$ for any $C \in co(F_{\mathcal{S}'})$. Due to $ss \subseteq pr \subseteq co$ and the universal definedness of σ we have $\emptyset \neq G \subseteq \bigcap \sigma(F_{\mathcal{S}'})$ as well as $\emptyset \neq G \subseteq \bigcup \sigma(F_{\mathcal{S}'})$. Hence, $\mathcal{S}'$ is a skeptical as well as credulous σ-diagnosis of F. Due to finiteness of F, there exists a minimal σ-diagnosis $\mathcal{S} \subseteq \mathcal{S}'$ concluding the proof. □

Combining Theorem 5.2.16 and Lemma 5.2.18 yields the subsequent main theorem for the considered universally defined semantics. As usual, there is a slight difference between argument-based and attack-based diagnoses.

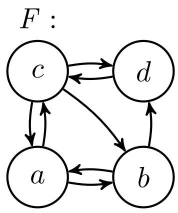
Theorem 5.2.19. *Let $\sigma \in \{ss, pr, co\}$. Let F be an AF.*

- *There exists a minimal argument-based σ-repair for F iff F is not self-controversial.*
- *There exists a minimal attack-based σ-repair for F.*

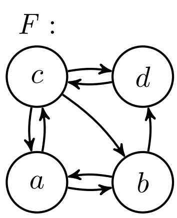
Proof. Apply Theorem 5.2.16 and Lemma 5.2.18. □

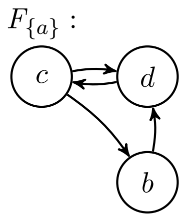
The following example shows, as promised in Lemma 5.2.18, that already computed grounded diagnoses can be used to find minimal preferred diagnoses. Moreover, in contrast to uniquely defined semantics we observe that minimal skeptical and minimal credulous diagnoses do not necessarily coincide. This is the case for both argument-based and attack-based diagnoses.

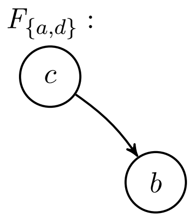
Example 5.2.20. Consider the following AF F. Since we have no unattacked arguments we deduce $gr(F) = \{\emptyset\}$, i.e., nothing is accepted.



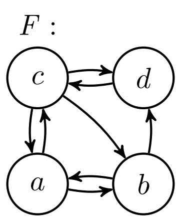
Argument-based repairs: Observe that $F_{\{a\}}$ and $F_{\{d\}}$ do not possess a grounded extension, either. Since $gr(F_{\{a,d\}}) = \{\{c\}\}$, $F_{\{a,d\}}$ is a minimal argument-based gr-repair. Note that $\{a, d\}$ is even a skeptical as well credulous preferred diagnosis of F. These diagnoses are not minimal for preferred semantics since $pr(F) = \{\{a, d\}, \{c\}\}$ implies $\bigcup pr(F) \neq \emptyset$ as well as $pr(F_{\{a\}}) = \{\{c\}\}$ entails $\bigcap pr(F_{\{a\}}) \neq \emptyset$. Altogether, we have strict subset relation ($\emptyset \subsetneq \{a\} \subsetneq \{a, d\}$) between minimal credulous preferred, minimal skeptical preferred and minimal grounded diagnoses.

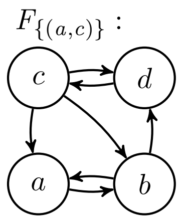


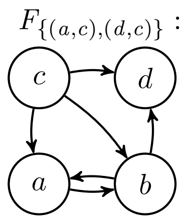




Attack-based reparis: Regarding attack-based diagnoses, we make similar observations for the chain $\emptyset \subsetneq \{(a, c)\} \subsetneq \{(a, c), (d, c)\}$.







5.2.5 Collapsing Semantics

Stable semantics is the only prominent semantics which may collapse even for finite AFs, i. e., there are AFs which do not possess any stable extension. However, in terms of existence of repairs we do not observe any differences to all other considered semantics.

Fact 5.2.21. Let F be an AF.

- There exists a minimal argument-based stb-repair for F iff F is not self-controversial.
- There exists a minimal attack-based stb-repair for F.

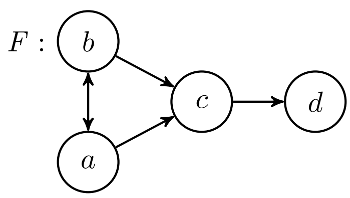
In contrast to all other considered semantics we have that stable diagnoses can not be necessarily found as subsets of an already computed grounded one (Lemmata 5.2.15 and 5.2.18). For instance, the AF F from Example 5.2.1 possesses the unique grounded extension $\{e, f\}$. Consequently, we have the trivial (least) gr-diagnosis, namely the empty set. As F does not possess a stable extension, all minimal stb-diagnoses are non-empty. Nevertheless, credulous as well as skeptical diagnoses for stable semantics can be found as supersets of grounded ones.

Lemma 5.2.22. *Let $F = (A, R)$ be an AF. If $\mathcal{S}'$ is a gr-diagnosis of F, then there is a stb-diagnosis $\mathcal{S}$ of F, st. $\mathcal{S}' \subseteq \mathcal{S}$.*

Proof. Argument-based: Given $\mathcal{S}'$ as gr-diagnosis of $F = (A, R)$, i.e., $gr\left(F_{\mathcal{S}'}\right) = \{E\}$ with $E \neq \emptyset$. Consider now $E^{\oplus}$ wrt. the attack-relation of $F_{\mathcal{S}'}$. Then $\mathcal{S}' \subseteq A \setminus E^{\oplus}$ and moreover, $gr\left(F_{A \setminus E^{\oplus}}\right) = \{E\}$. Obviously, by construction $E \in stb\left(F_{A \setminus E^{\oplus}}\right)$. Furthermore, since E is non-empty we deduce that there is at least one unattacked argument $a \in E$. Hence, for any $E' \in stb\left(F_{A \setminus E^{\oplus}}\right)$ we have $a \in E'$. Consequently, $A \setminus E^{\oplus}$ serves as a credulous as well as skeptical diagnosis for stable semantics.
Attack-based: Trivial since we may set $\mathcal{S} = R$. □

Note that Lemma 5.2.22 does not claim minimality of the stb-diagnosis $\mathcal{S}$. Indeed, the following example illustrates that existence of a *minimal* stb-diagnosis with $\mathcal{S}' \subseteq \mathcal{S}$ as above is not obtained in general.

Example 5.2.23. Consider the following AF F



Since every argument is attacked we infer that $\emptyset$ cannot be a gr-diagnosis. Consider now the gr-diagnosis $\{a\}$. Indeed, this is also a stb-diagnosis (wrt. credulous as well as skeptical reasoning), but not minimal since F itself possesses the skeptically accepted argument d. Moreover we make the same observations for the attack-based gr-diagnosis $\{(a, b)\}$. Hence, this is a counterexample for both types of diagnoses.

5.2.6 Summary

This finishes our discussion regarding rather general results pertaining to the existence and relationships of repairs. We saw that as long as one is not interested in stable semantics, one can find a minimal diagnosis as a subset of a given grounded one (see Lemmata 5.2.15 and 5.2.18). In any case, computing a gr-repair is a reasonable starting point in order to reduce the search space. In view of this, it is interesting to note that computing gr-repairs is tractable (see Proposition 6.4.4 in Section 6.4 below) and both reasoning modes coincide. Our discussion in Section 6.4 also confirms the intuition that skeptical repairs tend to be more demanding from a computational point of view. Hence if one is interested in skeptical diagnoses, then the cases covered by Theorem 5.2.8 are the most well-behaved ones: We can start by computing a minimal *cred*-diagnosis and then refine it until a minimal *skep*-diagnosis is found. Recall that this theorem is applicable when we are looking for an attack-based σ-*skep*-diagnosis with $\sigma \in \{ss, pr, stb\}$. The following scheme depicts a summary of these results:

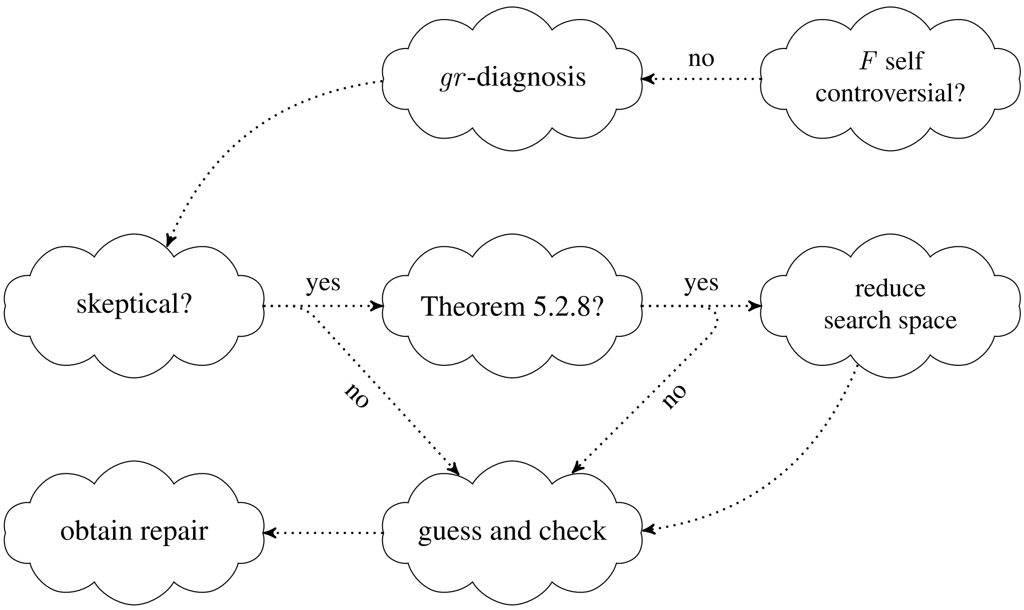


Figure 5.3: Finding minimal repairs for $\sigma \neq stb$: We start by computing a gr-repair. This helps reducing the search space. If we are interested in skeptical repairs and Theorem 5.2.8 is applicable, it might be helpful to compute credulous repairs first.

5.3 A Hitting Set Duality for Argumentation Frameworks

Let us now characterize all repairs of a given AF, independent of considered semantics, reasoning mode or type of repairs (argument-based or attack-based). Our goal is to apply our hitting set duality from Theorem 3.1.12. Recall the setting from the previous chapters, where we considered AFs only with stable semantics and credulous reasoning. We left A implicit and then a knowledge base was a set of attacks. Hence given an AF $F = (A, R)$ a strongly inconsistent subset (of attacks) was a set $\mathcal{H} \subseteq R$ such that each AF $F = (A, \mathcal{H}')$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq R$ does not possess a stable extension. We need to rephrase the notion of strong inconsistency for our more comprehensive treatment of AFs.

Definition 5.3.1. Let $F = (A, R)$ be an AF and let σ be any semantics.
Argument-based: We call $\mathcal{H} \subseteq A$ a *strongly inconsistent* set of arguments of F wrt. σ and credulous (skeptical) reasoning if for each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq A$ the AF

$$F_{A\setminus\mathcal{H}'} = (\mathcal{H}', R_{|\mathcal{H}'})$$

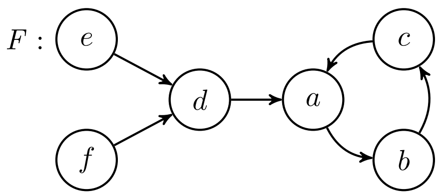
is inconsistent wrt. σ and credulous (skeptical) reasoning. Denote by $SI^A_{min}(F, \sigma, cred)$ ($SI^A_{min}(F, \sigma, skep)$) the set of all minimal strongly inconsistent sets of arguments of F wrt. σ and credulous (skeptical) reasoning.
Attack-based: We call $\mathcal{H} \subseteq R$ a *strongly inconsistent* set of attacks of F wrt. σ and credulous (skeptical) reasoning if for each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq R$ the AF

$$F_{R\setminus\mathcal{H}'} = (A, \mathcal{H}')$$

is inconsistent wrt. σ and credulous (skeptical) reasoning. Denote by $SI^R_{min}(F, \sigma, cred)$ ($SI^R_{min}(F, \sigma, skep)$) the set of all minimal strongly inconsistent sets of attacks of F wrt. σ and credulous (skeptical) reasoning.

As usual the semantics as well as the reasoning mode will sometimes be clear from the context or irrelevant. In this case we will leave them implicit and simply write $SI^A_{min}(F)$ resp. $SI^R_{min}(F)$.

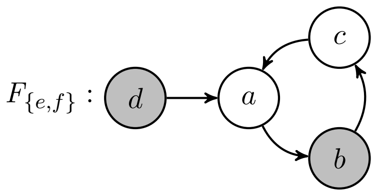
Example 5.3.2. Consider again the AF $F = (A, R)$ with $A = \{a, b, c, d, e, f\}$ and the attacks $R = \{(a, b), (b, c), (c, a), (d, a), (e, d), (f, d)\}$:



Argument-based: Let us focus on credulous reasoning. We already observed that F has no stable extension, i.e., A itself is a strongly inconsistent set of arguments wrt. stable semantics. The subset $\mathcal{H}_1 \subseteq A$ with $\mathcal{H}_1 = \{a, b, c\}$ induces the inconsistent AF

$$F_1 = (\{a, b, c\}, \{(a, b), (b, c), (c, a)\})$$

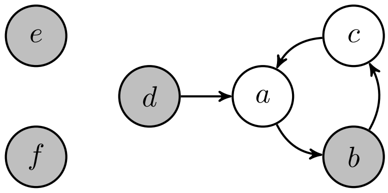
corresponding to the odd circle contained in F. However, $\mathcal{H}_1$ is not a strongly inconsistent set of arguments since the framework induced by $\mathcal{H}_1 \subseteq \mathcal{H}_2$ with $\mathcal{H}_2 = \{a, b, c, d\}$ corresponds to $F_{\{e,f\}}$ which has the stable extension $\{b, d\}$:



One may easily verify that $SI^A_{min}(\mathcal{K}) = \{\{a, b, c, e\},\ \{a, b, c, f\}\}$.

Attack-based: We have $R = \{(a,b),\ (b,c),\ (c,a),\ (d,a),\ (e,d),\ (f,d)\}$. Again consider credulous reasoning and stable semantics. Now $\mathcal{H}_1 = \{(a,b),\ (b,c),\ (c,a)\}$ induces the odd circle contained in F. Again, $\mathcal{H}_1$ is not strongly inconsistent since the superset $\mathcal{H}_2$ with $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq R$ given as $\mathcal{H}_2 = \{(a,b),\ (b,c),\ (c,a),\ (d,a)\}$ induces the consistent AF $F_{R\setminus\mathcal{H}_2} = F_{\{(e,d),(f,d)\}}$ with the stable extension $\{b,d,e,f\}$. Note however that the represented framework is now different, namely F with two attacks removed:

$F_{\{(e,d),(f,d)\}}$:



We obtain

$$SI^R_{min}(F) = \{\{(a,b),\ (b,c),\ (c,a),\ (e,d)\},\ \{(a,b),\ (b,c),\ (c,a),\ (f,d)\}\}.$$

Now we are ready to apply Theorem 3.1.12 to diagnoses of AFs. The result holds for all semantics and both reasoning modes. It can be stated for both argument-based as well as attack-based repairs. To prove this, we proceed as in Example 2.5.5 where we modeled AFs as set of attacks: We construct a logic where any formula α of a knowledge base $\mathcal{K}$ corresponds to an argument (for argument-based diagnoses) resp. attack (for attack-based diagnoses). Since an AF is a tuple consisting of both arguments and attacks, one of them will be fixed. The logic under consideration will also depend on the semantics as well as the reasoning mode.

Proposition 5.3.3 (Duality: argument-based)**.** *Let $F = (A, R)$ be an AF. Let σ be any semantics and consider any reasoning mode. Then $\mathcal{S}$ is a minimal hitting set of $SI^A_{min}(F)$ if and only if $\mathcal{S}$ is a minimal σ-diagnosis of F.*

Proof. Assume the set R of attacks is fixed. Define a logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ with $\mathcal{WF} = A$ (that is, a knowledge base $\mathcal{K}$ is a finite set of arguments in A), $\mathcal{BS} = A$, $\mathcal{INC} = \emptyset$ and $\mathcal{ACC}(\mathcal{K}) = \bigcup\sigma(F)$ resp. $\mathcal{ACC}(\mathcal{K}) = \bigcap\sigma(F)$ where F is the AF given as $F = (\mathcal{K}, R_{|\mathcal{K}})$. Now given an AF $F = (A, R)$ a minimal argument-based repair corresponds to a maximal consistent subset $\mathcal{H} \subseteq \mathcal{K} = A$ and a minimal strongly inconsistent set of arguments of F corresponds to a minimal strongly inconsistent subset of $\mathcal{K}$. Thus, the claim can be seen by applying Theorem 3.1.12 to the logic L we defined here. □

Proposition 5.3.4 (Duality: attack-based)**.** *Let $F = (A, R)$ be an AF. Let σ be any semantics and consider any reasoning mode. Then $\mathcal{S}$ is a minimal hitting set of $SI^R_{min}(F)$ if and only if $\mathcal{S}$ is a minimal σ-diagnosis of F.*

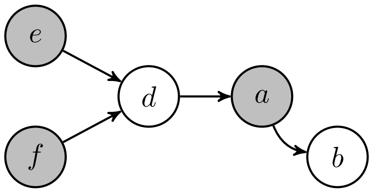
Proof. Assume the set A of arguments is fixed. Define a logic $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ with $\mathcal{WF} = A \times A$ (that is, a knowledge base $\mathcal{K}$ is a finite set of attacks over A), $\mathcal{BS} = A$, $\mathcal{INC} = \emptyset$ and $\mathcal{ACC}(\mathcal{K}) = \bigcup\sigma(F)$ resp. $\mathcal{ACC}(\mathcal{K}) = \bigcap\sigma(F)$ where F is the AF given as $F = (A, \mathcal{K})$. Now given an AF $F = (A, R)$ a minimal attack-based repair corresponds to a maximal consistent subset $\mathcal{H} \subseteq \mathcal{K} = R$ and a minimal strongly inconsistent set of attacks of F corresponds to a minimal strongly inconsistent subset of $\mathcal{K}$. Thus, the claim can be seen by applying Theorem 3.1.12 to the logic L we defined here. □

Example 5.3.5. Consider again our AF F with stable semantics and credulous reasoning. *Argument-based*: For the argument-based diagnoses we checked that

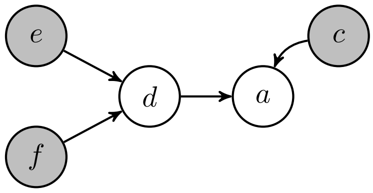
$$SI^A_{min}(\mathcal{K}) = \{\{a, b, c, e\}, \{a, b, c, f\}\}.$$

We obtain four minimal hitting sets, namely $\{a\}$, $\{b\}$, $\{c\}$ and $\{e, f\}$. Observe that $\{e, f\}$ is the already found stb-diagnosis presented in Example 5.3.2. The minimal hitting sets for F can be interpreted as follows: Either one argument from the odd circle needs to be removed or both e and f to facilitate d. These sets correspond to the stb-repairs $F_{\{e,f\}}$ (considered in Example 5.3.2) as well as $F_{\{c\}}$, $F_{\{b\}}$ and $F_{\{a\}}$ depicted below.

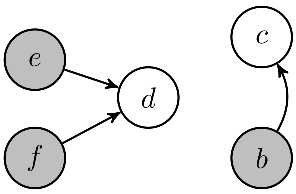
$F_{\{c\}}$:



$F_{\{b\}}$:



$F_{\{a\}}$:

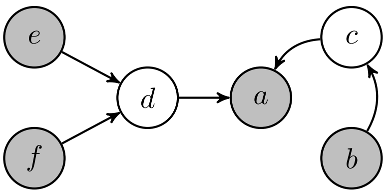


Attack-based: Recall

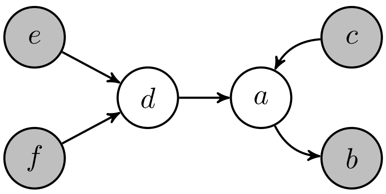
$$SI^R_{min}(F) = \{\{(a, b),\ (b, c),\ (c, a),\ (e, d)\},\ \{(a, b),\ (b, c),\ (c, a),\ (f, d)\}\}.$$

for the attack-based representation. We find the hitting sets $\{(a, b)\}$, $\{(b, c)\}$, $\{(c, a)\}$ as well as $\{(e, d),\ (f, d)\}$. The latter one yields $F_{\{(e,d),(f,d)\}}$ from Example 5.3.2. The former ones correspond to the following AFs:

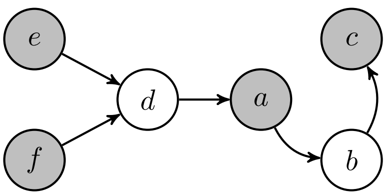
$F_{\{(a,b)\}}$:



$F_{\{(b,c)\}}$:



$F_{\{(c,a)\}}$:



This finishes our discussion regarding a characterization of all diagnoses of a given AF. The results of this demonstrate indicate that finding repairs can be achieved by consideration of the strongly inconsistent arguments resp. attacks. This might not be the most efficient approach if one is just interested in a single diagnosis, but it helps representing all repairs in a concise way. In the next section, we focus on particular aspects and properties of AFs in order to infer tailored, more advanced properties.

5.4 Special Cases For Argumentation Frameworks

We may obtain more insightful results when focusing on certain aspects of the AF under consideration. We will start our investigation with symmetric AFs [43]. In a nutshell, an AF is symmetric if the attack relation is. Symmetry is a rather strong property, yielding a variety of additional connections between repairs and diagnoses. The same is true for compact [24] and acyclic AFs. We then continue with a splitting method introduced in [12]. Splitting methods are an important concept in non-monotonic reasoning which allow for a certain modularization of the knowledge base under consideration which is usually not given due to non-monotonic interactions between formulas (see [11] for an excellent overview). The structural properties that come along with splitting can be utilized to infer results for diagnoses. The last part of this section will be devoted to *infinite* AFs [21]. Allowing the underlying set of arguments to be infinite possesses additional challenges since the behavior of an infinite AF is less intuitive and *existence* and *uniqueness* of certain extensions as well as diagnoses is no longer guaranteed.

5.4.1 Symmetric, Compact and Acyclic Frameworks

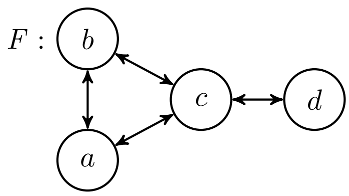
Let us start with so-called *symmetric* AFs. According to [43] an AF $F = (A, R)$ is symmetric if R is symmetric, nonempty and irreflexive.

Definition 5.4.1. Let $F = (A, R)$ be an AF. We call F *symmetric* if

- $R \neq \emptyset$,
- $(a, b) \in R$ implies $(b, a) \in R$ for each $a, b \in A$,
- $(a, a) \notin R$ for each $a \in A$.

For example, the following AF F is symmetric:

Example 5.4.2. Assume $F = (A, R)$ is as depicted below:



The three items of Definition 5.4.1 are straightforward to verify.

Now assume we are given a symmetric AF F. Clearly, when moving from F to $F_{\mathcal{S}}$ for $\mathcal{S} \subseteq A$ resp. $\mathcal{S} \subseteq R$, we do not change the fact that the attack relation is irreflexive. We might end up with an AF with no attack (violating non-emptiness), but this does not concern us since an AF of the form $F = (A, \emptyset)$ is consistent in almost all considered cases; the only exception is $\sigma = ad$ with skeptical reasoning. However, we need to make sure that the symmetry of the attack relation is preserved, otherwise we lose the properties we want to utilize. This is clearly no issue for argument-based diagnoses:

Lemma 5.4.3. *If $F = (A, R)$ is symmetric and $\mathcal{S} \subseteq A$, then the attack-relation of $F_{\mathcal{S}}$ is empty or symmetric and irreflexive.*

Of course, we want to consider attack-based diagnoses as well. Fortunately, the restriction we need to make to removal of attacks is quite natural. We simply need to ensure that the diagnosis operates symmetric in the sense that removal of (a, b) implies removal of (b, a) as well. Formally:

Definition 5.4.4. Let $F = (A, R)$ be a symmetric AF. An attack-based diagnosis $\mathcal{S} \subseteq R$ of F is *symmetric* if $(a, b) \in \mathcal{S} \Leftrightarrow (b, a) \in \mathcal{S}$.

Clearly, we now have:

Lemma 5.4.5. *If $F = (A, R)$ is symmetric and $\mathcal{S} \subseteq R$ a symmetric diagnosis, then the attack relation of $F_{\mathcal{S}}$ is empty or symmetric and irreflexive.*

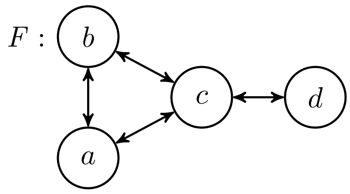
In case of stable, semi-stable and preferred semantics we obtain a very useful property, namely any argument $a \in A$ belongs to at least one extension [43, Proposition 6]. Consequently, we may show the following properties.

Proposition 5.4.6. *Let $F = (A, R)$ be a symmetric AF and $\sigma \in \{stb, ss, pr\}$.*

- *$\emptyset$ is the least $cred$-σ-diagnosis and*
- *$\mathcal{S} \subseteq F$ is a (minimal) symmetric $skep$-σ-diagnosis iff $\mathcal{S}$ is a (minimal) symmetric gr-diagnosis.*

Let us assume that our current knowledge base underlies further external revision processes (cf. [18; 44; 47] for belief revision in abstract argumentation). Both items can be gainfully used if we know that certain types of revision do not affect the symmetry of an AF. More precisely, the items 1 and 2 ensure that we have either nothing to do (if interested in credulous reasoning) or we may act according to grounded semantics instead of σ (if skeptical reasoning is chosen).

Example 5.4.7. Consider again the symmetric AF F from above:



We have $stb(F) = \{\{a, d\}, \{b, d\}, \{c\}\}$. This means, no argument is skeptically accepted. In order to repair regarding grounded semantics we have to ensure the existence of at least one unattacked argument. Consequently, the least argument-based $skep$-gr-repair is given as $F_{\{c\}}$. As promised by the second item in Proposition 5.4.6 this indeed coincides with the least $skep$-stb-repair.

Let us briefly consider two further classes of frameworks, namely so-called *compact* and *acyclic* ones. The first one is semantically defined and characterized by the feature that each argument of the AF occurs in at least one extension of the AF [15; 24]. For instance, the AF F depicted in Example 5.4.7 is compact wrt. stable semantics. Compact frameworks obviously fulfill the first item of Proposition 5.4.6 and thus build an interesting subclass of AFs if interested in credulous reasoning. The second class is syntactically defined and as expected: An AF is acyclic if it does not contain any cycles. Such frameworks are known to be *well-founded* [49] which means, they possess exactly one complete extension which is grounded, preferred and stable [43, Propositions 1 and 2]. This means, the agent is able to act (in both reasoning modes) whenever we are faced with an acyclic AF.

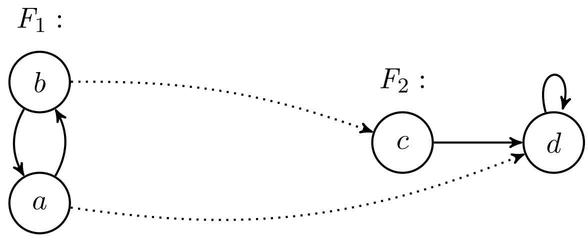
5.4.2 Splitting

Let us now investigate situations where we are given a *splitting* of the AF under consideration. Splitting is an important concept in non-monotonic reasoning as it abuses structural properties of a knowledge base in order to identify a certain monotonic behavior. More precisely, splitting methods try to divide a theory in subtheories such that the formal semantics of the entire theory can be obtained by constructing the semantics of the subtheories. For AFs, splitting was considered in [11; 12; 14]. We briefly recall the required notions here and then demonstrate how to infer properties of repairs and diagnoses.

Definition 5.4.8. Let $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$ be two AFs with $A_1 \cap A_2 = \emptyset$. Let $R_3 \subseteq A_1 \times A_2$. We call (F_1, F_2, R_3) a *splitting* of the AF $F = (A_1 \cup A_2, R_1 \cup R_2 \cup R_3)$.

In a nutshell, if (F_1, F_2, R_3) is a splitting of F, then extensions of F_1 can be computed as a first step to find those of F. The AF F_2 does not influence F_1 and can be considered later.

Example 5.4.9. Let $F_1 = (A_1, R_1)$ with $A_1 = \{a, b\}$ and $R_1 = \{(a, b), (b, a)\}$ as well as $F_2 = (A_2, R_2)$ with $A_2 = \{c, d\}$ and $R_2 = \{(c, d), (d, d)\}$, and let $R_3 = \{(a, d), (b, c)\}$. Then, (F_1, F_2, R_3) is a splitting of the following AF:

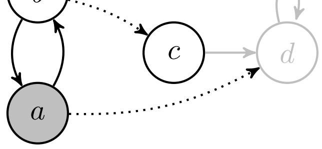


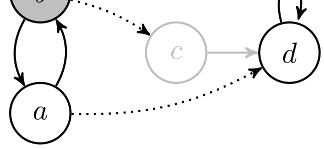
The idea of splitting is as follows: Once we are given an extension E_1 of the AF F_1, based on E_1 we want to construct a reduced version of F_2. Then we compute an extension E_2 of this reduced AF to obtain an extension $E_1 \cup E_2$ of F. In the following, we define how to reduce F_2 for stable semantics. The other semantics will be discussed afterwards.

Definition 5.4.10 (Reduct)**.** Let $F_2 = (A_2, R_2)$ be an AF and A_1 such that $A_1 \cap A_2 = \emptyset$. Let $S \subseteq A_1$ and $L \subseteq A_1 \times A_2$. The (S, L)-reduct of F_2, denoted by $F_2^{S,L}$, is the AF

$$F_2^{S,L} = \left(A_2^{S,L}, R_2^{S,L}\right) \quad \text{with} \quad A_2^{S,L} = \{a_2 \in A_2 \mid \nexists a_1 \in S : (a_1, a_2) \in L\}$$
$$R_2^{S,L} = \left\{(a, b) \in R_2 \mid a, b \in A_2^{S,L}\right\}.$$

Example 5.4.11. Consider again our previous example. The AF $F_1 = (A_1, R_1)$ as above has two stable extensions $E_1 = \{a\}$ and $E_1' = \{b\}$. So we are interested in the (E_1, R_3)- and (E_1', R_3)-reduct of F_2 which are $F_2^{E_1,R_3} = (\{c\}, \emptyset)$ and $F_2^{E_1',R_3} = (\{d\}, \{(d, d)\})$:





The former has the stable extension $E_2 = \{c\}$, the latter none. Indeed, the unique stable extension of the whole AF F is $\{a, c\} = E_1 \cup E_2$.

Now the following theorem states that we can indeed find extensions E of F by considering an extension E_1 of F_1 and then reduce F_2 and continue computing. More precisely, we find all extensions of F this way:

Theorem 5.4.12 ([12]). *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$.*

- *If E_1 is a stable extension of F_1 and E_2 a stable extension of the (E_1, R_3)-reduct of F_2, then $E_1 \cup E_2$ is a stable extension of F.*
- *Vice versa, if E is a stable extension of F, then $E_1 = E \cap A_1$ is a stable extension of F_1 and $E_2 = E \cap A_2$ a stable extension of the (E_1, R_3)-reduct of F_2.*

We can utilize this in order to find properties of repairs. Each stable extension of F contains a stable extension of F_1, so the latter is required to be consistent wrt. credulous reasoning. This means when trying to find repairs for F, one may start with repairs of F_1. In the following, we show that one can extend minimal repairs of F_1 to minimal repairs of F.

Proposition 5.4.13. *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$. If $\mathcal{S}_1$ is a minimal stb-$cred$-diagnosis of F_1, then there is a minimal stb-$cred$-diagnosis $\mathcal{S}$ of F with $\mathcal{S}_1 \subseteq \mathcal{S}$.*

Proof. Let $\mathcal{S}_1$ be a minimal stb-$cred$-diagnosis of F_1. Let E_1 be a stable extension of $(F_1)_{\mathcal{S}_1}$. Now consider $F_2^{E_1,R_3}$, i. e., the (E_1, R_3)-reduct of F_2. If $F_2^{E_1,R_3}$ possesses a stable extension, then we are done. If this is not the case, we need to be careful since two different extensions of $(F_1)_{\mathcal{S}_1}$ might induce two reducts where the minimal repairs are in a subset relation; so we cannot just take the minimal repair of $F_2^{E_1,R_3}$. So assume for the moment there is an extension of $(F_1)_{\mathcal{S}_1}$ such that the reduct is not self-controversial and let

$$\mathcal{S}_2 \in \min_{E_1 \in stb\left((F_1)_{\mathcal{S}_1}\right)} \left\{ \mathcal{S} \mid \mathcal{S} \text{ is a minimal repair of } F_2^{E_1,R_3} \right\}$$

where we observe straightforwardly that the minimum exists since we are dealing with finite AFs. Now let $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$. We claim that $\mathcal{S}$ is a minimal diagnosis of F. For this, we observe that we cannot remove any element from $\mathcal{S}_1$ since this was assumed to be a minimal diagnosis of F_1 and from Theorem 5.4.12 we know that it needs to possess a stable extension. Moreover, if we are given $\mathcal{S}_2$ and the extension E_1 in which the minimum is attained, we see that $\left(F_2^{E_1,R_3}\right)_{\mathcal{S}_2}$ possesses a stable extension ensuring that $\mathcal{S}$ is a diagnosis. Now minimality is due to construction of $\mathcal{S}_2$. Finally, if the reduct is self-controversial for each extension of $(F_1)_{\mathcal{S}_1}$, we can set $\mathcal{S}_2 = A_2$ for argument-based and $\mathcal{S}_2 = R_2$ for attack-based diagnoses. □

Although being rather simple, the most important observation in the previous proposition was that F_1 needs to be consistent in order for F to be consistent. We can also phrase this observation in terms of inconsistency, i. e., the sets $SI^A_{min}(F)$ and $SI^R_{min}(F)$. The following proposition formalizes that strongly inconsistent subsets of F_1 are also strongly inconsistent for F, at least for credulous reasoning. This holds for sets of attacks as well as sets of arguments.

Proposition 5.4.14. *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$.*

- $SI^A_{min}(F_1, stb, cred) \subseteq SI^A_{min}(F, stb, cred)$,
- $SI^R_{min}(F_1, stb, cred) \subseteq SI^R_{min}(F, stb, cred)$.

Proof. We prove the first item. The second one is similar. We simply write $SI^A_{min}(F_1)$ instead of $SI^A_{min}(F_1, stb, cred)$. Assume $\mathcal{H}_1 \in SI^A_{min}(F_1)$. Then, for any set $\mathcal{H}_1'$ with $\mathcal{H}_1 \subseteq \mathcal{H}_1' \subseteq A_1$, the AF $\left(\mathcal{H}_1', (R_1)_{|\mathcal{H}_1'}\right)$ has no stable extension. Due to Theorem 5.4.12, this implies $\left(\mathcal{H}_1', (R_1)_{|\mathcal{H}_1'}\right) \cup F_2 \cup R_3$ has no stable extension, either. Since we can also apply the splitting theorem after moving to a subframework of F_2, we see that the AF

$$\left(\mathcal{H}_1', (R_1)_{|\mathcal{H}_1'}\right) \cup \left(\mathcal{H}_2', (R_2)_{|\mathcal{H}_2'}\right) \cup R_3$$

is inconsistent for any $\mathcal{H}_1'$ with $\mathcal{H}_1 \subseteq \mathcal{H}_1' \subseteq A_1$ and any $\mathcal{H}_2'$ with $\mathcal{H}_2' \subseteq A_2$. Thus, $\mathcal{H}$ is a strongly inconsistent sets of arguments of F. Minimality can be inferred from the splitting theorem in a similar way. Hence, $\mathcal{H}_1 \in SI^A_{min}(F)$. □

Let us now assume we are given $SI^A_{min}(F_1)$. Due to the hitting set duality for this setting, i. e., Propositions 5.3.3 and 5.3.4 we find a minimal diagnosis of F_1 by removing a minimal hitting set $\mathcal{S}_1$ of $SI^A_{min}(F_1)$. In general it is not quite clear whether we can now extend $\mathcal{S}_1$ to a *minimal* hitting set $\mathcal{S}$ of $SI^A_{min}(F)$. Due to $SI^A_{min}(F_1) \subseteq SI^A_{min}(F)$ we can surely extend $\mathcal{S}_1$ to a hitting set of $SI^A_{min}(F)$, but this way, minimality is not yet guaranteed. We can however prove it via Proposition 5.4.13:

Proposition 5.4.15. *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$. If $\mathcal{S}_1$ is a minimal hitting set of $SI^A_{min}(F_1, stb, cred)$ ($SI^R_{min}(F_1, stb, cred)$), then there is a minimal hitting set $\mathcal{S}$ of $SI^A_{min}(F, stb, cred)$ ($SI^R_{min}(F, stb, cred)$) with $\mathcal{S}_1 \subseteq \mathcal{S}$.*

Proof. By Propositions 5.3.3 and 5.3.4, $\mathcal{S}_1$ is a minimal hitting set of $SI^A_{min}(F_1, stb, cred)$ ($SI^R_{min}(F_1, stb, cred)$) iff $\mathcal{S}_1$ is a minimal diagnosis of F_1. Due to Proposition 5.4.13 there is a minimal diagnosis $\mathcal{S}$ of F with $\mathcal{S}_1 \subseteq \mathcal{S}$. Again due to Propositions 5.3.3 and 5.3.4, $\mathcal{S}$ is a minimal hitting set of $SI^A_{min}(F, stb, cred)$ ($SI^R_{min}(F, stb, cred)$). □

We want to mention that the situation differs when considering skeptical reasoning. If F_1 is such that $stb(F_1) \neq \emptyset$ but no argument is skeptically accepted, then it might still be the case that F possesses a skeptically accepted argument due to F_2. So consistency of F_1 is not a necessary condition anymore. It is also not yet sufficient, but with an additional premise:

Proposition 5.4.16. *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$. If F_1 is consistent wrt. skeptical reasoning and stable semantics, then so is F iff there is at least one extension $E_1 \in stb(F_1)$ such that $stb\left(F_2^{E_1, R_3}\right) \neq \emptyset$.*

Proof. Immediate from Theorem 5.4.12. □

So far, our investigation was restricted to stable semantics. The reason is quite simple: Theorem 5.4.12 is based on the (E_1, R_3)-reduct which requires further adjustments to obtain the desired result for other semantics. The intuitive reason is that there might be arguments $a \in A$ which are neither in a σ-extension E nor attacked by E if we consider $\sigma \neq stb$ (see [12] for more details). Formally, we consider the set of *undefined* arguments wrt. E (also known as *undecided* arguments) as follows.

Definition 5.4.17. If $F = (A, R)$ is an AF, σ any semantics and $E \in \sigma(F)$, then the set of *undefined* arguments wrt. E is

$$U_E = \{b \in A \mid b \notin E, \nexists a \in E : (a, b) \in R\}.$$

To obtain the splitting result for other semantics, we use the (E_1, R_3)-reduct as usual, but in addition we introduce dummy attacks to those arguments in F_2 which are attacked by an undefined argument from F_1 wrt. E_1. More precisely, the (S, L)-modification is defined as follows:

Definition 5.4.18 (Modification)**.** Let $F_2 = (A_2, R_2)$ be an AF and let A_1 be a set of arguments such that $A_1 \cap A_2 = \emptyset$. Let $S \subseteq A_1$ and $L \subseteq A_1 \times A_2$. The (S, L)-modification of F_2, denoted by F_2, is the AF

$$mod_{S,L}(F_2) = (A_2, R_2 \cup \{(b, b)\} \mid \exists a \in S : (a, b) \in L).$$

Observe that we are not interested in $mod_{U_{E_1},R_3}(F_2)$, but in $mod_{U_{E_1},R_3}\left(F_2^{E_1,R_3}\right)$, so we consider the (U_{E_1}, R_3)-modification of the (E_1, R_3)-reduct of F_2.

Theorem 5.4.19 ([12])**.** *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$. Let $\sigma \in \{stb, ad, pr, co, gr\}$.*

- *If E_1 is a σ-extension of F_1 and E_2 a σ-extension of the (U_{E_1}, R_3)-modification of $F_2^{E_1,R_3}$, then $E_1 \cup E_2$ is a σ-extension of F.*
- *Vice versa, if E is a σ-extension of F, then $E_1 = E \cap A_1$ is a σ-extension of F_1 and $E_2 = E \cap A_2$ a σ-extension of the (U_{E_1}, R_3)-modification of $F_2^{E_1,R_3}$.*

For stable semantics and credulous reasoning, the splitting theorem can be used to infer that consistency of F_1 is *necessary* for consistency of F. For the other semantics, it is not a necessary, but a *sufficient* condition.

Proposition 5.4.20. *Let $F = (A, R)$ be an AF and let (F_1, F_2, R_3) be a splitting of F, i. e., we have $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ where $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$. Let $\sigma \in \{ad, pr, co, gr\}$. If F_1 is consistent wrt. σ, then F is consistent wrt. σ as well. This holds for both credulous and skeptical reasoning.*

Proof. Immediate from Theorem 5.4.19. □

The treatment of an AF using splitting is convenient since the structural properties induce strong results. It is thus not surprising, yet encouraging to see that this principle is capable of improving the investigation of diagnoses and repairs of AFs. As it turns out, splitting can be used to reduce the search space for repairs (see Proposition 5.4.20) or compute minimal diagnoses stepwise (as in Proposition 5.4.13). Moreover, splitting is also meaningful when looking for strongly inconsistent arguments resp. attacks. We believe this is a promising research direction for further investigation, including concrete algorithms to compute repairs.

5.4.3 Infinite Frameworks

Until now, our investigation was restricted to finite AFs, i. e., $F = (A, R)$ where A is a finite set of arguments (and thus R a finite set of attacks). Within this section we want to drop this restriction and investigate which results still hold. As usual, when moving from the finite to the infinite case, we are concerned about *existence* and *uniqueness* of certain sets as this might not be clear anymore (cf. [20; 21] for a treatment of unrestricted AFs). In order to keep this section concise, we focus on universally defined semantics $\sigma \in \{co, pr, ss\}$ and in addition gr since gr-diagnoses play an important role similar to the finite case described in Section 5.2. We focus on argument-based diagnoses.

Let us start with grounded semantics. In the finite case, we observed that an F is consistent wrt. grounded semantics iff there is at least one unattacked argument. Let us formally state that this is also the case for infinite AFs.

Proposition 5.4.21. *If $F = (A, R)$ is an infinite AF, then F possesses a non-empty grounded extension iff there is at least one unattacked argument.*

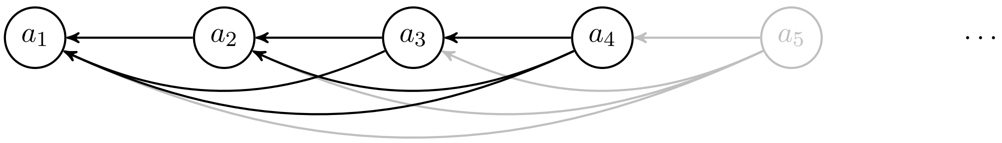
This also means we find gr-diagnoses as before, namely by removing arguments resp. attacks until at least one argument is unattacked. Hence:

Fact 5.4.22. If $F = (A, R)$ is an infinite AF, then F possesses a

- argument-based gr-diagnosis iff it is not self-controversial,
- attack-based gr-diagnosis.

Whether a *minimal* gr-diagnosis exists for a given AF $F = (A, R)$ is no longer trivial since we cannot just move from a diagnosis $\mathcal{S}$ to a minimal one $\mathcal{S}'$ as it is the case for finite frameworks. Indeed, when considering argument-based diagnoses, there is no minimal gr-diagnosis in general.

Example 5.4.23. Recall the AF $F = (A, R) = (\{a_i \mid i \in \mathbb{N}\}, \{(a_i, a_j) \mid i > j\})$ from Example 3.4.2 (Section 3.4 about infinite knowledge bases):



It is easy to check that this AF possesses gr-diagnoses: Let $j \in \mathbb{N}$. Then, there is an attack $(a_i, a_j) \in R$ for each $i \in \mathbb{N}$ with $i > j$. So in order to obtain a gr-diagnosis of F we may remove each argument a_i with $i > j$, i. e., we let $\mathcal{S}_j = \{a_i \mid i > j,\ i \in \mathbb{N}\}$. This is a diagnosis since no argument attacks a_j within $F_{\mathcal{S}_j}$, so the grounded extension is non-empty. It is not minimal though since j was arbitrary. More precisely, for any $j' > j$, the set $\mathcal{S}_{j'}$ is a gr-diagnosis of F as well satisfying $\mathcal{S}_{j'} \subseteq \mathcal{S}_j$. Since we can always move to a smaller diagnosis, we see that there is no minimal one.

Now let $\sigma \in \{pr, co\}$. In [49] it has been noted that those semantics are also universally defined when considering infinite AFs. Thus given a gr-repair, we also have a σ-repair as in the finite case.

Proposition 5.4.24. *Let $F = (A, R)$ be an infinite AF and $\sigma \in \{pr, co\}$. If $\mathcal{S}$ is a gr-diagnosis of F, then $\mathcal{S}$ is a σ-diagnosis of F as well. This holds for both reasoning modes.*

Proof. Consider complete semantics and skeptical reasoning. Since the gr-extension is the least co-extension and nonempty for $F_{\mathcal{S}}$ by assumption, we have that at least one argument, say $a \in A$, is skeptically accepted wrt. complete semantics. This also implies the claim for credulous reasoning. Now consider $\sigma = pr$. Recall that if E is a preferred extension, then it is a complete extension as well. Hence,

$$a \in \bigcap co(F_{\mathcal{S}}) \subseteq \bigcap pr(F_{\mathcal{S}})$$

implying a is also skeptically accepted for preferred semantics. This finishes our proof. □

This does not work for semi-stable extensions since an infinite AF does not necessarily posses one [20; 116]. Augmenting an arbitrary AF with no semi-stable extension with an argument participating in no attack whatsoever yields a counterexample for Proposition 5.4.24 for $\sigma = ss$. Moreover, we note that Example 5.4.23 shows that minimal argument-based σ-diagnoses do not necessarily exist for $\sigma \in \{pr, co\}$ since this example works analogously for $\sigma \in \{pr, co\}$. We thus see that argument-based diagnoses are not as well-behaved as in the finite case.

We conclude this section with some good regarding so-called *finitary* AFs. In a finitary AF, each argument is only allowed to have finitely many attackers. This solves nearly all issues we had during this section with *minimal* diagnoses at once: Now, any gr-diagnosis is necessarily finite and given a finite diagnosis, we can easily move to a minimal one. Hence, we obtain existence of minimal complete and preferred diagnoses. Moreover, any finitary AF possesses as semi-stable extension [20; 116], so the same can be guaranteed here.

Definition 5.4.25. The AF $F = (A, R)$ is called *finitary* if $\{a \in A \mid (a, b) \in R\}$ is finite for each $b \in A$.

Theorem 5.4.26. *Let F be finitary. Any minimal gr-diagnosis of F is finite. If $\mathcal{S}$ is a gr-diagnosis of F, then there is a minimal gr-diagnosis $\mathcal{S}'$ of F with $\mathcal{S}' \subseteq \mathcal{S}$. If $\mathcal{S}'$ is a minimal gr-diagnosis of F, then there is a minimal σ-diagnosis $\mathcal{S}''$ of F with $\mathcal{S}'' \subseteq \mathcal{S}'$ for any $\sigma \in \{co, pr, ss\}$.*

Proof. The claims about gr-diagnoses are clear. Given a finite gr-diagnosis, apply Proposition 5.4.24 to obtain a finite σ-diagnosis for $\sigma \in \{co, pr\}$. Due to finiteness, one can turn this diagnosis into a minimal one. Moreover, due to [116], any finitare AF F possesses a semi-stable extension. Now if $F_{\mathcal{S}'}$ is a minimal gr-repair, then there is at least one unattacked argument. It is easy to see that this occurs in each semi-stable extension, so $\mathcal{S}'$ is a ss-diagnosis as well. Again we can move to a minimal one due to finiteness. □

Observe that –speaking in terms of Section 3.4– we just proved that finitary AFs possess the co-compactness property (see Definition 3.4.16) for $\sigma \in \{co, pr, ss\}$ wrt. both reasoning modes.

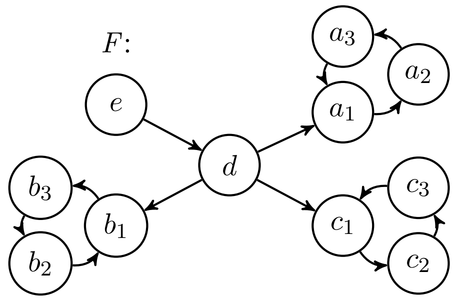
When investigating infinite instead of finite AFs, one needs to accept the possibility that certain results are not conveyed. In our case, the investigation showed that argument-based diagnoses are somewhat clumsy and minimality can almost never be guaranteed. It is worth mentioning that a reasonably simple example suffices to demonstrate this for all considered semantics and both reasoning modes. As formalized in Theorem 5.4.26, the issues we faced in this section stem from the infinite number of *attacks* (to single arguments) rather than the infinite number of *arguments*.

5.5 How to Repair? - A Short Case Study

As mentioned before, due to Lemmata 5.2.15, 5.2.18 and 5.2.22 we may reduce the search space for diagnoses as long as we are equipped with an already computed grounded one. If one is interested in *all* diagnoses, the notion of strong inconsistency in order to use the hitting set duality might be useful. The aim of this section is to briefly demonstrate how to repair a given AF. We discuss both credulous and skeptical reasoning.

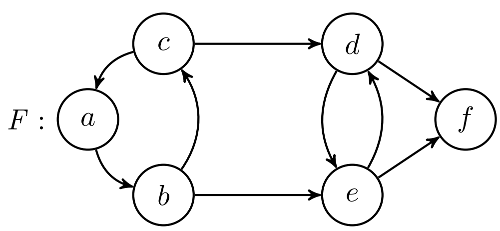
First let us consider an example with stable semantics. Let us start with credulous reasoning. It is well-known that in case of finite AFs the non-existence of acceptable positions implies the existence of odd-cycles. This means, by contraposition, one possible strategy for repairing AFs in case of stable semantics is to break odd-cycles. This approach corresponds to the minimal stb-repairs $F_{\{a\}}$, $F_{\{b\}}$ and $F_{\{c\}}$ from Example 5.3.5. Since possessing odd-cycles is not sufficient for the collapse of stable semantics further considerations are required. Indeed, in case of our running example, we have seen that eliminating the arguments e and f results in a minimal stb-repair, namely $F_{\{e,f\}}$, too. Regarding the principle of minimal change one may argue that breaking the odd-cycle in F has to be preferred over the latter strategy since less arguments are involved. The following slightly modified version of this example shows that this observation is not true in general. A further intensive study of this issue will be part of future work.

Example 5.5.1. Consider the following AF F. One may easily confirm that there are 9 minimal $cred$-stb-diagnoses, namely $\{a_i, b_j, c_k\}$ with $i, j, k \in \{1, 2, 3\}$. They comply with the idea to break all odd loops of the given AF. However, $\{e\}$ is a minimal $cred$-stb-diagnosis as well, and arguably the most immediate one.

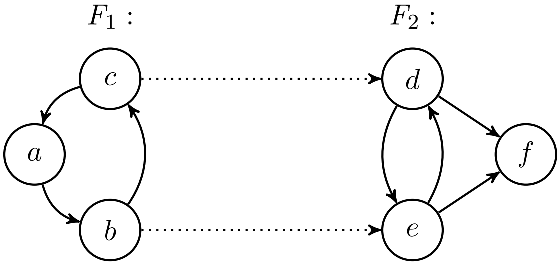


As the reader may have already observed, the same applies to attack-based diagnoses. For example, $\{(a_1, a_2), (b_1, b_2), (c_1, c_2)\}$ is an attack-based diagnosis breaking all odd loops. However, consideration of $\{(e, d)\}$ suffices.

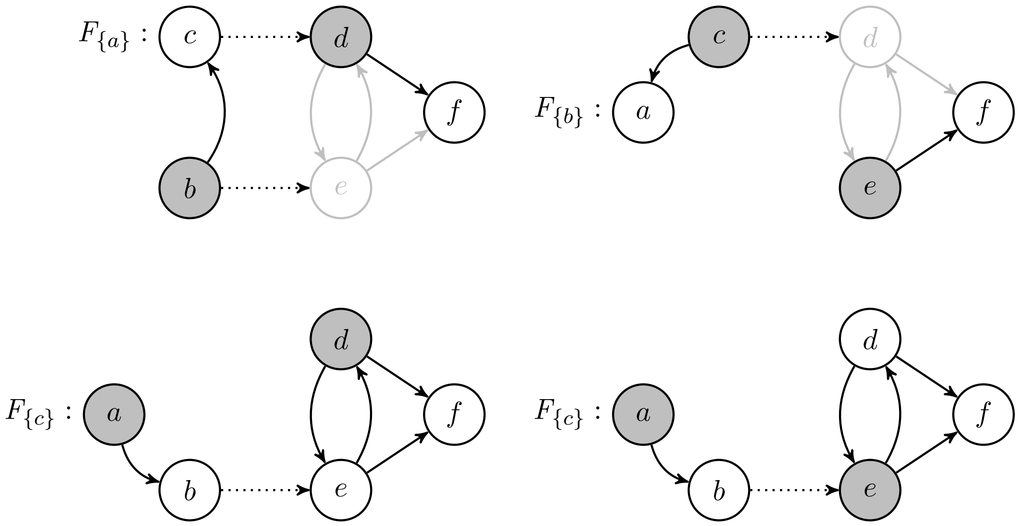
Example 5.5.2. Let now F be the following AF:



We set $F_1 = (A_1, R_1)$ with $A_1 = \{a, b, c\}$ and $R_1 = \{(a, b), (b, c), (c, a)\}$ as well as $F_2 = (A_2, R_2)$ with $A_2 = \{d, e, f\}$ and $R_2 = \{(d, e), (e, d), (d, f), (e, f)\}$, and finally $R_3 = \{(b, e), (c, d)\}$. Then (F_1, F_2, R_3) is a splitting of F:



Since F_1 does not possess a stable extension, we need to remove a, b or c. In any case, there is a sceptically accepted argument. The only stb-repair with two extensions is $F_{\{c\}}$:

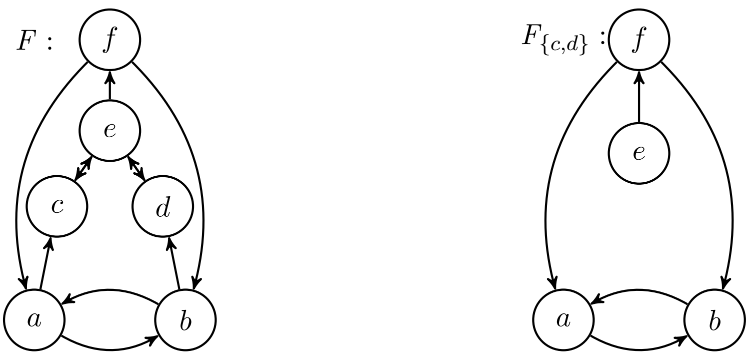


Due to the splitting Theorem 5.4.12, we can be sure that we found exactly the three minimal stb-repairs.

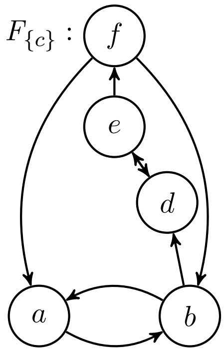
The subsequent example considers a semantical defect wrt. preferred semantics which is tackled via grounded repairs.

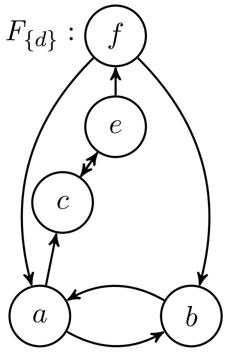
Example 5.5.3. The following AF F exemplifies a situation where preferred semantics do not possess any skeptically accepted argument. More precisely, $\bigcap pr(F) = \emptyset$ due to

$$pr(F) = \{\{a, e\}, \{b, e\}, \{c, d, f\}\}.$$



Our goal is to find a minimal $skep$-pr-diagnosis $\mathcal{S}$, i.e., a set $\mathcal{S}$ such that $\bigcap pr\big(F_{\mathcal{S}}\big) \neq \emptyset$ and $pr\big(F_{\mathcal{S}}\big) \neq \emptyset$. Lemma 5.2.18 suggests that looking for gr-repairs is a reasonable starting point. In order to guarantee at least one unattacked argument one finds $\{c, d\}$ as minimal gr-diagnosis. Let $F_{\{c,d\}}$ denote the associated minimal repair. We have $gr\left(F_{\{c,d\}}\right) = \{\{e\}\}$. Hence, $\bigcap pr\big(F_{\{c,d\}}\big) \neq \emptyset$ is implied. This means, $\{c, d\}$ is a $skep$-pr-diagnosis. Moreover, $\{c, d\}$ is even minimal proven by the following two AFs $F_{\{c\}}$ and $F_{\{d\}}$.





Indeed, we have $\bigcap pr\big(F_{\{c\}}\big) = \bigcap pr\big(F_{\{d\}}\big) = \emptyset$ since $\{a, e\}, \{d, f\} \in pr\big(F_{\{c\}}\big)$ and $\{a, e\}, \{c, f\} \in pr\big(F_{\{d\}}\big)$.

5.6 Conclusions and Related Work

In this chapter, we investigated approaches aiming at repairing argumentation frameworks which are inconsistent in the sense that they do not possess any accepted argument. We considered a reasonable range of semantics, the standard reasoning modes, namely credulous and skeptical reasoning, and two different tools to repair, namely removal of a certain (minimal) sets of arguments or attacks. We identified repairs wrt. grounded semantics as the arguably most important case: They can be utilized as a starting point in order to calculate repairs wrt. other semantics, coincide for both reasoning modes and in Section 6.4 below we will even see that they are tractable from an algorithmic point of view. We illustrated how to derive stronger results for specific situations like restricting the AFs to certain subclasses or if the AF allows for a splitting. We also investigated infinite AFs.

The topic of *diagnoses* and *repairs* as introduced in [96] is less developed in the area of abstract argumentation. The closest one to our work is probably [88]. The authors define an operator and provide an algorithm, st. the resulting framework does not collapse. The mentioned work considers a semantical defect as the absence of any extension. Consequently, only stable semantics can be considered in contrast to our setup which additionally includes a treatment of semantics which may provide the empty set as unique extension. All semantics known from the literature do so. Moreover, restoring consistency is achieved via dropping a minimal set of arguments or attacks. In the latter case, all arguments survive the revision process.

The very first and basic works which are dealing with dynamics in abstract argumentation are [13; 16; 17] as well as [31; 40]. The first two are tackling the so-called *enforcing problem* wrt. possibility as well as minimal change. More precisely, they are dealing with the question whether it is possible (and if yes, as little effort as possible) to add new information in such a way that a desired set of arguments becomes an extension or at least a subset of one. In [74] the authors studied this problem under the name *σ-repair* and provided parametrized complexity results. Although adding information as well as desired sets are not the focus of our study there is at least one interesting similarity to our work, namely: given an AF where nothing is credulously accepted, then enforcing a certain non-empty set can be seen as a special kind of repairing. The other two works are case studies of what happens with the set of extensions if one deletes or adds one argument. The so-called *destructive change* is somehow the inverse of our notion of credulous repair since the initial framework possesses at least one credulously accepted argument whereas the result does not. Quite recently, in [19] the so-called *extension removal problem* was studied. That is, is it possible to modify a given AF in such a way that certain undesired extensions are no longer generated?

Chapter 6

Computational Complexity

This chapter is devoted to the computational complexity of decision and function problems which naturally arise in sight of the results we obtained up until this point. The most apparent ones are concerning *strong inconsistency*. We demonstrated how strong inconsistency generalizes the properties of inconsistency to non-monotonic logics. However, the price we need to pay is consideration of all supersets instead of just $\mathcal{H} \subseteq \mathcal{K}$ itself. We will hence not only focus on *strong* inconsistency, but in particular how it compares to mere inconsistency in monotonic logics. Consideration of minimal inconsistency in monotonic logics is not novel: For example the paper [91] discusses minimal inconsistency of a given propositional formula and the discussion of the complexity of inconsistency measures in [108] also involves counting minimal inconsistent subsets of a knowledge base. With the results from the literature in mind, we are interested in the following problems:

- How demanding is finding strongly inconsistent subsets from a computational point of view?
- How demanding is the corresponding *counting* problem, i. e., computing $|SI_{min}(\mathcal{K})|$?
- How do our problems compare to corresponding ones regarding ordinary inconsistency in monotonic logics?

It is clear that the answer to those questions depends on the logic under consideration, more precisely on the satisfiability check. Most of our results are thus relative to the following decision problem:

SAT_L	**Input**:	$\mathcal{K} \subseteq \mathcal{WF}$
	Output:	TRUE iff $\mathcal{K}$ is consistent

We may thus give some general upper bounds for the problems we consider, depending on the complexity of SAT_L. However, lower bounds usually require constructions which utilize specific properties a particular logic under consideration possesses. To demonstrate how to infer lower bounds for our setting, we discuss most of the problems related to strong inconsistency for ASP.

In contrast to Chapters 3 and 4, Chapter 5 about inconsistency in abstract argumentation was mainly concerned about *consistent* subframeworks and repairs, as well as their relations. We will proceed similarly with our discussion about the computational complexity and focus on deciding *maximal consistency*. Here we keep in mind that repairs for *grounded* semantics were helpful when reducing the search space for repairs wrt. other semantics. We thus ask:

- How demanding is finding maximal consistent subframeworks from a computational point of view?
- How do skeptical and credulous reasoning compare? How do different semantics compare?
- Can gr-diagnoses be computed within a reasonable amount of time?

We proceed similarly here and give general upper bounds before discussing lower bounds for the various semantics. In comparison to the prior discussion on strong inconsistency this shall demonstrate the similarities between the notions and the results we obtain.

In a nutshell, the outline of this chapter is as follows: We cover monotonic and non-monotonic logics, decision and counting problems, and consider minimal strong inconsistency for ASP and maximal consistency for AFs.

Figure 6.1: Aspects of computational complexity

Some of the proofs in this chapter contain quite lengthy constructions and are not necessary to follow the investigation we are going to perform. They can be found in Appendix A.

6.1 Background

We assume the reader to be familiar with the classes P, NP and coNP. Furthermore, we consider the polynomial hierarchy as usual (see for instance [90]): we let $\Sigma_0^p = \Pi_0^p = \mathsf{P}$ and for $m \geq 1$, Σ_m^p is the class of all languages L such that there is a polynomial time Turing machine M (with output $M(x, X_1, ..., X_m)$) and a polynomial p such that $x \in L$ if and only if

$$\exists X_1 \dots Q_m X_m : \; M(x, X_1, \dots, X_m) = 1 \tag{6.1}$$

with $|X_1|, ..., |X_m| \leq p(|x|)$ and alternating $Q_i \in \{\exists, \forall\}$ for $2 \leq i \leq m$. The class Π_m^p is defined analogously, but the expression in (6.1) starts with a universal quantifier rather then an existential one. Note that $\Sigma_1^p = \mathsf{NP}$ and $\Pi_1^p = \mathsf{coNP}$.

An equivalent formalization makes use of oracle machines. Let $\mathcal{C}^{\mathcal{D}}$ be the class of decision problems solvable in $\mathcal{C}$ having access to an oracle for some problem that is complete in $\mathcal{D}$. Then we have $\Sigma_0^p := \Pi_0^p := \mathsf{P}$ and for each $m \geq 0$,

$$\Sigma_{m+1}^p = \mathsf{NP}^{\Sigma_m^p}, \qquad \Pi_{m+1}^p = \mathsf{coNP}^{\Sigma_m^p}.$$

The generic framework to capture complexity classes within the polynomial hierarchy is *quantified boolean formulas* (QBFs). A QBF Φ is a formula

$$\Phi = Q_1 X_1 \dots Q_m X_m \, \phi \tag{6.2}$$

with alternating quantifiers $Q_1, \dots, Q_m \in \{\forall, \exists\}$, pair-wise disjoint sets $X_1, \dots, X_m$ of variables, and a propositional formula ϕ over the variables $X_1 \cup \dots \cup X_m$. Here, we slightly abuse notation and consider quantifiers over *sets* of variables. More precisely, $Q_i X_i$ is to be understood as $Q_i x_{i,1}, \dots, Q_i x_{i,n_i}$ where $X_i = \{x_{i,1}, \dots, x_{i,n_i}\}$.

A QBF Φ is *true* if ϕ evaluates to true wrt. the quantifiers. A QBF Φ of the form (6.2) is in *prenex normal form* (PNF) if the quantifiers $Q_1, \dots, Q_m$ alternate between $\forall$ and $\exists$.

Example 6.1.1. The QBF

$$\forall x_1 \exists x_2 (x_1 \Leftrightarrow \neg x_2)$$

is true and in prenex normal form.

We will also consider *open QBFs*. They are defined similar as QBFs, but also contain free variables. Thus, an open QBF is of the form

$$\Phi = \Phi(X) = Q_1 X_1 \dots Q_m X_m \, \phi(X, X_1, \dots, X_m).$$

If $\Phi = \Phi(X)$ is an open QBF, then $\textsc{Mod}(\Phi)$ is the set of all models of Φ, i. e., the set of all assignments to the X-variables rendering ϕ true with respect to the quantifiers.

Example 6.1.2. For the open QBF

$$\Phi = \Phi(x) = \forall x_1 \exists x_2 \, x \wedge (x_1 \Leftrightarrow \neg x_2)$$

we have $\textsc{Mod}(\Phi) = \{x \mapsto 1\}$.

If a QBF is not an open QBF, it is called a *sentence*.

The problem of deciding whether a QBF Φ with m alternating quantifiers starting with $\exists$ (resp. starting with $\forall$) is true is the canonical Σ_m^p-complete (resp. Π_m^p-complete) problem [90]. More precisely, this is the case wrt. *polynomial-time many-one reductions* (or *polynomial reductions* for short). The hardness results we are going to give for decision problems are wrt. polynomial reductions and we assume the reader to be familiar with this notion.

We also make use of the differences classes D_m^p [90]. They were introduced to capture threshold problems like "Is it true that a given graph G has a Hamiltonian cycle, but if one removes an arbitrary edge, the resulting graph does not?" Observe that this corresponds to checking that a certain instance belongs to a language decidable in NP, while another instance does not. Formally,

$$\mathsf{D}_1^p = \{L_1 \setminus L_2 \mid L_1, L_2 \in \mathsf{NP}\}$$

or, equivalently,

$$\mathsf{D}_1^p = \{L_1 \cap L_2 \mid L_1 \in \mathsf{NP},\ L_2 \in \mathsf{coNP}\}.$$

For example, the generic D_1^p-complete problem is SAT-UNSAT, where we are given two propositional formulas ϕ_1 and ϕ_2, and have to decide whether ϕ_1 is satisfiable while ϕ_2 is not. The natural generalization of D_1^p to higher levels of the polynomial hierarchy is

$$\mathsf{D}_m^p = \{L_1 \cap L_2 \mid L_1 \in \Sigma_m^p,\ L_2 \in \Pi_m^p\}.$$

Moreover, we consider PSPACE, i. e., the class of all languages L that can be computed using polynomial space. In contrast to Σ_m^p and Π_m^p, where we can decide the truth value of a QBF with m alternating quantifiers, the generic PSPACE-complete problem is deciding the truth value of an *arbitrary* QBF.

Our investigation of the inconsistency measures will lead to the consideration of counting complexity classes (cf. [114]). They are defined using witness functions w that assign words from an input alphabet Σ to finite subsets of an alphabet Γ. Given a string x from the alphabet Σ, the task is to return $|w(x)|$, i. e., the number of witnesses. Given a class $\mathcal{C}$ of decision problems, by $\#\cdot\mathcal{C}$ we denote the class of counting problems such that

- for every input string x, each $y \in w(x)$ is polynomially bounded,
- the decision problem "Is $y \in w(x)$?" is in $\mathcal{C}$.

For example, the generic $\#\cdot\Pi_2^p$-complete problem is counting $|\textsc{Mod}(\Phi)|$ for an open QBF $\Phi = \forall Y \phi(X, Y)$ (see [50]). Here, $\textsc{Mod}(\Phi)$ is the witness function assigning to a given formula the corresponding models. As required, each truth assignment is polynomial bounded and given an assignment to the X-variables, the decision problem whether $\forall Y \phi(X, Y)$ holds is in coNP.

Hardness results for function problems are going to be given under subtractive reductions [50]. For that, let $\#V$ and $\#W$ be counting problems with witness sets $v(x)$ and $w(y)$. The problem $\#V$ reduces to $\#W$ under *strong subtractive reductions* if there are polynomial-time computable functions f and g such that $w(f(x)) \subseteq w(g(x))$ and $|v(x)| = |w(g(x))| - |w(f(x))|$. *Subtractive reductions* are the transitive closure of strong subtractive reductions. The reason to consider novel notions of reductions for function problems are also discussed in [50], let us briefly mention an issue with polynomial reductions: From results in [109] one can, for example, infer that there is a problem that is $\#\cdot P$-complete, but also $\#\cdot\Pi_k^p$-complete under polynomial-time many-one reductions for any integer k. This strongly suggests that the classes $\#\cdot\Pi_k^p$ are *not* closed under those reductions.

For some decision problems we require the counting polynomial hierarchy [115]. For this, consider the counting quantifier C: Given a predicate $R(x, y)$ with free variables x and y, let

$$\mathsf{C}_y^k R(x, y) :\Leftrightarrow |\{y \mid R(x, y) \text{ true }\}| \geq k.$$

The counting quantifier is true for the predicate R and the bound k iff there are at least k values for y st. R is true. The polynomially bounded version of this quantifier is defined as follows. For any class $\mathcal{C}$ of problems, A is in $\mathsf{C}\mathcal{C}$ if there is a $B \in \mathcal{C}$, a function f computable in P and a polynomial p such that

$$x \in A :\Leftrightarrow \mathsf{C}_{|y| \leq p(|x|)}^{f(x)}(x, y) \in B,$$

i. e., there are at least $f(x)$ many (by a polynomial in x bounded) values y such that a predicate holds for (x, y). The latter check shall be in B.

Example 6.1.3. A generic complete problem for the class CP is the following: Given a propositional formula ϕ, is it true that $|\textsc{Mod}(\phi)| \geq k$? To give a generic example for the class CcoNP, we make use of open QBFs: Given an open QBF $\Phi = \forall Y \phi(X, Y)$, is it true that $|\textsc{Mod}(\Phi)| \geq k$?

6.2 The Complexity of Inconsistency

Let us now focus on decision problems that are related to strong inconsistency. The notion of strong inconsistency includes consideration of all supersets of a given set, which is apparently more involved than ordinary inconsistency in monotonic logics. We are thus interested in the computational complexity of deciding (minimal) strong inconsistency and, in particular, the difference between monotonic and non-monotonic logics. In order to formalize the comparison between ordinary and strong inconsistency, we start by investigating QBFs. They are a suitable tool due to the following reasons:

- QBFs are *monotonic* in the following sense: If ϕ evaluates to false wrt. the given quantifiers, then the same is true if we augment ϕ with additional clauses.
- QBFs capture the whole polynomial hierarchy in the following sense: For any class Σ_k^p resp. Π_k^p there is a class of QBFs such that deciding their truth value is complete in this class.

After discussing QBFs, we will continue with upper bounds for our decision problems which involve strong inconsistency. Even though these bounds appear rather generic, we will see that they cannot be improved in general. We will also give corresponding lower bounds for ASP and conclude this section with a generic algorithm to compute strongly inconsistent subsets of a given knowledge base.

Let us now formally introduce the decision problems we are going to consider. They are phrased for an arbitrary logic L, but keep in mind that they should be assessed relative to the satisfiability check $\textsc{Sat}_L$.

S-INC$_L$	**Input**:	$(\mathcal{K}, \mathcal{H})$ with $\mathcal{K} \subseteq \mathcal{WF}$, $\mathcal{H} \subseteq \mathcal{K}$
	Output:	TRUE iff $\mathcal{H} \in SI(\mathcal{K})$
MIN-S-INC$_L$	**Input**:	$(\mathcal{K}, \mathcal{H})$ with $\mathcal{K} \subseteq \mathcal{WF}$, $\mathcal{H} \subseteq \mathcal{K}$
	Output:	TRUE iff $\mathcal{H} \in SI_{min}(\mathcal{K})$

So, S-INC$_L$ simply asks whether a given subset $\mathcal{H}$ of a knowledge base $\mathcal{K}$ is *strongly inconsistent*. The problem MIN-S-INC$_L$ additionally requires minimality of $\mathcal{H}$.

6.2.1 Minimal Unsatisfiability for QBFs

In the literature, there is a counterpart to our problem MIN-S-INC$_L$ for propositional logic. One of the main results in [91] is that the following problem Minimal Unsatisfiability (MU) is D_1^p-complete:

MU	**Input**:	A propositional formula $\phi = \{C_1, \ldots, C_r\}$ in CNF
	Output:	TRUE iff ϕ is inconsistent, but $\phi \setminus C_k$ is consistent for any k

In other words, a formula ϕ in CNF consisting of clauses $C_1, \ldots, C_r$ is a positive instance if it is not satisfiable, but removing an *arbitrary* clause C_k of ϕ ensures existence of a satisfying assignment. Since we do not want to restrict ourselves to logics whose satisfiability check is in NP, we need to lift this result to a wider range of complexity classes.

One such generalization can be found in [39], where it has been shown that deciding whether a QBF is false, but any subformula of it is true is PSPACE-complete. Unfortunately, the class PSPACE is, roughly speaking, too big for our comparison between ordinary and strong inconsistency. To see this we need to recall the the two inconsistency notions: Checking whether $\mathcal{H}$ is inconsistent requires a satisfiability check for $\mathcal{H}$. Checking whether $\mathcal{H}$ is strongly inconsistent requires taking the supersets of $\mathcal{H}$ into account as well. Now assume the satisfiability check is in PSPACE. Then, within PSPACE we can not only perform the satisfiability checks, but also enumerate each subset of a given knowledge base. Hence, there is no difference between ordinary and strong inconsistency.

Proposition 6.2.1. *If the decision problem* SAT_L *is in PSPACE, then both* S-INC_L *as well as* MIN-S-INC_L *are in PSPACE.*

Proof. Enumerate all subsets of $\mathcal{K}$ and perform corresponding consistency checks. It is then clear whether or not $(\mathcal{K}, \mathcal{H})$ is a "yes" instance of S-INC_L resp. MIN-S-INC_L. □

We thus focus on logics whose decision problem SAT_L lies within the polynomial hierarchy. In terms of QBFs this means we need to fix the number of quantifiers. This motivates the following problem QBF-MU$(Q_1, ..., Q_m)$:

QBF-MU$(Q_1, ..., Q_m)$ **Input**: $\Phi = Q_1X_1 \dots Q_mX_m\, \phi$ in PNF, $\phi = C_1 \wedge \dots \wedge C_r$
Output: TRUE iff Φ is false, but true for $\phi \setminus C_k$ for any k

Since ϕ is a conjunction, Φ is true if and only if all conjuncts $C_1, \dots, C_r$ evaluate to true (wrt. the quantifiers). Note that Φ evaluates to true if ϕ is the empty conjunction.

Recall that we are looking for a generalization of the result in [91] since we are interested in minimal inconsistency for monotonic logics beyond cases where SAT_L is in NP. We are now ready to give such a generalization for the problem QBF-MU$(Q_1, ..., Q_m)$.

Theorem 6.2.2. *If* $m \geq 2$*, then* QBF-MU$(Q_1, ..., Q_m)$ *is* D^p_m*-complete.*

Combining Theorem 6.2.2 and the result in [91] (i. e., $m = 1$ and $Q_1 = \exists$), one can observe that the case where Φ is of the form $\Phi = \forall X \phi$ is missing. Indeed, it turns out to be easier, at least under standard assumptions.

Proposition 6.2.3. QBF-MU$(\forall)$ *is NP-complete.*

The following scheme depicts the complexity of QBFs. The cases "inc?" are clear and the cases "MI?" (minimal inconsistent) are our previous results. Note that we require $m \geq 2$.

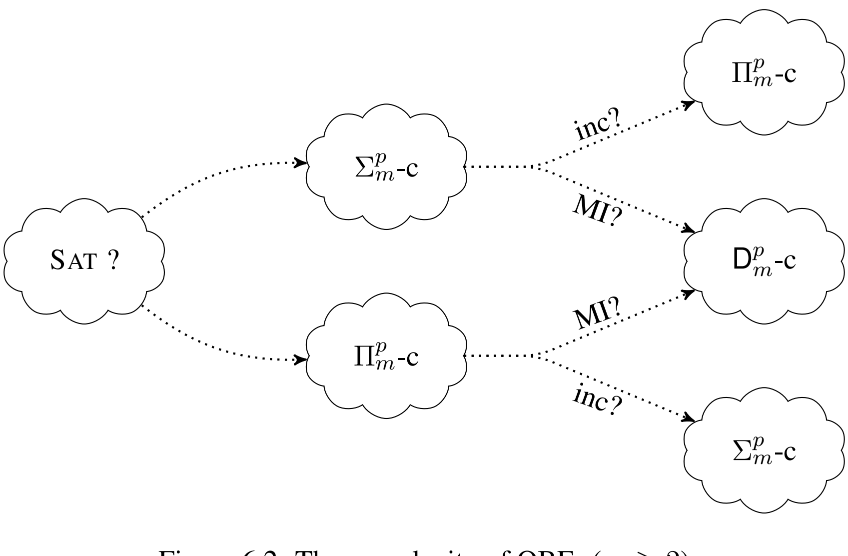


Figure 6.2: The complexity of QBFs ($m \geq 2$)

We want to mention that it is straightforward to cast deciding the truth value of a QBF into our general definition of a logic. Theorem 6.2.2 can then be interpreted as a result for QBFs regarding the problem MIN-S-INC$_L$. More precisely, this means that MIN-S-INC$_L$ is D^p_m-hard in general, if L is monotonic and SAT$_L$ is in Σ^p_m or Π^p_m for $m \geq 2$. Clearly, if the satisfiability check of our QBF is in Σ^p_P (Π^p_P), then verifying unsatisfiability is in Π^p_P (Σ^p_P). We will now see how these results generalize to *arbitrary* logics.

6.2.2 (Minimal) Strong Inconsistency in General

We now turn to the general discussion on the computational complexity of problems related to strong inconsistency. Assume $L = (\mathcal{WF}, \mathcal{BS}, \mathcal{INC}, \mathcal{ACC})$ is an arbitrary but fixed logic. If L is monotonic and SAT$_L \in \mathcal{C}$ for some class $\mathcal{C}$, then S-INC$_L$ is in co-$\mathcal{C}$. However, the notion of strong inconsistency is not only a property of $\mathcal{H}$ itself, but also of all sets $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$. Hence, deciding whether a given set $\mathcal{H}$ is strongly inconsistent may be more demanding in some cases, but not always as the following result shows.

Theorem 6.2.4. *Let* $m \geq 1$. *If the decision problem* SAT$_L$ *is in*

(a) Σ^p_m, *then* S-INC$_L$ *is in* Π^p_m,

(b) Π^p_m, *then* S-INC$_L$ *is in* Π^p_{m+1},

(c) Π^p_m *and L is monotonic, then* S-INC$_L$ *is in* Σ^p_m.

Proof. See proof of Theorem 6.2.5. □

Theorem 6.2.2 already showed how difficult MIN-S-INC$_L$ is compared to the decision problem SAT$_L$ in the generic framework of QBFs. As stated in Theorem 6.2.4, checking strong inconsistency is in general more difficult in non-monotonic frameworks and we obtain a similar result in the case of MIN-S-INC$_L$. However, the increase of the computational complexity stems from checking the “strong“ part in “minimal strong inconsistency” rather than the “minimal” part. For that reason and as the following result shows, moving from the problem S-INC$_L$ to the problem MIN-S-INC$_L$ -i. e., additionally asking for minimality- does not involve going up an additional level in the polynomial hierarchy but only moving to the corresponding D^p_m class.

Theorem 6.2.5. *Let* $m \geq 1$. *If the decision problem* SAT$_L$ *is in*

(a) Σ^p_m, *then* MIN-S-INC$_L$ *is in* D^p_m,

(b) Π^p_m, *then* MIN-S-INC$_L$ *is in* D^p_{m+1},

(c) Π^p_m *and L is monotonic, then* MIN-S-INC$_L$ *is in* D^p_m.

Proof. (a): We need to show that it is sufficient to solve one problem in Σ^p_m and one in Π^p_m. To check whether $\mathcal{H}$ is strongly inconsistent, we need to check that $\mathcal{H}'$ is inconsistent for each $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$. Checking that $\mathcal{H}'$ is a “no” instance of SAT$_L$ is in Π^p_m. Since there are only exponentially many $\mathcal{H}' \subseteq \mathcal{K}$, we can also decide in Π^p_m whether

$$\forall \mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K} : \ \mathcal{H}' \text{ is inconsistent}$$

is true. That is, a TM non-deterministically guesses a set $\mathcal{H}'$ as above and performs a satisfiability check. The minimality of $\mathcal{H}$ can be written as

$$\forall \mathcal{D} \subsetneq \mathcal{H}\, \exists \mathcal{D}' :\ \mathcal{D} \subseteq \mathcal{D}',\ \mathcal{D}' \text{ is consistent} \tag{6.3}$$

stating that each proper subset $\mathcal{D}$ of $\mathcal{H}$ has a consistent superset $\mathcal{D}'$, which ensures that $\mathcal{D}$ is not strongly $\mathcal{K}$-inconsistent. However, it is sufficient to check $|\mathcal{H}|$ subsets of $\mathcal{H}$: Let $\mathcal{H} = \{\alpha_1, \dots, \alpha_k\}$ and let $\mathcal{H}_i = \mathcal{H} \setminus \{\alpha_i\}$, i. e., we consider the $|\mathcal{H}|$ possible subsets of size $|\mathcal{H}| - 1$. Now consider an arbitrary set $\mathcal{D} \subsetneq \mathcal{H}$. Clearly, there is an i such that $\mathcal{D} \subseteq \mathcal{H}_i$. Hence, if $\mathcal{H}_i$ has a consistent superset $\mathcal{H}_i'$, then so has $\mathcal{D}$. Thus, all proper subset of $\mathcal{H}$ do have a consistent superset if and only if this is the case for $\mathcal{H}_1, ..., \mathcal{H}_k$. Hence, we only need to check these subsets of $\mathcal{H}$ and therefore, deciding whether

$$\exists \mathcal{H}_i \subseteq \mathcal{H}_i' :\ \mathcal{H}_i' \text{ is consistent}$$

is true (with $\mathcal{H}_1, ..., \mathcal{H}_k$ as described) is sufficient. Since there are only linearly many $\mathcal{H}_i$, this is in Σ^p_m if $\textsc{Sat}_L$ is in Σ^p_m. That is, for $i = 1, \dots, k$, a TM non-deterministically guesses a set $\mathcal{H}_i'$ as above and performs a satisfiability check. Now, the claim follows due to the definition of D^p_m.

(b): This follows from *(a)* since $\Pi^p_m \subseteq \Sigma^p_{m+1}$.

(c): If L is monotonic and $\textsc{Sat}_L$ in Π^p_m, then $\textsc{Min-S-Inc}_L$ corresponds to

- verifying that $\mathcal{H}$ is inconsistent (which is in Σ^p_m) and
- verifying that $\mathcal{H}_1, ..., \mathcal{H}_k$ as in (a) are consistent (which is in Π^p_m).

Thus, $\textsc{Min-S-Inc}_L$ is in D^p_m. This finishes our proof. □

The following scheme summarizes the membership results for $\textsc{Min-S-Inc}$ we obtained so far. We see that most cases yield membership in D^p_m:

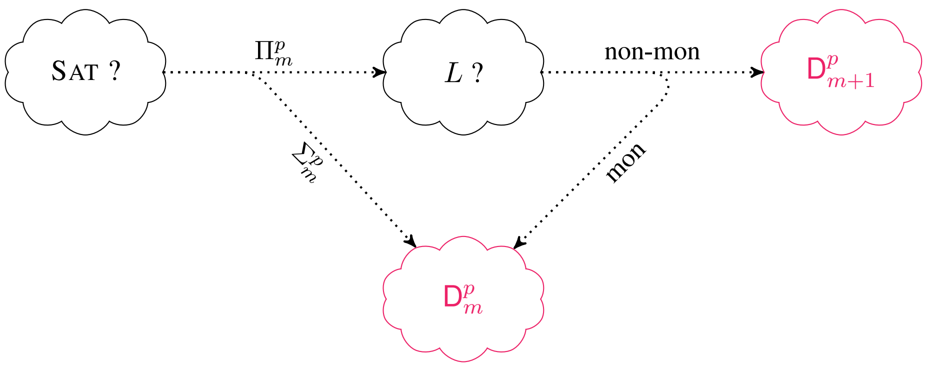


Figure 6.3: Membership results for $\textsc{Min-S-Inc}$ (red) depending on different cases ($m \geq 1$)

In the above Theorems 6.2.4 and 6.2.5, only membership statements are given. However, bear in mind that these results are valid for every logic that can be phrased via Definition 2.5.1. One cannot expect to obtain similar general hardness results as reductions may work very differently for different logics. In the subsequent section, we will give some concrete hardness results for ASP.

Before doing so, let us take a closer look at the two above theorems. Since they are only about upper bounds, it is natural to ask whether they can be improved in general. Here, the cases (a) and (c) are quite clear, since our investigation regarding QBFs already witnesses that the upper bounds we found are also lower bounds in general. The most interesting case is thus probably (b). Indeed, this upper bound can also not be improved in general. To prove this, we construct an artificial non-monotonic logic where the problems are complete for the corresponding classes.

Theorem 6.2.6. *For* $m \geq 1$*, there is a logic* $L_{\Pi_m^p} = (\mathcal{WF}_{\Pi_m^p}, \mathcal{BS}_{\Pi_m^p}, \mathcal{INC}_{\Pi_m^p}, \mathcal{ACC}_{\Pi_m^p})$ *such that* $\textsc{Sat}_{L_{\Pi_m^p}}$ *is in* Π_m^p *and*

(a) $\text{S-INC}_{L_{\Pi_m^p}}$ *is* Π_{m+1}^p*-complete and*

(b) $\text{Min-S-INC}_{L_{\Pi_m^p}}$ *is* D_{m+1}^p*-complete.*

Note that in the previous theorems we excluded logics with $\textsc{Sat}$ in P, i. e., we required $m \geq 1$. This condition is indeed necessary as we will see in Theorem 6.2.9 below.

6.2.3 Hardness Results for Answer Set Programming

In this section, we will give the corresponding hardness results for the statements of Theorems 6.2.4 and 6.2.5 from above when instantiating the logic with ASP. We consider normal and disjunctive logic programs. Those frameworks belong to case (a) of the theorems. We will also discuss the case $m = 0$ which is excluded in Theorems 6.2.4 and 6.2.5. As it will turn out, the results do indeed not hold for $m = 0$. We will give a counterexample based on so-called *stratified* logic programs, where the satisfiability check can be performed in P.

For our formal investigation of normal and disjunctive logic programs, we consider the logics L_{ASP} and L_{ASP^*} as introduced in Section 2.5.1. Be reminded that deciding whether a given normal logic program P is consistent is NP-complete [52].

Theorem 6.2.7. *For normal logic programs,*

(a) the problem $\text{S-INC}_{L_{\text{ASP}^*}}$ *is* coNP*-complete,*

(b) the problem $\text{Min-S-INC}_{L_{\text{ASP}^*}}$ *is* D_1^p*-complete.*

Proof. Membership: The membership statements follow from Theorems 6.2.4 and 6.2.5 since $\textsc{Sat}_{L_{\text{ASP}^*}}$ is in NP.

Hardness: As mentioned above, the problem MU, i. e., checking whether a given formula ϕ in CNF is unsatisfiable, but removing one clause renders it satisfiable, is D_1^p-complete [91]. Given such a formula, we construct a program P and a subprogram H that is minimal strongly inconsistent if and only if ϕ is a "yes" instance of MU.

Let ϕ be in 3-CNF, i. e., the conjunction of $C_1, ..., C_r$ with $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$. Let $a_1, ..., a_n$ be the atoms occurring in ϕ. Let σ be the mapping translating classical negation into default negation, i. e.,

$$\sigma(l) = \begin{cases} a_i & \text{if } l = a_i \in \{a_1, ..., a_n\}, \\ \text{not } a_i & \text{if } l = \neg a_i \in \{\neg a_1, ..., \neg a_n\}. \end{cases}$$

Let P and H be the following programs:

$$
\begin{array}{ll}
P: & \\
a_1., \ldots, a_n. & \\
w_{C_k} \leftarrow \sigma(l_{k,1}). & k = 1, \ldots, r \\
w_{C_k} \leftarrow \sigma(l_{k,2}). & k = 1, \ldots, r \\
w_{C_k} \leftarrow \sigma(l_{k,3}). & k = 1, \ldots, r \\
\leftarrow \text{not } w_{C_k}. & k = 1, \ldots, r \\
H: & \\
\leftarrow \text{not } w_{C_k}. & k = 1, \ldots, r
\end{array}
$$

Intuitively, for each k the atom w_{C_k} shall witness that the clause C_k is true. That is why it can be included in an answer set whenever one of the literals $\sigma(l_{k,1}), \ldots, \sigma(l_{k,3})$ is. Note that the construction of P is polynomial. We prove the claimed hardness results, starting with the second one.

(b): We claim that H is minimal strongly inconsistent iff ϕ is a "yes" instance of MU.

"$\Rightarrow$": Assume $H \in SI_{min}(P)$. Hence, no program H' with $H \subseteq H' \subseteq P$ is consistent.

We argue that ϕ must be unsatisfiable: For the sake of contradiction, assume there is a satisfying assignment ω to the variables $a_1, \ldots, a_n$. Augment H with the fact "a_i" for the i such that $\omega(a_i) = 1$. Moreover, add all rules of the form "$w_{C_k} \leftarrow \sigma(l_{k,i})$". Since ω renders ϕ true, all constraints in H are satisfied. Thus, H' is consistent, which is a contradiction since H was assumed to be strongly inconsistent. Similarly, minimality of H in $SI(P)$ ensures that removing any conjunct from ϕ renders it satisfiable.

"$\Leftarrow$": Now assume ϕ is a "yes" instance of MU. Then, the construction as described above does not work; no super-program of H is consistent. Hence, H is strongly inconsistent. Minimality is similar, again.

Since MU is D_1^p-hard, we obtain hardness of MIN-S-INC$_{L_{\text{ASP*}}}$ in D_1^p.

(a): Since ϕ is an arbitrary formula, deciding whether it is unsatisfiable is coNP-complete in general. As argued above, H is strongly inconsistent if and only if ϕ is unsatisfiable. Hardness in coNP follows. □

In the proof of Theorem 6.2.7 a program P is constructed which can be turned into a stratified one with a minor adjustment [6]. This is particularly interesting since computing the single answer set and then checking whether it is consistent and all constraints are satisfied can be done in P as well. This observation can be turned into a proof that Theorems 6.2.4 and 6.2.5 are not applicable for $m = 0$. We recall the notion of stratification.

Definition 6.2.8 ([6]). A logic program P over a set A of atoms is called *stratified* if there is a mapping $\|\cdot\| : A \to \mathbb{N}$ such that for any rule

$$r: \quad l_0 \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n.$$

from P it holds that

- $k(l_0) \geq k(l_i)$ for $i = 1, ..., m$,
- $k(l_0) > k(l_j)$ for $j = m + 1, ..., n$.

We can now construct a simple logic that rejects any program which is not stratified. A knowledge base is going to consist of rules and integers corresponding to strata for the atoms. It is thus quite easy to see that the satisfiability check of this logic is in P. We utilize the a similar construction as in the previous proof for the hardness results and thus obtain:

Theorem 6.2.9. *There is a logic* $L_{P_{Strat}} = (\mathcal{WF}_{P_{Strat}}, \mathcal{BS}_{P_{Strat}}, \mathcal{INC}_{P_{Strat}}, \mathcal{ACC}_{P_{Strat}})$ *such that* $\text{SAT}_{L_{P_{Strat}}}$ *is in* $\mathsf{P} = \Sigma_0^p$ *and*

(a) $\text{S-INC}_{L_{P_{Strat}}}$ *is* **coNP***-complete and*

(b) $\text{MIN-S-INC}_{L_{P_{Strat}}}$ *is* D_1^p*-complete.*

More generally, if SAT is in P for a logic L, then we almost never expect verifying (minimal) strong inconsistency of $\mathcal{H} \subseteq \mathcal{K}$ to be in P as well. The intuitive reason is that we cannot non-deterministically guess supersets of $\mathcal{H}$ in P.

Let us now consider disjunctive logic programs i. e., the logic L^A_{ASP} from Section 2.5.1. Due to [52], deciding whether a given disjunctive program is consistent is Σ_2^p-complete.

Theorem 6.2.10. *For disjunctive logic programs,*

(a) the problem $\text{S-INC}_{L_{ASP}}$ *is* Π_2^p*-complete,*

(b) the problem $\text{MIN-S-INC}_{L_{ASP}}$ *is* D_2^p*-complete.*

Normal and disjunctive LPs belong to case (a) of Theorems 6.2.4 and 6.2.5 and we saw that the corresponding lower bounds can be proved as well. Moreover, the notion of stratification [6] yielded a counterexample for $m = 0$ in the theorems. The following scheme depicts a summary of the results.

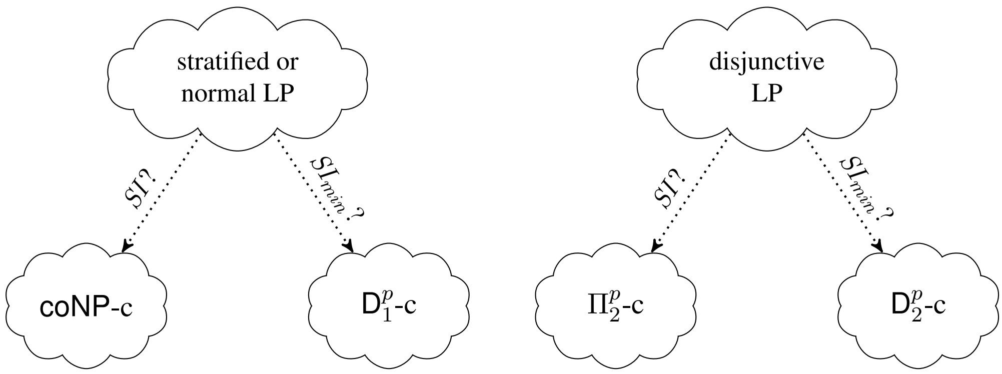


Figure 6.4: The complexity of ASP

6.2.4 Computing Strongly Inconsistent Subsets

To conclude this section, we present a generic algorithm for computing $SI(\mathcal{K})$. Algorithm 1 computes strongly inconsistent subsets in the order of decreasing cardinality, starting with $\mathcal{K}$. It is based on the observation that a proper subset S of $\mathcal{K}$ can only be strongly inconsistent if all subsets of $\mathcal{K}$ which contain one additional element are also strongly inconsistent (this property is checked during the computation of New). This additional check presumably reduces the search space in many cases, but a detailed evaluation of this algorithm is left for future work.

Input: a knowledge base $\mathcal{K}$
Result: $SI(\mathcal{K})$
$n := |\mathcal{K}|$; $H := \emptyset$; $H' := \emptyset$;
if $\mathcal{K}$ inconsistent **then** $H' := \{\mathcal{K}\}$;
while $H \neq H'$ **do**
  $n := n - 1$; $H := H'$; $New := \emptyset$;
  for *each* $S \in H$ *with* $|S| = n + 1$ **do**
    for *each* $S' \subseteq S$ *with* $|S'| = n$ **do**
      if S' inconsistent and
      $S' \cup \{\phi\} \in H$ for each $\phi \in \mathcal{K} \setminus S'$
      then $New := New \cup \{S'\}$;
    end
  end
  $H' := H' \cup New$;
end
return H.

Algorithm 1: A generic algorithm for computing $SI(\mathcal{K})$

The algorithm is somewhat reminiscent of the Apriori algorithm for computing frequent sets in data mining [1], but rather than working bottom up from smaller to bigger sets, it works in the opposite direction. The algorithm can easily be turned into one for $SI_{min}(\mathcal{K})$ by deleting non-minimal elements whenever New is added to H'.

Proposition 6.2.11. *Algorithm 1 is sound, complete, and has runtime $O(2^n {*} n {*} f(n))$ where $n = |\mathcal{K}|$ and $f(n)$ is the runtime of an algorithm for checking whether $\mathcal{K}$ is consistent.*

Proof. In order to prove soundness let H be the result of applying Algorithm 1 on $\mathcal{K}$ and let $S \in H$. We have to show that $S \in SI(\mathcal{K})$. If $S = \mathcal{K}$, then S has been added to H before the **while**-loop because $\mathcal{K}$ is inconsistent. By definition it follows $S \in SI(\mathcal{K})$. If $S \neq \mathcal{K}$, then S has been added to H at the end of the **while**-loop. This is due to the fact that S is inconsistent and, by induction, each union of S with another formula is strongly inconsistent (second **if**-statement). It follows that S is strongly inconsistent as well. For completeness, let $T \in SI(\mathcal{K})$. Then there is a chain $T = T_0 \subsetneq T_1 \subsetneq \ldots \subsetneq T_{k-1} \subsetneq T_k = \mathcal{K}$ such that $T_0, \ldots, T_k \in SI(\mathcal{K})$ (note that this statement actually holds for all such chains). As $T_k = \mathcal{K}$ is strongly inconsistent it is inconsistent as well and added to H before the **while**-loop. By induction, each T_i (in reverse order) is found in the following **while**-loops as all subsets of the given cardinality are tested for inconsistency.

Let now $f(n)$ be the runtime of an algorithm for checking consistency. First, observe that the worst-case runtime of Algorithm 1 is attained when $SI(\mathcal{K}) = 2^{\mathcal{K}} \setminus \{\emptyset\}$, i. e., all subsets of $\mathcal{K}$ (except the empty set) are strongly inconsistent. Then the first **for**-loop is iterated exactly once for each $S \in SI(\mathcal{K})$—i. e. $2^{|\mathcal{K}|} - 1 = 2^n - 1$ times— during the execution of the algorithm as it considers all sets with decreasing cardinality (note that the actual number of iterations of the outer **while**-loop is thus irrelevant for the runtime analysis). For each $S \in SI(\mathcal{K})$ we then consider each subset of S with cardinality $|S| - 1$ of which there are at most $|\mathcal{K}| = n$ many. For each of those one consistency check with runtime $f(n)$ is executed and at most $|\mathcal{K}| = n$ many member checks (of constant runtime) are performed. In total we have that Algorithm 1 has runtime $O(2^n * n * f(n))$. □

We expect that for specific logics one can do better. For instance, for logic programs without classical negation it is well-known that inconsistency can only arise if there are certain negative loops in the dependency graph. The analysis of such loops may lead to more direct algorithms. Given the results of this section it is however clear that (minimal) strong inconsistency is quite demanding in terms of computational complexity in general.

This finishes our discussion on the decision problems for minimal strong inconsistency. We found general upper bounds for them and constructed an artificial logic to witness that they cannot be improved in general. Moreover, we considered normal and disjunctive logic programs as exemplary frameworks to demonstrate how to derive the corresponding hardness results. We believe that similar results can be proved for AFs, for both credulous as well as skeptical reasoning (see Chapter 5). As already mentioned, in order to keep this chapter varied and within a reasonable space, we focus on the dual notion, namely maximal consistency when investigating the complexity of AFs.

6.3 On The Number of Strongly Inconsistent Subsets

We now address the complexity related to the *number* of strongly inconsistency subsets of a knowledge base. So instead of verifying a given subset $\mathcal{H} \subseteq \mathcal{K}$ we ask how many sets $\mathcal{H} \in SI_{min}(\mathcal{K})$ a knowledge base $\mathcal{K}$ possesses. There are several computational problems related to this question. Inspired by [108], we consider two decision problems UPPER_L and LOWER_L, and the natural function problem VALUE_L, where L is an arbitrary logic:

UPPER_L	**Input**:	$\mathcal{K} \subseteq \mathcal{WF}$, $x \in [0, \infty]$
	Output:	TRUE iff $\lvert SI_{min}(\mathcal{K})\rvert \leq x$
LOWER_L	**Input**:	$\mathcal{K} \subseteq \mathcal{WF}$, $x \in (0, \infty]$
	Output:	TRUE iff $\lvert SI_{min}(\mathcal{K})\rvert \geq x$
VALUE_L	**Input**:	$\mathcal{K} \subseteq \mathcal{WF}$
	Output:	The number $\lvert SI_{min}(\mathcal{K})\rvert$

We omit the superscripts whenever the logic L is implicit.

6.3.1 General Membership Results

Similar to Section 6.2.2 above, we start with some general membership results. Note that as in Theorems 6.2.4 and 6.2.5 the case where SAT_L is in Π_m^p for an m and L is non-monotonic is the hardest one in general.

Theorem 6.3.1. *Let* $m \geq 1$. *If the decision problem* SAT_L *is in*

(a) Σ_m^p*, then* UPPER *and* LOWER *are in* $\text{C}\Sigma_m^p$*,*

(b) Π_m^p*, then* UPPER *and* LOWER *are in* $\text{C}\Sigma_{m+1}^p$*,*

(c) Π_m^p *and L is monotonic, then* UPPER *and* LOWER *are in* $\text{C}\Sigma_m^p$.

Proof. (a): Given an integer k, $(\mathcal{K}, k)$ is a positive instance of LOWER if there are at least k minimal strongly inconsistent sets. Due to Theorem 6.2.5, deciding whether $\mathcal{H} \subseteq \mathcal{K}$ is in $SI_{min}(\mathcal{K})$ is in D_m^p. Moreover, any $\mathcal{H} \subseteq \mathcal{K}$ is of polynomial bounded size. Due to $\mathsf{CD}_m^p = \text{C}\Sigma_m^p$ (see [115]), LOWER is in $\text{C}\Sigma_m^p$. Regarding UPPER, $\text{C}\Sigma_m^p$ is closed under

complement (see [115]) and $(\mathcal{K}, k)$ is a yes instance iff $(\mathcal{K}, k+1)$ is a no instance of LOWER.
(b): Similar; here deciding whether $\mathcal{H} \in SI_{min}(\mathcal{K})$ holds is in D^p_{m+1} due to Theorem 6.2.5.
(c): Similar with decision problem in D^p_m due to Theorem 6.2.5. □

When considering the proof of the above theorem, it becomes apparent that the complexity of the *counting* problems heavily depends on the complexity of the underlying decision problems. Besides general properties of the counting polynomial hierarchy [115] the proof only makes use of the results given in Theorem 6.2.5. The same is true for the function problem *value*:

Theorem 6.3.2. *Let* $m \geq 1$. *If the decision problem* SAT_L *is in*

(a) Σ^p_m, *then* VALUE *is in* $\#\cdot\Pi^p_m$,

(b) Π^p_m, *then* VALUE *is in* $\#\cdot\Pi^p_{m+1}$,

(c) Π^p_m *and* L *is monotonic, then* VALUE *is in* $\#\cdot\Pi^p_m$.

Proof. We prove (a) only. The other cases are similar. We make use of the observation that

$$\#\cdot\Delta^p_{m+1} = \#\cdot\Pi^p_m$$

(see [64]). Furthermore, it is clear that

$$\mathsf{D}^p_m \subseteq \Delta^p_{m+1}.$$

Hence: If SAT_L is in Σ^p_m, then checking whether $\mathcal{H} \in SI_{min}(\mathcal{K})$ holds is in D^p_m (Theorem 6.2.5) and thus, VALUE is in $\#\cdot\Delta^p_{m+1} = \#\cdot\Pi^p_m$. □

This finishes our discussion regarding general membership results. As the reader may have already observed, they are quite simply corollaries of the results we already obtained for the corresponding decision problems. We want to emphasize that similar results can be shown for monotonic logics. For example, in [108, Proposition 11], it has been shown that computing $|I_{min}(\mathcal{K})|$ is in $\#\cdot\mathsf{coNP}$ (even complete under subtractive reductions). So as for the decision problems, we obtain comparable results except in case (c) of our theorems. This is too surprising since the results from this section are inferred from the corresponding results for decision problems, as we already mentioned.

6.3.2 Hardness Results for Answer Set Programming

We are going to give the corresponding hardness results for ASP. As in Section 6.2.3, ASP belongs to case (a) for both theorems, and as in Section 6.2.3, the corresponding lower bounds can indeed be proved. We start with normal logic programs. In Lemma 6.3.3 below we will give the central constructions required for the subsequent results regarding L_{ASP^*}. We will then infer those results from Lemma 6.3.3. Later on we will give the corresponding result for L_{ASP} (see Lemma 6.3.6). Inferring the analogous hardness results for disjunctive logic programs is then straightforward.

As already mentioned, the groundwork for our hardness results is the following observation. It is the required link between minimal strongly inconsistent sets of a program P to models of an open QBF Φ.

Lemma 6.3.3. *Given an open QBF $\Phi = \forall Y\, \phi(X,Y)$, there is a normal logic program $P(\Phi) \subseteq \mathcal{WF}_{ASP^*}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\textsc{Mod}(\Phi)|.$$

Proof. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. We abuse notation and write $\phi(X,Y) = 1$ ($\phi(X,Y) = 0$) if ϕ evaluates to true (false) under a given assignment to the X and Y variables, so we have

$$\begin{aligned}|\textsc{Mod}(\Phi)| &= |\{X \mid \forall Y\, \phi(X,Y) = 1)\}| \\ &= |\{X \mid \forall Y\, \neg\phi(X,Y) = 0\}|.\end{aligned}$$

We can assume ϕ to be a formula in 3-DNF and thus, $\neg\phi$ is in 3-CNF, i. e., the conjunction of $C_1, \ldots, C_r$ with $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$. Let $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m$ be fresh atoms. Intuitively, they shall correspond to the negated atoms $\neg x_1, \ldots, \neg x_n, \neg y_1, \ldots, \neg y_m$. Let σ be the appropriate mapping, i. e.,

$$\sigma(a) = \begin{cases} a & \text{if } a \in X \cup Y, \\ a' & \text{if } a \in \{\neg x_1, \ldots, \neg x_n\} \cup \{\neg y_1, \ldots, \neg y_m\}. \end{cases}$$

We construct a program $P = P(\Phi)$ whose minimal strongly inconsistent subsets correspond to $\{X \mid \forall Y\, \neg\phi(X,Y) = 0\}$. We include a fresh atom w not occurring in $X \cup Y$ which is going to witness whether $\neg\phi$ is satisfied. Hence, P contains a constraint

$$\leftarrow \text{not } w.$$

Moreover, $\neg\phi$ is satisfied if all conjuncts are true, where a conjunct $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$ is true if one of the literals occurring in C_k is true. Thus, we introduce atoms $w_1, \ldots, w_r$ and rules

$$\begin{array}{lll} w_1 \leftarrow \sigma(l_{1,1}). & \ldots & w_1 \leftarrow \sigma(l_{1,3}). \\ \ldots & \ldots & \ldots \\ w_r \leftarrow \sigma(l_{r,1}). & \ldots & w_r \leftarrow \sigma(l_{r,3}). \\ w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & & \end{array}$$

The meaning of the atoms w_{x_i} is as follows:

We want subsets of the program P to correspond to assignments to the X-variables. Thus, we introduce rules "x_i." and "x'_i." for $i = 1, \ldots, n$. A subprogram $H \subseteq P$ may contain both "x_i." and "x'_i." for an i. We want to ensure that H can only be consistent if "x_i." or "$x_i.'$" is *not* contained in H. That is the reason we need the atoms of the form w_{x_i} in the rule "$w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}$.". We include the following rules.

$$\begin{array}{ll} x_1., \ldots, x_n. & x'_1., \ldots, x'_n. \\ w_{x_1} \leftarrow \text{not } x_1. & w_{x_1} \leftarrow \text{not } x'_1. \\ \ldots & \ldots \\ w_{x_n} \leftarrow \text{not } x_n. & w_{x_n} \leftarrow \text{not } x'_n. \end{array}$$

Now, if H does not contain both "x_i." and "$x_i.'$", then the corresponding rule can be added to H to ensure w_{x_i} is entailed. This way we check that a subset $\mathcal{H} \subseteq P$ (which is either strongly inconsistent or not) corresponds to a proper assignment to the X-variables.

Moreover, we want to make sure that a strongly inconsistent subset corresponds to an assignment to *all* X-variables (and not to a partial assignment). In other words, if $H \subseteq P$ does *not* model an assignment to *all* X-variables, then H is *not* supposed to be strongly inconsistent. We thus ensure existence of a consistent superset H' with $H \subseteq H' \subseteq P$ as follows: We allow entailment of w if "x_i." or "x_i'." is missing for any i. This shall, however, only work for proper assignments, i. e., w_{x_i} is required for any i. We hence introduce a rule "$w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}$" for $i = 1, ..., n$. Now, if a subset H of P does not contain "x_i." or "x_i'." (and not both) for any $i = 1, \ldots, n$, we see that adding the corresponding rule to H ensures w is entailed, rendering H consistent. We include:

$$w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}. \qquad\qquad i = 1, ..., n$$

Since our goal is counting $|\{X \mid \forall Y\, \neg\phi(X, Y) = 0\}|$, we do not want rules corresponding to the choice of Y-variables to occur in a minimal strongly P-inconsistent subset H. The following translates assignments to Y-variables without any restrictions:

$$\begin{array}{ll} y_1 \leftarrow \text{not } y_1'. & y_1' \leftarrow \text{not } y_1. \\ ... & ... \\ y_m \leftarrow \text{not } y_m'. & y_m' \leftarrow \text{not } y_m. \end{array}$$

To summarize, P is given as follows.

$$\begin{array}{lll} P: & & \\ x_1., \ldots, x_n. & & \\ x_1'., \ldots, x_n'. & & \\ w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\ w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x_i'. & i = 1, ..., n \\ y_j \leftarrow \text{not } y_j'. & y_j' \leftarrow \text{not } y_j. & j = 1, ..., m \\ w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\ w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\ w_k \leftarrow \sigma(l_{k,3}). & & k = 1, ..., r \\ w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & & \\ \leftarrow \text{not } w. & & \end{array}$$

Note that the construction is polynomial. We show

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y\, \neg\phi(X, Y) = 0\}|.$$

We make a few observations in order to obtain this result.

(a) Any inconsistent subset of P contains the constraint "$\leftarrow$ not w." and the inconsistency stems from it.

(b) Let $H \in SI_{min}(P)$. Then, H only contains "$\leftarrow$ not w." and rules of the form "x_i." or "x_i'.".

For this, let $W \subseteq P$ be the following program

W :

$w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}.$		$i = 1, ..., n$
$w_{x_i} \leftarrow \text{not } x_i.$	$w_{x_i} \leftarrow \text{not } x'_i.$	$i = 1, ..., n$
$y_j \leftarrow \text{not } y'_j.$	$y'_j \leftarrow \text{not } y_j.$	$j = 1, ..., m$
$w_k \leftarrow \sigma(l_{k,1}).$		$k = 1, ..., r$
$w_k \leftarrow \sigma(l_{k,1}).$		$k = 1, ..., r$
$w_k \leftarrow \sigma(l_{k,3}).$		$k = 1, ..., r$
$w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}.$		

It is easy to see that rules in W can never introduce inconsistency, because they facilitate entailment of w. Hence, a set $H \in SI_{min}(P)$ consists of rules in $P \setminus W$.

(c) Let $H \in SI_{min}(P)$. If H does not contain both "x_i." and "x'_i." for an $i \in \{1, ..., n\}$, then it contains either "x_i." or "x'_i." for *all* $i \in \{1, ..., n\}$.

For this, assume neither "x_i." nor "x'_i." is in H. Augment H with the following rules:

- "$w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}.$",
- "$w_{x_j} \leftarrow \text{not } x_j.$" and "$w_{x_j} \leftarrow \text{not } x'_j.$" for $j = 1, ..., n$.

We obtain a consistent program since w is entailed now (cf. (a)). Hence, $H \notin SI(P)$.

Now, we explicitly give the two kinds of minimal strongly P-inconsistent subsets. The first one corresponds to inconsistent assignments, the second one to assignments where $\forall Y \neg\phi(X, Y) = 0$ holds.

(d) Let $i \in \{1, ..., n\}$ and let $H_i := \{x_i., \ x'_i., \ \leftarrow \text{not } w.\}$. Then, $H_i \in SI_{min}(P)$.

Since satisfying the constraint "$\leftarrow$ not w." requires the atom w_{x_i}, any program H with $H_i \subseteq H \subseteq P$ is clearly inconsistent. Hence, $H_i \in SI(P)$. For minimality, assume "x_i." is removed from H_i. Augment the obtained subprogram with the following rules:

- "$w \leftarrow \text{not } x_j, \text{not } x'_j, w_{x_1}, ..., w_{x_n}.$" for any $j \neq i$ (w.l.o.g. assume that $n \geq 2$),
- "$w_{x_j} \leftarrow \text{not } x_j.$" and "$w_{x_j} \leftarrow \text{not } x'_j.$" for $j = 1, ..., n$.

Since "x_i." is not contained in the subprogram anymore, w can be entailed now. Thus, we found a consistent superprogram. Hence, $H_i \setminus \{x_i.\} \notin SI(P)$. For the same reason, $H_i \setminus \{x'_i.\} \notin SI(P)$. The observation that $H_i \setminus \{\leftarrow \text{not } w.\} \notin SI(P)$ is trivial. Thus, H_i is minimal in $SI(P)$.

For our last step, let

$$H_I = \bigcup_{i \in \{1,...,n\}} H_i. \tag{6.4}$$

As we can conclude now, $H \in SI_{min}(P) \setminus H_I$ contains either "x_i." or "x'_i." for all indices $i \in \{1, ..., n\}$. More general, let $\mathcal{H}_\Omega \subseteq 2^P$ be the set of all subprograms of P containing either "$x_i.$" or "$x_i'.$" for all $i \in \{1, ..., n\}$. Hence for $H \in \mathcal{H}_\Omega$, it makes sense to define a corresponding assignment $\omega(H) : X \to \{0, 1\}$ with

$$\omega(H)(x_i) := \begin{cases} 1 & \text{if } x_i. \in H, \\ 0 & \text{if } x_i'. \in H. \end{cases}$$

We are now ready to prove our last step.

(e) Let $H \in \mathcal{H}_\Omega$. Then, $H \in SI_{min}(P) \setminus H_I$ if and only if $\forall Y\, (\neg\phi(X, Y) = 0)$ holds for the assignment $\omega(H)$ to the X-variables.

"$\Rightarrow$": Let $H \in SI_{min}(P) \setminus H_I$. Assume $\forall Y\, (\neg\phi(X, Y) = 0)$ is false for the assignment $\omega(H)$, i. e., $\exists Y\, \neg\phi(X, Y)$ holds. Consider the program $H \cup W$ where W is as in (b). Any answer set of W corresponds to one particular assignment to the Y-variables. By construction of P and since $\exists Y\, \neg\phi(X, Y)$ holds, $H \cup W$ has a stable model M with $w \in M$. Hence, $H \cup W$ is consistent and we conclude $H \notin SI_{min}(P)$.

"$\Leftarrow$": Minimality of H in $SI(P)$ follows from the observations we made above. Assume H' is consistent with $H \subseteq H'$. Due to (d), $H' \setminus H \subseteq W$ with W as above, because adding additional rules of the form "$x_i.$" or "$x_i'.$" to H renders the subprogram inconsistent. However, rules in W do not introduce inconsistency. Hence H' with $H \subseteq H' \subseteq H \cup W$ being consistent implies that $H \cup W$ is consistent as well. As above, we conclude $\exists Y\, \phi(X, Y)$ is true.

In (d) and (e), we found two possible cases for sets $H \in SI_{min}(P)$. Due to (b) and (c), no other case occurs. Thus,

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y\, \neg\phi(X, Y) = 0\}|$$

is proved. □

The main results follow via the above construction. We start with LOWER and UPPER.

Proposition 6.3.4. *The problems* $\text{LOWER}_{L_{\text{ASP*}}}$ *and* $\text{UPPER}_{L_{\text{ASP*}}}$ *are* CNP*-complete.*

Proof. Membership of $\text{LOWER}_{L_{\text{ASP*}}}$ and $\text{UPPER}_{L_{\text{ASP*}}}$ follows from Theorem 6.3.1, so we need to show hardness. We consider the CNP-complete problem of deciding whether $|\text{MOD}(\Phi)| \geq k$ for an open QBF $\Phi = \forall Y\, \phi(X, Y)$ (recall $\mathrm{C}\Sigma_m^p = \mathrm{C}\Pi_m^p$, see [115]). Since we can construct the program $P(\Phi)$ as in Lemma 6.3.3 in P, we already found a polynomial reduction: $\text{MOD}(\Phi) \geq k$ if and only if $SI_{min}(P) \geq k + |X|$. □

Next we show that VALUE is $\#{\cdot}\mathsf{coNP}$-complete under subtractive reductions. For $\#{\cdot}\mathsf{coNP}$, computing $|\text{MOD}(\Phi)|$ for an open QBF $\Phi = \forall Y\, \phi(X, Y)$ is the generic complete problem under subtractive reductions. Recall the idea behind this kind of reduction: We first overcount the value we actually aim at. Then, we correct this by subtracting unintended items. Consequently, the above construction where we found a program $P = P(\Phi)$ satisfying $SI_{min}(P)| = |X| + |\text{MOD}(\Phi)|$ is a suitable starting point for a subtractive reduction. We have left to find a program P' with

- $SI_{min}(P') \subseteq SI_{min}(P)$,
- $\text{MOD}(\Phi) = SI_{min}(P) - SI_{min}(P')$.

Given these programs P and P' we have proved that $\textsc{Mod}(\Phi)$ can be computed for an open QBF $\Phi = \forall Y\, \phi(X,Y)$ via $|SI_{min}(P)| - |SI_{min}(P')|$ in #·coNP (see [50]). This yields completeness under subtractive reductions.

Theorem 6.3.5. *The problem* $\textsc{Value}_{L_{ASP*}}$ *is #·coNP-complete under subtractive reductions.*

Proof. As membership is due to Theorem 6.3.2, we prove hardness. Consider an open QBF $\Phi = \forall Y\, \phi(X,Y)$. We use the same construction (and the same notation for ϕ, X and Y) as in Lemma 6.3.3, i. e., consider $P = P(\Phi)$ given as follows:

P :

$x_1., \ldots, x_n.$

$x'_1., \ldots, x'_n.$

$w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}.$ $\quad i = 1, ..., n$

$w_{x_i} \leftarrow \text{not } x_i.$ $\quad w_{x_i} \leftarrow \text{not } x'_i.$ $\quad i = 1, ..., n$

$y_j \leftarrow \text{not } y'_j.$ $\quad y'_j \leftarrow \text{not } y_j.$ $\quad j = 1, ..., m$

$w_k \leftarrow \sigma(l_{k,1}).$ $\quad k = 1, ..., r$

$w_k \leftarrow \sigma(l_{k,1}).$ $\quad k = 1, ..., r$

$w_k \leftarrow \sigma(l_{k,3}).$ $\quad k = 1, ..., r$

$w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}.$

$\leftarrow \text{not } w.$

Recall that P yields $|SI_{min}(P(\Phi))| = |X| + |\textsc{Mod}(\Phi)|$. So for our subtractive reduction we require a program P' with $|SI_{min}(P')| = |X|$ and $SI_{min}(P') \subseteq SI_{min}(P)$. For this, consider $P' = P'(\Phi)$ as follows:

P' :

$x_1., \ldots, x_n.$

$x'_1., \ldots, x'_n.$

$w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}.$ $\quad i = 1, ..., n$

$w_{x_i} \leftarrow \text{not } x_i.$ $\quad w_{x_i} \leftarrow \text{not } x'_i.$ $\quad i = 1, ..., n$

$\leftarrow \text{not } w.$

It is easy to see that $SI_{min}(P') = H_I$ with H_I as in (6.4). In particular,

$$SI_{min}(P') \subseteq SI_{min}(P)$$

and $|SI_{min}(P')| = |X|$. Since

$$\begin{aligned} |\{X \mid \forall Y\, \phi(X,Y)\}| &= |SI_{min}(P)| - |X| \\ &= |SI_{min}(P)| - |SI_{min}(P')| \end{aligned}$$

follows, we found a subtractive reduction. □

This ends our discussion pertaining to L_{ASP^*}. As already mentioned, the case L_{ASP} is similar but involves going up one level in the corresponding hierarchy. Analogously, the fundamental step is constructing a program P with $|SI_{min}(P(\Phi))| = |X| + |\text{MOD}(\Phi)|$ for an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$. The construction is rather similar: As we already did in Section 6.2.3 we augment our previous construction with features from the program in [52] which is used to prove Σ_2^p-completeness of the satisfiability check for disjunctive logic programs. Then, the subsequent steps are as above. Due to the technical similarities, we only sketch the proof of the following results in Appendix A:

Lemma 6.3.6. *Given an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$, there is a disjunctive logic program $P(\Phi) \subseteq \mathcal{WF}_{ASP}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\text{MOD}(\Phi)|.$$

Proposition 6.3.7. *The problems* $\text{LOWER}_{L_{ASP}}$ *and* $\text{UPPER}_{L_{ASP}}$ *are* $\mathsf{C}\Sigma_2^p$*-complete. The problem* $\text{VALUE}_{L_{ASP}}$ *is* $\#\cdot\Pi_2^p$*-complete under subtractive reductions.*

The following scheme summarizes the results of this section.

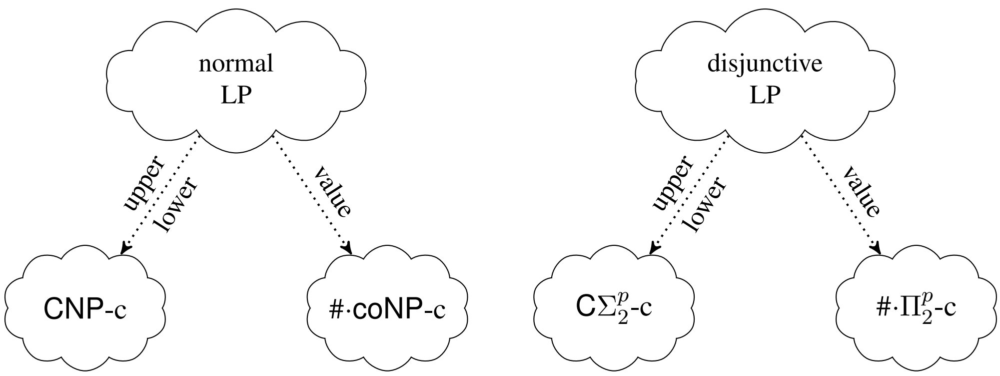


Figure 6.5: The complexity of counting in ASP

We want to emphasize the similarities to Section 6.2: Again the general membership results are rather generic and quite easy to see. Moreover, for our exemplary framework ASP we could prove the corresponding lower bounds as before. When comparing these results to the complexity analysis of inconsistency measures performed in [108] we see that counting minimal *strongly* inconsistent subsets is not more demanding than the same problem for ordinary inconsistency. More precisely, in [108, Proposition 11], it has been shown that computing $|I_{min}(\mathcal{K})|$ is $\#\cdot\mathsf{coNP}$-complete under subtractive reductions if $\mathcal{K}$ is a propositional knowledge base. Since satisfiability of propositional logic is NP-complete as well, we do not expect the analogous problem for normal logic programs to be easier. In fact, one might anticipate an increase in computational complexity since computing $|SI_{min}(\mathcal{K})|$ appears to be more demanding than computing $|I_{min}(\mathcal{K})|$.

The membership results from Section 6.2 show that this intuition should be taken with a grain of salt, at least in our setting of a *worst case* analysis. As it turns out, for normal logic programs computing $|SI_{min}(P)|$ is $\#\cdot\mathsf{coNP}$-complete under subtractive reductions as it is the case for $|I_{min}(\mathcal{K})|$. This is true even though ASP is non-monotonic, counting complexity classes are hard to define and their closure properties are not quite satisfying, calling for notions like *subtractive reductions* which appear unusual at a first glance. Nonetheless, $SI_{min}(P)$ and $I_{min}(\mathcal{K})$ remain comparable even up until this point.

6.4 Consistent Argumentation

In this section we discuss the computational complexity of two decision problems that naturally arise in view of Chapter 5 about inconsistency in AFs, namely the existence problem as well as the verification problem regarding minimal repairs. We mostly restrict our investigation to $\sigma \in \{gr, co, pr, stb\}$. We also consider argument-based diagnoses only. We believe this suffices to demonstrate the reader how to derive the results. Formally, we consider the following decision problems:

EX-MIN-REPAIR$_{\sigma,\diamond}$
Input: An AF F
Output: TRUE iff there is a minimal σ-$\diamond$-diagnosis for F

VER-MIN-REPAIR$_{\sigma,\diamond}$
Input: $(F, \mathcal{S})$ where $F = (A, R)$ is an AF and $\mathcal{S} \subseteq A$
Output: TRUE iff $\mathcal{S}$ minimal σ-$\diamond$-diagnosis for F

We start with the problem of deciding whether a minimal repair exists. As we know from Theorems 5.2.16, 5.2.19 and Fact 5.2.21 (see Section 5.2 about existence of repairs) it suffices for both reasoning modes to perform a simple syntactical check, namely whether the AF in question is self-controversial. This can be done in linear time.

Proposition 6.4.1. *For $\sigma \in \{ad, gr, eg, il, ss, pr, co, stb\}$ and $\diamond \in \{cred, skep\}$ the problem* EX-MIN-REPAIR$_{\sigma,\diamond}$ *can be solved in linear time.*

We turn to the problem VER-MIN-REPAIR$_{\sigma,\diamond}$, which will turn out to be more demanding in most cases. The hardness results we give in the subsequent subsections are oftentimes adjustments of existing constructions (see [51]). Membership results are a corollary of the following observation, similar to Theorem 6.2.5 which was concerned about minimal strong inconsistency instead of maximal consistency.

Proposition 6.4.2. *Let $\diamond \in \{cred, skep\}$. Let σ be any semantics. If deciding whether an AF F is consistent wrt. σ and $\diamond$ is in Σ_m^p for any integer $m \geq 1$, then* VER-MIN-REPAIR$_{\sigma,\diamond}$ *is in D_m^p. If deciding whether an AF F is consistent wrt. σ and $\diamond$ is in Π_m^p for any integer $m \geq 1$, then* VER-MIN-REPAIR$_{\sigma,\diamond}$ *is in Π_{m+1}^p.*

Proof. If checking consistency is in Σ_m^p, then we check in Σ_m^p whether we are given a repair $F_{\mathcal{S}}$ and for minimality we non-deterministically guess a subset $\mathcal{S}'$ and verify that $F_{\mathcal{S}'}$ is not a repair in Π_m^p, which needs to be the case for every $\mathcal{S}' \subseteq \mathcal{S}$. In summary, this procedure is in D_m^p. The other case is similar. □

6.4.1 Grounded Semantics

Given an AF (and a potential diagnosis), we know that the grounded extension is non-empty if and only if there is an argument which is not attacked. Thus, verifying that a given set is a gr-diagnosis is quite easy. It turns out that minimality is tractable as well.

Proposition 6.4.3. *For $\diamond \in \{cred, skep\}$, the problem* VER-MIN-REPAIR$_{gr,\diamond}$ *is in P.*

Proof. If $F_{\mathcal{S}}$ contains no unattacked argument, we reject since it is no repair. For minimality, we check whether there is $\alpha \in \mathcal{S}$ st. $F_{\mathcal{S}\setminus\{\alpha\}}$ has an unattacked argument as well. □

We want to mention that we can even compute *all* gr-diagnoses in P. We believe this observation is relevant since the grounded repairs play an essential role as the results from Section 5.2 suggest. Assume we are given the AF $F = (A, R)$ with $gr(F) = \{\emptyset\}$. Since a grounded diagnosis needs to ensure that at least one argument $a \in A$ is not attacked anymore, we can successively look at any $a \in A$ and consider $\mathcal{S} = \{b \in A \mid (b, a) \in R\}$. If $\mathcal{S}$ is a minimal gr-diagnosis (verification is in P due to Proposition 6.4.3), we add $\mathcal{S}$ to our list. Since there are at most $|A|$ gr-diagnoses, this procedure is in P.

Proposition 6.4.4. *Computing all gr-diagnoses of a given AF F can be done in P.*

Although finding a σ-diagnosis may become rather hard depending on σ, we can hence efficiently compute all gr-diagnoses and then utilize Lemmata 5.2.15, 5.2.18 and 5.2.22 in order to reduce the search space. This approach explains the central role of grounded semantics. In a nutshell, the gr-repairs can be seen as a (polynomial time computable) starting point in order to find minimal repairs for other semantics. A thorough investigation of this approach is part of future work.

6.4.2 Universally Defined Semantics

Considering the computational complexity of different reasoning problems in AFs, it is quite unsurprising that VER-MIN-REPAIR$_{\sigma,cred}$ is intractable for most semantics σ as it requires checking whether a *non-empty* extension exists. Due to the additional minimality check we require, the decision problem turn out to be in the corresponding difference class for the universally defined semantics we consider.

Theorem 6.4.5. VER-MIN-REPAIR$_{\sigma,cred}$ *is D_1^p-complete for $\sigma \in \{ad, pr, co\}$.*

Proof. Membership is due to Proposition 6.4.2. For hardness, we reduce the problem MC which is analogous to MU from above:

MC
Input: (ϕ, ϑ) where ϕ is a formula in 3-CNF and $\vartheta \subseteq \phi$
Output: TRUE iff ϑ is satisfiable, but any formula ϑ' with $\vartheta \subsetneq \vartheta' \subseteq \phi$ is not.

The problem MC is D_1^p-complete [90].

Let ϕ be in 3-CNF, i. e., the conjunction of $C_1, ..., C_r$ with $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$. Let $x_1, ..., x_n$ be the literals occurring in ϕ, set $\neg\neg x_i = x_i$. We can prove hardness utilizing a minor adjustment of the standard construction (see [51]) depicted in Figure 6.6:

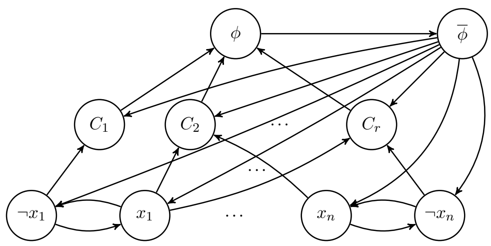


Figure 6.6: Standard Construction for 3-SAT reductions

Let F be the AF $F = (A, R)$ with $A = \{x_1, \neg x_1, \ldots, x_n, \neg x_n, C_1, \ldots, C_r, \phi, \overline{\phi}\}$ and

$$\begin{aligned} R = &\{(x_i, \neg x_i) \mid i \in \{1, \ldots, n\}\} \cup \{(\neg x_i, x_i) \mid i \in \{1, \ldots, n\}\} \\ &\cup \{(x_i, C_j) \mid x_i \text{ occurs in } C_j\} \cup \{(C_j, \phi) \mid j \in \{1, \ldots, r\}\} \\ &\cup \{(\phi, \overline{\phi})\} \cup \{(\overline{\phi}, x_i) \mid x_i \text{ occurs in } \phi\} \cup \{(\overline{\phi}, C_j) \mid j \in \{1, \ldots, r\}\} \end{aligned}$$

Consider $\sigma = ad$. It is well-known that ϕ is satisfiable iff there is a non-emtpy admissible extension of the framework depicted in Figure 6.6. The reader may verify that a non-empty admissible extension E needs to contain some of the X arguments. In order to defended them, $\phi \in E$ is required. In order to find an admissible set of arguments defending ϕ, the formula needs to be satisfiable. Now let ϑ be a subformula of ϕ, i. e., ϑ is w. l. o. g. of the form $\vartheta = \{C_1, \ldots, C_s\}$. Then, (ϕ, ϑ) is "yes" instance of MC iff $(F, \mathcal{S})$ with $\mathcal{S} = \{C_{s+1}, \ldots, C_r\}$ is a "yes" instance of VER-MIN-REPAIR$_{ad,cred}$.

Clearly, any framework possesses a non-empty admissible extension iff this is the case for $\sigma = co$ and $\sigma = pr$ □

We turn to skeptical reasoning. First observe that for ad semantics, the problem is trivial since any framework possess an empty admissible extension.

Recall that the unique grounded extension of an AF F is complete as well. Moreover we have $gr(F) \subseteq \bigcap co(F)$. Hence, any framework F possesses a skeptically accepted argument wrt. grounded semantics if and only if this is the case for complete semantics. Hence, applying Proposition 6.4.3 yields:

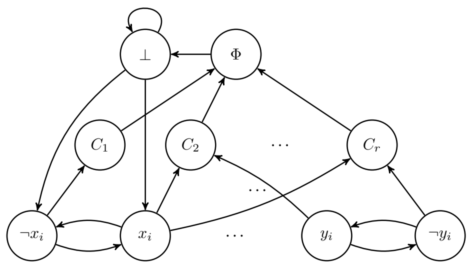
Corollary 6.4.6. VER-MIN-REPAIR$_{co,skep}$ *is in P.*

Now we consider $\sigma = pr$. Recall that deciding whether an argument is skeptically accepted is Π_2^p-complete [51]. Given a framework F and a set $\mathcal{S} \subseteq A$ of arguments, the decision problem VER-MIN-REPAIR$_{pr,skep}$ involves checking whether for all $\mathcal{S}'$ with $\mathcal{S}' \subseteq \mathcal{S}$ the framework $F_{\mathcal{S}'}$ does *not* possess a skeptically accepted argument. Since the latter check is in Σ_2^p for each $\mathcal{S}'$, we immediately see a Π_3^p upper bound for VER-MIN-REPAIR$_{pr,skep}$ due to Proposition 6.4.2. So the main work for the following theorem is the lower bound:

Theorem 6.4.7. VER-MIN-REPAIR$_{pr,skep}$ *is* Π_3^p*-complete.*

Proof. Membership is due to Proposition 6.4.2. Recall the following construction from [51] with the property that the AF accepts an argument skeptically wrt. preferred semantics if and only if a formula $\Phi = \forall Y \, \exists X : \phi(X, Y)$ in CNF evaluates to true.

$\mathcal{H}_{\Pi_2^p}$:



In order to prove hardness in Π_3^p for our problem we need to simulate an additional quantifier. So let us assume we are given a formula $\Psi = \exists Z \, \forall Y \, \exists X : \psi(X, Y, Z)$ in CNF. We augment the above construction from [51] with the intention that Ψ evaluates to true if and only if $(F, \mathcal{S})$ is a "no" instance of VER-MIN-REPAIR$_{pr,skep}$ (that is, there exists a subset $\mathcal{S}'$ with $\mathcal{S}' \subsetneq \mathcal{S}$ such that $F_{\mathcal{S}'}$ possesses a skeptically accepted argument wrt. preferred semantics).

Thereby we will construct an AF $F = (A, R)$ with a set $\mathcal{S} \subseteq A$ of arguments with the following properties:

- $F_{\mathcal{S}}$ itself is consistent, i. e., there is a skeptically accepted argument,
- subsets of $\mathcal{S}$, i. e., the sets $\mathcal{S}' \subseteq \mathcal{S}$ may correspond to assignments to the Z-variables,
- there is one $\mathcal{S}'$ with $\mathcal{S}' \subsetneq \mathcal{S}$ such that $F_{\mathcal{S}'}$ is consistent if and only if the formula $\Psi = \exists Z\, \forall Y\, \exists X : \psi(X, Y, Z)$ evaluates to true.

The first and the last item together ensure that $(F, \mathcal{S})$ is a positive instance of the decision problem VER-MIN-REPAIR$_{pr,skep}$ if and only if the formula Ψ is false.

Before depicting and explaining our construction, we name all arguments occurring in the AF. We hope this improves readability of the proof. Which attacks we include will be explained later. Our framework is $F = (A, R)$ with

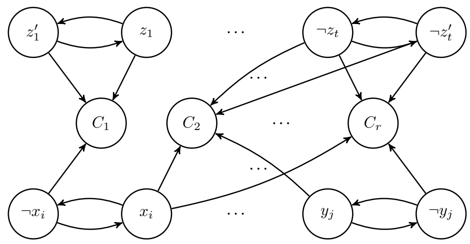
$$A = \{x_1, \neg x_1, \ldots, x_n, \neg x_n, y_1, \neg y_1, \ldots, y_m, \neg y_m, z_1, \neg z_1, z_1', \neg z_1', \ldots, z_t, \neg z_t, z_t', \neg z_t',$$
$$C_1, \ldots, C_r, D_{1,1}, \ldots, D_{1,4}, \ldots, D_{t,1}, \ldots, D_{t,4}, \Phi, \top, \bot\}$$

Moreover, we set

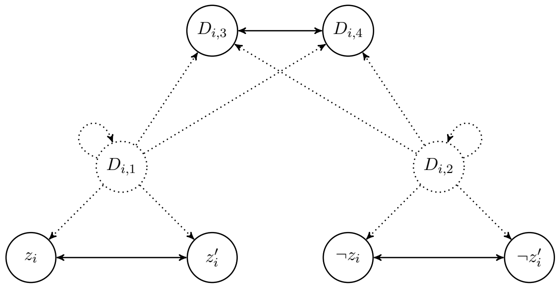
$$\mathcal{S} = \{D_{1,1}, D_{1,2}, \ldots, D_{t,1}, D_{t,2}\}$$

We will see that sets $\mathcal{S}'$ with $\mathcal{S}' \subseteq \mathcal{S}$ induce AFs $F_{\mathcal{S}'}$ with the intuitive meaning that values to Z variables are assigned.

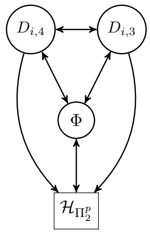
So first we consider Z variables $z_1, \neg z_1, \ldots, z_t, \neg z_t$ which attack the $C_1, \ldots, C_r$ in the natural way: We have $(z_j, C_i) \in R$ iff z_j occurs in the clause C_i and $(\neg z_j, C_i) \in R$ iff $\neg z_j$ occurs in the clause C_i. We also consider copies $z_1', \neg z_1', \ldots, z_t', \neg z_t'$.



The reason for the copies $z_1', \ldots, \neg z_t'$ is to ensure that the Z arguments themselves are not skeptically accepted. Now consider the following gadged, which will be included for any Z variable.



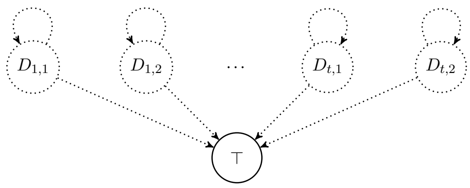
The dummy arguments $D_{i,1}$ and $D_{i,2}$ as well as their attacks are depicted with dotted lines to illustrate that they do *not* occur in $F_{\mathcal{S}}$, as those arguments belong to $\mathcal{S}$. Augmenting $F_{\mathcal{S}}$ with $D_{i,1}$, for example, ensures that z_i and z_i' are never defended and thus occur in no preferred extension. Hence, this choice corresponds to letting z_i be *false*. The role of the $D_{i,3}$ and $D_{i,4}$ becomes apparent considering the following arguments:



Now, for any $i \in \{1, \ldots, t\}$ we observe: Since $D_{i,3}$ and $D_{i,4}$ attack all arguments in $\mathcal{H}_{\Pi_2^p}$ we have that $\{D_{1,3}, \ldots, D_{t,3}\}$ and $\{D_{1,4}, \ldots, D_{t,4}\}$ are two preferred extensions for any $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ which contains *both* $D_{i,1}$ *and* $D_{i,2}$, i. e., in $F_{\mathcal{S}'}$ occur *neither* $D_{i,1}$ *nor* $D_{i,2}$. Hence, the intersection $\bigcap pr(F_{\mathcal{S}'})$ is empty. We thus see: A framework $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ can only possess a skeptically accepted argument if for each $i \in \{1, \ldots, t\}$ it is not true that both $D_{i,1}$ and $D_{i,2}$ occur in $\mathcal{S}'$.

Now assume this is given, i. e., we have a framework $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ as described. Recall that the choice of the $D_{i,1}$ and $D_{i,2}$ naturally corresponds to a (partial) assignment $\omega : Z \to \{0, 1\}$. As in the original construction $\mathcal{H}_{\Pi_2^p}$ we see that Φ is skeptically accepted iff $\forall Y \, \exists X : \psi(X, Y, Z)$ evaluates to true under the assignment $\omega : Z \to \{0, 1\}$. In this case, $F_{\mathcal{S}'}$ is consistent. Since this applies to any $\mathcal{S}'$ of the form described above we see: Every $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ is inconsistent iff for any assignment $\omega : Z \to \{0, 1\}$ the formula $\forall Y \, \exists X : \psi(X, Y, Z)$ evaluates to false iff the formula $\exists Z \, \forall Y \, \exists X : \psi(X, Y, Z)$ evaluates to false.

To summarize, we have established: $\exists Z \, \forall Y \, \exists X : \psi(X, Y, Z)$ is false iff $F_{\mathcal{S}'}$ is inconsistent for all $\mathcal{S}' \subsetneq \mathcal{S}$. The latter *nearly* means that $(F, \mathcal{S})$ for $\mathcal{S} = A \setminus H$ is a positive instance of VER-MIN-REPAIR$_{pr,skep}$. Finally, we make sure that $F_{\mathcal{S}}$ itself is consistent, i. e., there is a skeptically accepted argument. The following final gadget does the job:



There is no other argument attacking $\top$. Hence, as long as no $D_{i,1}$ resp. $D_{i,2}$ argument is chosen, $\top$ is skeptically accepted. As soon as a proper superset of H is under consideration, $\top$ can never be defended and is thus rendered pointless. □

This finishes our discussion on the universally defined semantics. In summary, we obtained the following results:

- for credulous reasoning, we find a D_1^p lower bound for all cases,
- skeptical reasoning trivializes for ad semantics, is tractable for co semantics (since it coincides with gr semantics) and is quite demanding for pr semantics (Π_3^p-complete).

6.4.3 Stable Semantics

Let us now turn to our only example for collapsing semantics, namely $\sigma = stb$. If we are interested in credulous reasoning, we face a similar situation (and a similar proof which can be found in Appendix A) as in Theorem 6.4.5.

Theorem 6.4.8. $\textsc{Ver-Min-Repair}_{stb,cred}$ *is* D_1^p*-complete.*

We turn to skeptical reasoning. Since finding a stable extension is NP-complete, it is not hard to see that there is a coNP lower bound for skeptical reasoning. However, as the framework in question might collapse, we also need to verify that there is *at least one* stable extension of a given framework. The result is a D_1^p lower bound (see [93]). Interestingly, this observation is not relevant in our case. The coNP lower bound is already responsible for $\textsc{Ver-Min-Repair}_{stb,skep}$ to have a Π_2^p lower bound: Given $\mathcal{H} \subseteq \mathcal{K}$ the decision problem $\textsc{Ver-Min-Repair}_{stb,skep}$ involves checking whether *all* sets $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ do *not* possess any skeptically accepted argument. Since the latter test has a NP lower bound, we have a Π_2^p lower bound for $\textsc{Ver-Min-Repair}_{stb,skep}$. More precisely:

Theorem 6.4.9. $\textsc{Ver-Min-Repair}_{stb,skep}$ *is* Π_2^p*-complete.*

The following table summarizes the results for the decision problem $\textsc{Ver-Min-Repair}_{\sigma,\diamond}$ we obtained. Except for gr semantics, credulous reasoning yields D_1^p-completeness for all considered semantics. Due to special properties of semantics, the upper bound from Proposition 6.4.2 is not always a lower bound for skeptical reasoning.

	credulous	skeptical
gr	P	P
ad	D_1^p-c	trivial
pr	D_1^p-c	Π_3^p-c
co	D_1^p-c	P
stb	D_1^p-c	Π_2^p-c

Table 6.1: Complexity of verifying maximal consistency in AFs

6.5 Conclusion and Related Work

In this chapter, we investigated the computational complexity of decision and function problems related to strong inconsistency. We placed value on a comparison between this refined notion and ordinary inconsistency in monotonic logics. We argued that the this comparison is not meaningful if the satisfiability check for the logic is in PSPACE (or more demanding) and thus restricted the investigation to logics where this problem is in the polynomial hierarchy. For this, we extended a result [91] about the problem MU to our generalized version QBF-MU$(Q_1, ..., Q_m)$. We then gave general upper bounds for the decision problems of verifying (minimal) strong inconsistency. Although they appear rather generic, we showed that they cannot be improved in general. We also demonstrated how to infer the corresponding lower bounds for ASP. In a similar fashion, we investigated the problem of computing $|\mathcal{I}_{\mathsf{MSI}}(\mathcal{K})|$, i. e., the number of minimal strongly inconsistent subsets of a given knowledge base. Our last step was a similar investigation of inconsistency in AFs, where we focused on maximal consistency rather than minimal (strong) inconsistency.

The results of this chapter suggest that in terms of a worst case analysis, strong inconsistency is in many cases not more demanding than mere inconsistency. This result does not transfer to logics where the satisfiability check is in P since this class does not allow for a non-deterministic guess of supersets of a given $\mathcal{H} \subseteq \mathcal{K}$. Similar observations can be made for the corresponding counting problems. We did not consider the notions concerned with adding formulas to a knowledge base, for example bidirectional non-repairs. It is easy to see that similar results transfer to these notions, which we did not state explicitly here in order to keep our investigation concise.

We already mentioned that Reiter [96] was also concerned about computing hitting sets. Many algorithms and systems for enumerating minimal inconsistent sets –[8; 78; 79]– build on the duality results. Hitting sets are also utilized in computation of causes and responsibilities of inconsistency in databases [98]. Our discussion regarding the complexity of minimal unsatisfiability for QBFs was inspired by [91]. A rather thorough discussion about the complexity of inconsistency measures has been done in [107; 108], which also discusses various problems about minimal inconsistent and maximal consistent sets of a given knowledge base. The paper [46] discusses computational complexity as well.

Chapter 7

Conclusion

7.1 Summary

In this work we studied inconsistency in an abstract setting covering arbitrary logics, in particular non-monotonic ones. We demonstrated that in the general case the standard notion of inconsistency is unable to play the same role it does in monotonic reasoning. One of our main contributions is the identification and investigation of an adequate strengthening of inconsistency. Our main results can be summarized as follows:

In *Chapter 3* we focused on structural properties of knowledge bases, especially the connection between consistent and inconsistent subsets. We gave a generalization of Reiter's well-known hitting set duality to non-monotonic logics (Theorem 3.1.12). Reiter's duality – tailored for a monotonic setting– is only concerned about *removing* formulas. We also gave a duality characterization for repairs which can be obtained by adding (Theorem 3.2.9) or adding *and* removing formulas (Theorem 3.2.32). We demonstrated structural properties of knowledge bases, which are themselves interesting, not just as tools for proving the main theorems (cf. Propositions 3.2.25 and 3.2.27 as well as Propositions 3.2.37 and 3.2.40). We also considered more fine-grained modifications like strengthening and weakening instead of adding and removing formulas (Corollary 3.3.13). Using several examples we illustrated that infinite knowledge bases are in general not as well-behaved as finite ones. We identified sufficient conditions in order to overcome some of the arising issues, most notably the so-called *compactness property* (see Theorem 3.4.15).

We devoted *Chapter 4* to measuring inconsistency in non-monotonic logics. We introduced inconsistency measures for this setting –$\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$– which are natural generalizations of measures from the literature. In order to help assessing the quality of our measures, we refined existing rationality postulates to obtain meaningful ones for non-monotonic logics. Thereby, the most important ones were the four postulates for a *basic inconsistency measure* (see [68]). As a result, we proposed *strong monotony* which is similar to the *monotony* postulate, but requires an additional premise. Based on two refined notions of *free formulas* we considered *SI-free* and *Independence*, and we argued that *dominance* is not meaningful in non-monotonic logics. We analyzed the compliance of our generalized measures with the refined rationality postulates. Our investigation continued with the question how to assess inconsistencies of a knowledge base within the context of a larger one. Interestingly, a well-behaved approach was based on our notion of bidirectional non-repairs (see Definition 3.2.19). We also discussed measuring inconsistency in ASP as a special case of our definition of a general logic.

The results of this chapter suggest that the existing work on measuring inconsistency in propositional knowledge bases can be extended to non-monotonic logics, when considering appropriate adjustments to the established approaches. Interesting novel questions arise, because additional information may resolve conflicts. However, one needs to accept that not all aspects can be covered by a general definition of a logic. Consideration of particular frameworks cannot easily be subsumed if a thorough understanding of inconsistency is desired.

In *Chapter 5* we investigated inconsistency in AFs, with respect to various semantics and the two standard reasoning modes, credulous and skeptical. We pointed out that existence of repairs is guaranteed in most cases (see Theorem 5.2.16, Theorem 5.2.19 and Fact 5.2.21). We investigated the relations between repair notions (see Theorems 5.2.4 and 5.2.5 and in particular Conjecture 5.2.5, Theorem 5.2.8 and the examples we gave). We demonstrated how our previous results yield duality characterizations for repairs of AFs (Propositions 5.3.3 and 5.3.4). In order to refine our analysis, we considered specific situations, e.g., symmetry of the attack relation (see Proposition 5.4.6) or splitting (Propositions 5.4.13 and 5.4.14). In a brief discussion on infinite AFs our main result was proving the co-compactness property (introduced in Section 3.4) for finitary AFs (see Theorem 5.4.26). We also performed a short case study, illustrating how to repair some given AFs.

This chapter shall demonstrate how to apply our techniques to tackle inconsistency, when given a specific logic. We found connections between the repair notions and were able to apply our duality results to characterize repairs of AFs, independent of the underlying semantics or reasoning mode. We gave encouraging connections to subclasses of AFs and splitting results from the literature.

In *Chapter 6* we performed an analysis of the computational complexity of related decision and function problems. For this, we established results for the generic monotonic framework of QBFs (Theorem 6.2.2). We found that in many cases, strong inconsistency is not more demanding than the ordinary one, from a computational point of view (Theorem 6.2.5). In addition to general upper bounds, we gave corresponding lower bounds for ASP (Theorems 6.2.7 and 6.2.10). We extended the investigation to the corresponding problem of counting strongly inconsistent subsets. Since the results can be derived from the corresponding decision problems, strong inconsistency is again comparable to mere inconsistency in many cases (Theorems 6.3.1 and 6.3.2). In Section 6.4 we gave complexity results for verifying minimal repairs for AFs, covering various semantics and the two reasoning modes considered in Chapter 5.

7.2 Future Work

Several future directions arise considering restrictions and choices we made in this thesis. First, the following aspects were only discussed for removing formulas from a knowledge base $\mathcal{K}$ and could be extended to a setting which takes additional information into account:

- infinite knowledge bases (Section 3.4),
- inconsistency in AFs wrt. different semantics and reasoning modes (Chapter 5),
- computational complexity (Chapter 6).

Second, we considered measuring inconsistency in ASP, but not in AFs. Chapter 5 points out several aspects that are worth taking into account when investigating inconsistency in

AFs. An in-depth discussion on inconsistency measurement in abstract argumentation appears to be a promising research direction. Although we discussed inconsistency measures for ASP, we did not perform an investigation of different reasoning modes as we did for AFs. Doing so would not only be interesting on its own, but (due to the close link between ASP and stable semantics) probably also yield insights in order to find a proof or a counterexample for Conjecture 5.2.7. More precisely, as shown in [101, Theorem 4.13] there is a standard translation T from AFs to LPs, st. for any AF F, $\sigma(F)$ coincides with $\tau(T(F))$ for certain pairs of semantics σ and τ. This means, one interesting research question is to which extent our results for AFs can be conveyed to repairing in ASP and vice versa. Independent of ASP, Conjecture 5.2.7 appears to be one of the most exciting open problems regarding our investigation of AFs. One could also cover additional argumentation semantics. A further intensive study of subclasses of AFs seems to be very promising since certain useful semantical properties are already ensured by syntactic properties.

Our discussion on measuring inconsistency is mostly restricted to adjusted versions of three measures from the literature. All of them are based on (the number of) minimal strongly inconsistent subsets of a knowledge base $\mathcal{K}$. It would of course be interesting to extend this investigation to a wider range of inconsistency measures. We briefly demonstrated that the classification of inconsistency measures proposed in [32] can be generalized to non-monotonic logics, at least to a certain extent. A simple corollary of the result in [32] was that all our considered measures can be defined as functions on the strong inconsistency graph. It would be interesting to find meaningful measures for non-monotonic logics that are no SIG measures. We also mentioned that measuring inconsistency in this general setting would probably greatly benefit from information measures for non-monotonic logics. The reason is that some conflicts can be resolved by adding information, but this is hard to formalize if the added information cannot be assessed appropriately. An extension of the discussion on inconsistency values similar in spirit to [68] would probably also benefit from tools to measure information.

The analysis of the computational complexity covered hardness results for minimal strong inconsistency in ASP and maximal consistency in AFs. One could complete this picture by discussing the remaining two cases. Moreover, conceiving concrete algorithms to compute minimal strongly inconsistent or maximal consistent sets would amplify our investigation. For example, our identification of grounded repairs for diagnoses in AFs appears to be a promising starting point. Another direction for future work is the application of strong inconsistency for unsatisfiable core analysis[1] in reasoning algorithms. Works such as [2; 3] use the classical notion of minimal inconsistency to determine models in non-monotonic formalisms such as answer set programming and circumscription. Using strong inconsistency instead might boost performance further in these settings.

[1]An unsatisfiable core is a minimal inconsistent set of formulas in conjunctive normal form

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases*, pages 487–499, 1994. URL http://www.vldb.org/conf/1994/P487.PDF.

[2] Mario Alviano. Model enumeration in propositional circumscription via unsatisfiable core analysis. *TPLP*, 17(5-6):708–725, 2017. doi: 10.1017/S1471068417000278. URL https://doi.org/10.1017/S1471068417000278.

[3] Mario Alviano and Carmine Dodaro. Anytime answer set optimization via unsatisfiable core shrinking. *Theory and Practice of Logic Programming*, 16(5-6):533–551, 2016.

[4] Leila Amgoud and Jonathan Ben-Naim. Measuring disagreements in argumentation graphs. In *Proceedings of the 11th International Conference on Scalable Uncertainty Management (SUM'17)*, 2017.

[5] Meriem Ammoura, Yakoub Salhi, Brahim Oukacha, and Badran Raddaoui. On an mcs-based inconsistency measure. *International Journal of Approximate Reasoning*, 2016.

[6] Krzysztof R. Apt, Howard A. Blair, and Adrian Walker. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann, 1988.

[7] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, and Daniele Nardi. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[8] James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *Practical Aspects of Declarative Languages, 7th International Symposium, PADL 2005, Long Beach, CA, USA, January 10-11, 2005, Proceedings*, pages 174–186, 2005. doi: 10.1007/978-3-540-30557-6_14. URL https://doi.org/10.1007/978-3-540-30557-6_14.

[9] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[10] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract argumentation frameworks and their semantics. In Pietro Baroni, Dov Gabbay, Massimiliano

Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 4. College Publications, February 2018.

[11] Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. Locality and modularity in abstract argumentation. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 19. College Publications, February 2018.

[12] Ringo Baumann. Splitting an argumentation framework. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 40–53. Springer, 2011.

[13] Ringo Baumann. What does it take to enforce an argument? minimal change in abstract argumentation. In *20th European Conference on Artificial Intelligence.*, pages 127–132, 2012. doi: 10.3233/978-1-61499-098-7-127. URL `https://doi.org/10.3233/978-1-61499-098-7-127`.

[14] Ringo Baumann. *Metalogical Contributions to the Nonmonotonic Theory of Abstract Argumentation*. College Publications - Studies in Logic, 2014.

[15] Ringo Baumann. On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 14. College Publications, February 2018. also appears in IfCoLog Journal of Logics and their Applications 4(8):2779-2886.

[16] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Proceedings of COMMA 2010*, pages 75–86, 2010. doi: 10.3233/978-1-60750-619-5-75. URL `http://dx.doi.org/10.3233/978-1-60750-619-5-75`.

[17] Ringo Baumann and Gerhard Brewka. Spectra in abstract argumentation: An analysis of minimal change. In *LPNMR, Proceedings of 12th International Conference in Logic Programming and Nonmonotonic Reasoning,*, pages 174–186, 2013. doi: 10.1007/978-3-642-40564-8_18. URL `https://doi.org/10.1007/978-3-642-40564-8\_18`.

[18] Ringo Baumann and Gerhard Brewka. AGM meets abstract argumentation: Expansion and revision for Dung frameworks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2734–2740. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL `http://dl.acm.org/citation.cfm?id=2832581.2832631`.

[19] Ringo Baumann and Gerhard Brewka. Extension removal in abstract argumentation – an axiomatic approach. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019. in press.

[20] Ringo Baumann and Christof Spanring. Infinite argumentation frameworks - On the existence and uniqueness of extensions. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, pages 281–295, 2015. doi: 10.1007/978-3-319-14726-0_19.

[21] Ringo Baumann and Christof Spanring. A study of unrestricted abstract argumentation frameworks. In *IJCAI, Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 807–813, 2017.

[22] Ringo Baumann and Hannes Strass. An abstract logical approach to characterizing strong equivalence in logic-based knowledge representation formalisms. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2016.

[23] Ringo Baumann and Markus Ulbricht. If nothing is accepted - repairing argumentation frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, pages 108–117, 2018. URL `https://aaai.org/ocs/index.php/KR/KR18/paper/view/17979`.

[24] Ringo Baumann, Wolfgang Dvorák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran. Compact argumentation frameworks. In *21st European Conference on Artificial Intelligence*, pages 69–74, 2014. doi: 10.3233/978-1-61499-419-0-69. URL `https://doi.org/10.3233/978-1-61499-419-0-69`.

[25] Salem Benferhat, Didier Dubois, and Henri Prade. A local approach to reasoning under inconsisteny in stratified knowledge bases. In *Proceedings of the Third European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, 1995.

[26] Claude Berge. Hypergraphs: Combinatorics of finite sets, vol. 45, 1989.

[27] Philippe Besnard. Revisiting postulates for inconsistency measures. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, pages 383–396, 2014.

[28] Philippe Besnard. Forgetting-based inconsistency measure. In *Proceedings of the 10th International Conference on Scalable Uncertainty Management (SUM'16)*, pages 331–337, 2016.

[29] Jean-Yves Béziau, Walter Carnielli, and Dov Gabbay, editors. *Handbook of Paraconsistency*. College Publications, London, 2007.

[30] Elazar Birnbaum and Eliezer L Lozinskii. Consistent subsets of inconsistent systems: structure and behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(1):25–46, 2003.

[31] Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in argumentation systems: Exploring the interest of removing an argument. In *Scalable Uncertainty Management - SUM*, pages 275–288, 2011. doi: 10.1007/978-3-642-23963-2_22. URL `https://doi.org/10.1007/978-3-642-23963-2_22`.

[32] Glauber De Bona, John Grant, Anthony Hunter, and Sbastien Konieczny. Towards a unified framework for syntactic inconsistency measures. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[33] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 2004.

[34] Gerhard Brewka. *Nonmonotonic reasoning: logical foundations of commonsense*, volume 12. Cambridge University Press, 1991.

[35] Gerhard Brewka and Thomas Eiter. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 385–390, 2007.

[36] Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski. Answer set programming at a glance. *Commun. ACM*, 54(12):92–103, 2011. doi: 10.1145/2043174.2043195. URL `http://doi.acm.org/10.1145/2043174.2043195`.

[37] Gerhard Brewka, Matthias Thimm, and Markus Ulbricht. Strong inconsistency in nonmonotonic reasoning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 901–907, 2017. doi: 10.24963/ijcai.2017/125. URL `https://doi.org/10.24963/ijcai.2017/125`.

[38] Gerhard Brewka, Matthias Thimm, and Markus Ulbricht. Strong inconsistency. *Artificial Intelligence*, 267:78–117, 2019.

[39] Hans Kleine Büning and Xishun Zhao. Minimal false quantified boolean formulas. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 339–352. Springer, 2006.

[40] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, 38:49–84, 2010. doi: 10.1613/jair.2965. URL `https://doi.org/10.1613/jair.2965`.

[41] Laurence Cholvy and Anthony Hunter. Information fusion in logic: A brief overview. In *Qualitative and Quantitative Practical Reasoning (ECSQARU'97/FAPR'97)*, volume 1244 of *Lecture Notes in Computer Science*, pages 86–95. Springer, 1997.

[42] Jean-Francois Condotta, Badran Raddaoui, and Yakoub Salhi. Quantifying conflicts for spatial and temporal information. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, 2016.

[43] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 317–328. Springer, 2005.

[44] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. On the revision of argumentation systems: Minimal change of arguments statuses. In *Proceedings of Principles of Knowledge Representation and Reasoning, KR*, 2014. URL `http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7967`.

[45] Glauber De Bona and Marcelo Finger. Measuring inconsistency in probabilistic logic: Rationality postulates and dutch book interpretation. *Artificial Intelligence*, 227:140–164, 2015.

[46] Glauber De Bona and Anthony Hunter. Localising iceberg inconsistencies. *Artificial Intelligence*, 246:118–151, 2017.

[47] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation. *Int. J. Approx. Reasoning*, 93:395–423, 2018. doi: 10.1016/j.ijar.2017.11.013. URL https://doi.org/10.1016/j.ijar.2017.11.013.

[48] Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency in possibilistic knowledge bases: To live with it or not live with it. In L.A. Zadeh and J. Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty* , pages 335–351. Wiley, New York, 1992.

[49] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995. doi: 10.1016/0004-3702(94)00041-X. URL http://dx.doi.org/10.1016/0004-3702(94)00041-X.

[50] Arnaud Durand, Miki Hermann, and Phokion G Kolaitis. Subtractive reductions and complete problems for counting complexity classes. *Theoretical Computer Science*, 340(3):496–513, 2005.

[51] Wolfgang Dvorák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*. College Publications, February 2018. also appears in IfCoLog Journal of Logics and their Applications 4(8):2623–2706.

[52] Thomas Eiter and Georg Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15(3-4):289–323, 1995.

[53] Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. The diagnosis frontend of the dlv system. *AI Communications*, 12(1-2):99–111, 1999.

[54] Thomas Eiter, Michael Fink, Hans Tompits, and Stefan Woltran. Strong and uniform equivalence in answer-set programming: Characterizations and complexity results for the non-ground case. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 695–700, 2005. URL http://www.aaai.org/Library/AAAI/2005/aaai05-109.php.

[55] Thomas Eiter, Michael Fink, Peter Schüller, and Antonius Weinzierl. Finding explanations of inconsistency in multi-context systems. *Artif. Intell.*, 216:233–274, 2014. doi: 10.1016/j.artint.2014.07.008. URL https://doi.org/10.1016/j.artint.2014.07.008.

[56] Laurent Garcia, Claire Lefèvre, Odile Papini, Igor Stéphan, and Eric Würbel. A semantic characterization for asp base revision. In *International Conference on Scalable Uncertainty Management*, pages 334–347. Springer, 2017.

[57] Michael Gelfond and Nicola Leone. Logic programming and knowledge representation – the A-Prolog perspective. *Artificial Intelligence*, 138(1–2):3–38, 2002.

[58] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991. doi: 10.1007/BF03037169. URL `http://dx.doi.org/10.1007/BF03037169`.

[59] John Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.

[60] John Grant and Anthony Hunter. Measuring Inconsistency in Knowledgebases. *Journal of Intelligent Information Systems*, 27:159–184, 2006.

[61] John Grant and Anthony Hunter. Measuring consistency gain and information loss in stepwise inconsistency resolution. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 362–373. Springer, 2011.

[62] John Grant and Anthony Hunter. Analysing inconsistent information using distance-based measures. *International Journal of Approximate Reasoning*, In press, 2016.

[63] Sven Ove Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[64] Lane A Hemaspaandra and Heribert Vollmer. The satanic notations: counting classes beyond# p and other definitional adventures. *ACM SIGACT News*, 26(1):2–13, 1995.

[65] Anthony Hunter. How to act on inconsistent news: Ignore, resolve, or reject. *Data & Knowledge Engineering*, 57(3):221–239, 2006.

[66] Anthony Hunter and Sébastien Konieczny. Approaches to Measuring Inconsistent Information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 189–234. Springer International Publishing, 2004.

[67] Anthony Hunter and Sébastien Konieczny. Measuring inconsistency through minimal inconsistent sets. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008)*, pages 358–366. AAAI Press, 2008.

[68] Anthony Hunter and Sebastien Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artificial Intelligence*, 174(14):1007–1026, July 2010.

[69] Anthony Hunter, Simon Parsons, and Michael Wooldridge. Measuring inconsistency in multi-agent systems. *KI-Künstliche Intelligenz*, 28(3):169–178, 2014.

[70] Katsumi Inoue and Chiaki Sakama. Abductive framework for nonmonotonic theory change. In *IJCAI*, volume 95, pages 204–210. Citeseer, 1995.

[71] Said Jabbour and Lakhdar Sais. Exploiting MUS Structure to Measure Inconsistency of Knowledge Bases. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16)*, pages 991–998, 2016.

[72] Said Jabbour, Yue Ma, Badran Raddaoui, and Lakhdar Sais. On the characterization of inconsistency: A prime implicates based framework. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 146–153. IEEE, 2014.

[73] Said Jabbour, Yue Ma, Badran Raddaoui, Lakhdar Sais, and Yakoub Salhi. A MIS Partition Based Framework for Measuring Inconsistency. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 84–93, 2016.

[74] Eun Jung Kim, Sebastian Ordyniak, and Stefan Szeider. The complexity of repairing, adjusting, and aggregating of extensions in abstract argumentation. In *Theory and Applications of Formal Argumentation - Second International Workshop, TAFA*, pages 158–175, 2013. doi: 10.1007/978-3-642-54373-9_11. URL https://doi.org/10.1007/978-3-642-54373-9_11.

[75] Sébastien Konieczny and Ramon Pino Perez. Logic based merging. *Journal of Philosophical Logic*, 40:239–270, 2011.

[76] Sébastien Konieczny, Jérôme Lang, and Pierre Marquis. Reasoning under inconsistency: the forgotten connective. In *Proceedings of IJCAI-2005*, pages 484–489, 2005.

[77] Jerome Lang and Pierre Marquis. Reasoning under inconsistency: A forgetting-based approach. *Artificial Intelligence*, 174(12–13):799–823, July 2010.

[78] Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008. doi: 10.1007/s10817-007-9084-z. URL https://doi.org/10.1007/s10817-007-9084-z.

[79] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016. doi: 10.1007/s10601-015-9183-0. URL https://doi.org/10.1007/s10601-015-9183-0.

[80] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, pages 23–37, 1994.

[81] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Trans. Comput. Log.*, 2(4):526–541, 2001. doi: 10.1145/502166.502170. URL http://doi.acm.org/10.1145/502166.502170.

[82] Nicolas Madrid and Manuel Ojeda-Aciego. Measuring instability in normal residuated logic programs: adding information. In *Proceedings of the 19th IEEE International Conference on Fuzzy Systems*, 2010.

[83] Joao Marques-Silva, Mikolas Janota, Alexey Ignatiev, and Antonio Morgado. Efficient model based diagnosis with maximum satisfiability. In *IJCAI*, volume 15, pages 1966–1972, 2015.

[84] Maria Vanina Martinez, Andrea Pugliese, Gerardo I Simari, VS Subrahmanian, and Henri Prade. How dirty is your relational database? an axiomatic approach. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 103–114. Springer, 2007.

[85] Amit Metodi, Roni Stern, Meir Kalech, and Michael Codish. A novel sat-based approach to model based diagnosis. *Journal of Artificial Intelligence Research*, 51: 377–411, 2014.

[86] Kedian Mu. Responsibility for inconsistency. *International Journal of Approximate Reasoning*, 61:43–60, 2015.

[87] Kedian Mu, Weiru Liu, and Zhi Jin. A general framework for measuring inconsistency through minimal inconsistent sets. *Knowledge and Information Systems*, 27 (1):85–114, 2011.

[88] Farid Nouioua and Eric Würbel. Removed set-based revision of abstract argumentation frameworks. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014*, pages 784–791, 2014. doi: 10.1109/ICTAI.2014.121. URL `https://doi.org/10.1109/ICTAI.2014.121`.

[89] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.*, 175(14-15):1985–2009, 2011. doi: 10.1016/j.artint.2011.06.003. URL `http://dx.doi.org/10.1016/j.artint.2011.06.003`.

[90] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[91] Christos H Papadimitriou and David Wolfe. The complexity of facets resolved. *Journal of Computer and System Sciences*, 37(1):2–13, 1988.

[92] Nico Potyka. Linear Programs for Measuring Inconsistency in Probabilistic Logics. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, pages 568–577, 2014.

[93] Iyad Rahwan and Guillermo R Simari. *Argumentation in artificial intelligence*, volume 47. Springer, 2009.

[94] Wolfgang Rautenberg. *Einführung in die mathematische Logik*. Springer, 2002.

[95] Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980. doi: 10.1016/0004-3702(80)90014-4. URL `http://dx.doi.org/10.1016/0004-3702(80)90014-4`.

[96] Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987. doi: 10.1016/0004-3702(87)90062-2. URL `http://dx.doi.org/10.1016/0004-3702(87)90062-2`.

[97] Chiaki Sakama and Katsumi Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming*, 3(6):671–715, 2003.

[98] Babak Salimi and Leopoldo Bertossi. From causes for database queries to repairs and model-based diagnosis and back. *arXiv preprint arXiv:1412.4311*, 2014.

[99] Claudia Schulz, Ken Satoh, and Francesca Toni. Characterising and explaining inconsistency in logic programs. In Francesco Calimeri, Giovambattista Ianni, and Miroslaw Truszczynski, editors, *Logic Programming and Nonmonotonic Reasoning: 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, pages 467–479. Springer International Publishing, Cham, 2015.

[100] Roni Tzvi Stern, Meir Kalech, Alexander Feldman, and Gregory M. Provan. Exploring the duality in conflict-directed model-based diagnosis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5182`.

[101] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013. doi: 10.1016/j.artint.2013.09.004. URL `https://doi.org/10.1016/j.artint.2013.09.004`.

[102] Matthias Thimm. Measuring inconsistency in probabilistic knowledge bases. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 530–537. AUAI Press, 2009.

[103] Matthias Thimm. Inconsistency Measures for Probabilistic Logics. *Artificial Intelligence*, 197:1–24, 2013.

[104] Matthias Thimm. Stream-based inconsistency measurement. *International Journal of Approximate Reasoning*, 68:68–87, 2016.

[105] Matthias Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234:120–151, 2016.

[106] Matthias Thimm. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *Künstliche Intelligenz*, August 2016.

[107] Matthias Thimm. On the evaluation of inconsistency measures. In John Grant and Maria Vanina Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[108] Matthias Thimm and Johannes Peter Wallner. Some complexity results on inconsistency measurement. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 114–124, April 2016.

[109] Seinosuke Toda and Osamu Watanabe. Polynomial-time 1-turing reductions from# ph to# p. *Theoretical Computer Science*, 100(1):205–221, 1992.

[110] Markus Ulbricht. Repairing non-monotonic knowledge bases. In *Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA 2019), to appear.*, 2019.

[111] Markus Ulbricht, Matthias Thimm, and Gerhard Brewka. Measuring Inconsistency in Answer Set Programs. In *Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA'16)*, pages 577–583, November 2016.

[112] Markus Ulbricht, Matthias Thimm, and Gerhard Brewka. Inconsistency measures for disjunctive logic programs under answer set semantics. In John Grant and Maria Vanina Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[113] Markus Ulbricht, Matthias Thimm, and Gerhard Brewka. Measuring strong inconsistency. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, February 2018.

[114] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[115] Klaus W Wagner. The complexity of combinatorial problems with succinct input representation. *Acta informatica*, 23(3):325–356, 1986.

[116] Emil Weydert. Semi-stable extensions for infinite frameworks. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, pages 336–343, 2011.

Appendix A

Technical Proofs of Chapter 6

Theorem 6.2.2. *If* $m \geq 2$*, then* QBF-MU$(Q_1, ..., Q_m)$ *is* D^p_m*-complete.*

Proof. Membership: Given a QBF

$$\Phi = Q_1 X_1 \dots Q_m X_m \, \phi$$

with $\phi = C_1 \wedge \dots \wedge C_r$ we can verify that it is a postive instance of our decision problem QBF-MU$(Q_1, ..., Q_m)$ by

- checking that Φ is false which is in Π^p_m if $Q_1 = \exists$ and in Σ^p_m if $Q_1 = \forall$ and
- checking that for each $k = 1, ..., r$ the formula

  $$Q_1 X_1 \dots Q_m X_m \, C_1 \wedge \dots \wedge C_{k-1} \wedge C_{k+1} \wedge \dots \wedge C_r$$

  is true which is in Σ^p_m if $Q_1 = \exists$ and in Π^p_m if $Q_1 = \forall$.

Since the latter consists of only linearly many checks, we obtain membership in D^p_m.

Hardness: We consider the generic D^p_m-complete problem which asks for two given QBFS $\Phi_i = Q_1 X_1 \dots Q_m X_m \, \phi_i$, $i = 1, 2$, whether Φ_1 is false while Φ_2 is true. First, we give a (polynomial) construction to obtain a formula Ψ which is a positive instance of the decision problem QBF-MU$(Q_1, ..., Q_m)$ if and only if Φ_1 is false. A minor adjustment will lead to a second formula Ψ' which is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only Φ_2 is true. Combining both constructions will yield D^p_m-hardness. Since both reductions are considered independently, we omit the indices 1 and 2 for ease of presentation and denote the formula by $\Phi = Q_1 X_1 \dots Q_m X_m \, \phi$ in both steps.

So let $\phi = C_1 \wedge \dots \wedge C_r$ be a conjunction and let

$$\Phi = Q_1 X_1 \dots Q_m X_m \, \phi$$

be a QBF in prenex normal form where $m \geq 2$. We distinguish two cases, depending on the *final* quantifier.

Case 1: $Q_m = \exists$.

Here, we can assume that ϕ is in 3-CNF, i. e., all C_k are disjunctions containing at most three literals, say $C_k = x_{k,1} \vee \dots \vee x_{k,3}$ for $k = 1, ..., r$. Our proof is similar to the one given in [91], Theorem 1. The reader may be referred to this proof since we do not give as many details. We use a similar construction, where we, roughly speaking, ignore variables that occur in the scope of an universal quantifier for the most part.

For any $k \in \{1, \ldots, r\}$, we assume w. l. o. g. that the disjunction C_k is not a tautology, i. e., does not contain both a and $\neg a$ for an atom a.

Let $\mathcal{X}$ be the set of all variables in Φ that occur in the scope of an existential quantifier. Let $y_1, \ldots, y_r$ be fresh atoms not appearing in Φ and define

$$Y = y_1 \vee \ldots \vee y_r.$$

Let

$$Y - y_k := y_1 \vee \ldots \vee y_{k-1} \vee y_{k+1} \vee \ldots \vee y_r$$

for each $k = 1, \ldots, r$. We construct a formula ψ containing the following conjuncts.

- If C_k occurs in ϕ, then
$$D_k = C_k \vee (Y - y_k)$$
is a conjunct of ψ.
- Let $C_k = x_{k,1} \vee \ldots \vee x_{k,3}$. Then, for $j = 1, 2, 3$ and $x_{k,j} \in \mathcal{X}$
$$E_{k,j} = \neg x_{k,j} \vee (Y - y_k) \vee \neg y_k$$
occurs in ψ as a conjunct.
- For each $i, j = 1, \ldots, r$ with $i \neq j$
$$H_{i,j} = \neg y_i \vee \neg y_j$$
is a conjunct of ψ.

Consider

$$\Psi = Q_1 X_1 \ldots Q_{m-1} X_{m-1} \exists X_m \cup \{y_1, \ldots, y_r\}\, \psi.$$

We claim that Φ is false if and only if Ψ is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$.

"$\Rightarrow$": We start by showing that Ψ is false. Afterwards, we prove minimality. The first step is to argue that all y-variables would need to be false in order for Ψ to be true. Due to the $H_{i,j}$ conjuncts, at most one y-variable can be true, say y_k. We can assume that at least one $E_{k,j}$ ($j = 1, 2, 3$) exists, i. e., C_k contains variables in $\mathcal{X}$. Otherwise, all variables in C_k occur in the scope of an universal quantifier. Since we assumed no tautological conjunct exists, there is an assignment to those variables rendering C_k false. Thus, in order for Ψ to be true, there has to be a $k' \neq k$ such that $y_{k'}$ is true (because of the conjunct D_k), which is not possible, as argued above.

Now consider $E_{k,j}$. To render it true, $x_{k,j}$ needs to be false ($j = 1, 2.3$). However, we also need to consider the conjunct

$$D_k = x_{k,1} \vee x_{x,2} \vee x_{k,3} \vee (Y - y_k).$$

If all three variables $x_{k,1}, \ldots, x_{k,3}$ are in $\mathcal{X}$, then D_k is false. Otherwise, the remaining variables occur in the scope of an universal quantifier. Again, there is a assignment to those variables rendering D_k false.

Hence, if one y-variable is true, then Ψ is false. This, however, finishes the first step already: Since Φ is false by assumption and all y-variables need to be false, Ψ is false as well.

We turn to minimality and argue that removal of any conjunct in ψ renders Ψ true.

- If we remove $H_{i,j}$, then letting y_i and y_j be true ensures satisfaction of all conjuncts. Of course, the other y-variables need to be false.

- If $E_{k,j}$ is removed (assuming it exists and hence $x_{k,j} \in \mathcal{X}$), then let $x_{k,j}$ and y_k be true, the other y-variables false. If $E_{k,j'}$ exists for $j' \neq j$, then $x_{k,j'}$ needs to be false. This assignment ensures satisfaction of all conjuncts.

- If we remove D_k, then all conjuncts are satisfied if y_k is true and $x_{k,1}, \ldots, x_{k,3}$ are false (for the j such that $x_{k,j} \in \mathcal{X}$). The latter are needed for the $E_{k,j}$ (which exists for the j such that $x_{k,j} \in \mathcal{X}$).

"$\Leftarrow$": Assume Ψ is false. In particular, this means that Ψ is false even if all y-variables are false. Thus, Φ is false.

So, Ψ is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if Φ is false.

As mentioned above, we need a second reduction to obtain a formula Ψ' that is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if Φ is true. To obtain such a formula, use the same construction and add $y_1 \vee \ldots \vee y_r$ as a conjunct to ψ. Then, Ψ' is definitely false and minimal if and only if Φ is true, as similar considerations show.

Hence, we obtain the two desired formulas Ψ and Ψ'. We describe below how they are combined, which does not depend on the cases we distinguish.

<u>Case 2</u>: $Q_m = \forall$.

We may assume ϕ to be in 3-DNF here, i. e., $r = 1$ and $C := C_1$ is in 3-DNF. Assume $C = D_1 \vee \ldots \vee D_n$ with $D_i = x_{i,1} \wedge x_{i,2} \wedge x_{i,3}$ for $i = 1, \ldots, n$. We see that $\Psi = \Phi$ is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if the formula Φ is false, since removing the only conjunct (i. e., the whole formula ϕ) renders Φ true, because the empty conjunct is true by definition. Moreover,

$$\Psi' = Q_1 X_1 \ldots Q_{m-2} X_{m-2} \exists X_{m-1} \cup \{y\} \forall X_m \, \psi'$$

with

$$\psi' = ((D_1 \wedge y) \vee \ldots \vee (D_n \wedge y)) \wedge (\neg y)$$

is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if Φ is true.

<u>Combining</u>:

In both cases above, we obtain two formulas Ψ and Ψ' that are both positive instances of QBF-MU$(Q_1, ..., Q_m)$. To finish our proof for hardness, we turn them into *one* formula. We assume w. l. o. g. that Ψ and Ψ' are QBFs over disjoint sets of variables. Given these two QBFs, we construct one formula Θ as it is done in [91], Lemma 3. Let

$$\Psi = Q_1 X_1 \ldots Q_m X_m \, \psi, \qquad \Psi' = Q_1 X_1' \ldots Q_m X_m' \, \psi'$$

with

$$\psi = C_1 \wedge \ldots \wedge C_r, \qquad \psi' = C_1' \wedge \ldots \wedge C_{r'}'.$$

Now, Θ contains of all possible pairs of clauses, one from ψ and one from ψ', i. e., for

$$\theta_{p,q} = C_p \wedge C_q', \quad p = 1, \ldots, r, q = 1, \ldots, r'$$

we let

$$\Theta = Q_1 X_1 \cup X_1' \dots Q_m X_m \cup X_m' \bigwedge_{p=1,\dots,r,q=1,\dots,r'} \theta_{p,q}$$

Now, removing $\theta_{p,q}$ from Θ corresponds to removing C_p from Ψ and C_q' from Ψ'. Hence, Θ is a "yes" instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if both Ψ and Ψ' are.

To summarize, Θ is a positive instance of QBF-MU$(Q_1, ..., Q_m)$ if and only if Φ_1 is false while Φ_2 is true. Hardness in D_m^p follows. □

Proposition 6.2.3. QBF-MU$(\forall)$ *is NP-complete.*

Proof. Let Φ be the sentence

$$\Phi = \forall X \phi$$

with $\phi = C_1 \wedge \dots \wedge C_r$. Verifying that Φ is false is obviously NP-complete in general. However, there is nothing to check beyond that, because removal of an arbitrary conjunct renders Φ true if and only if $r = 1$, i. e., if ϕ consists of one conjunct only. That is easy to see: Assume $r \geq 2$ and assume Φ is false. Then, there is a k and a truth assignment to the variables such that C_k is false. If we remove k' for any $k' \neq k$, then C_k is still false for the same truth assignment and thus,

$$\forall X \ (C_1 \wedge \dots \wedge C_{k'-1} \wedge C_{k'+1} \wedge \dots \wedge C_r)$$

is still false, i. e., Φ is a "no" instance of QBF-MU$(\forall)$. □

Theorem 6.2.6. *For $m \geq 1$, there is a logic $L_{\Pi_m^p} = (\mathcal{WF}_{\Pi_m^p}, \mathcal{BS}_{\Pi_m^p}, \mathcal{INC}_{\Pi_m^p}, \mathcal{ACC}_{\Pi_m^p})$ such that* $\text{SAT}_{L_{\Pi_m^p}}$ *is in Π_m^p and*

(a) $\text{S-INC}_{L_{\Pi_m^p}}$ *is Π_{m+1}^p-complete and*

(b) $\text{MIN-S-INC}_{L_{\Pi_m^p}}$ *is D_{m+1}^p-complete.*

Proof. For this proof, we make use of the following notation: Assume we are given a set X of atoms and a formula ϕ over X. Consider a (partial) assignment $\omega : X \to \{0, 1\}$. Now, we let $\phi[\omega]$ be the formula where the atoms are evaluated according to ω, i. e., $a \in X$ is substituted by 1 if $\omega(a) = 1$, by 0 if $\omega(a) = 0$ and remains unchanged if ω is not defined on a.

Now let $m \geq 1$. Consider quantifiers $Q_1 = \forall, \dots, Q_m$ and sets of variables $X_1, \dots, X_m$. We define a logic

$$L_{\Pi_m^p} = L_{\Pi_m^p}(Q_1, \dots, Q_m, X_1, \dots, X_m) = \left(\mathcal{WF}_{\Pi_m^p}, \mathcal{BS}_{\Pi_m^p}, \mathcal{INC}_{\Pi_m^p}, \mathcal{ACC}_{\Pi_m^p}\right).$$

The set $\mathcal{WF}_{\Pi_m^p}$ consists of tuples of the form (ϕ, L), where ϕ is a boolean formula over the variables in $X_1 \cup \dots \cup X_m$ and L is a set of literals over $X_1 \cup \dots \cup X_m$. Let $\mathcal{BS}_{\Pi_m^p} = \{\bot, \top\}$ and $\mathcal{INC}_{\Pi_m^p} = \{\bot\}$. In a nutshell, the set L in (ϕ, L) defines a partial assignment. So given a knowledge base

$$\mathcal{K} = \{(C_1, L_1), \dots, (C_r, L_r)\}$$

we let $L_{\mathcal{K}} = L = L_1 \cup \dots \cup L_r$. If L contains two complementary literals, then we let $\mathcal{ACC}_{\Pi_m^p}(\mathcal{K}) = \{\bot\}$, rendering $\mathcal{K}$ inconsistent (the reason is that L shall correspond to a (partial) assignment to the variables; hence, it cannot contain a complementary pair of literals). Otherwise, let $\omega_{\mathcal{K}} = \omega : X_1 \cup \ldots \cup X_m \to \{0,1\}$ be the (partial) assignment corresponding to L, i. e.,

$$\omega(l) = \begin{cases} 1 & \text{if } l \in L, \\ 0 & \text{if } l \notin L. \end{cases}$$

The mapping $\mathcal{ACC}_{\Pi_m^p}$ treats the formulas $C_1, \ldots, C_r$ in $\mathcal{K}$ as conjuncts while respecting the (partial) assignment ω, i. e., we let $\phi_{\mathcal{K}} = \phi = C_1 \wedge \ldots \wedge C_r$ and

$$\mathcal{ACC}_{\Pi_m^p}(\mathcal{K}) = \begin{cases} \{\top\} & \text{if } \forall X_1 \ldots Q_m X_m\, \phi[\omega], \\ \{\bot\} & \text{otherwise.} \end{cases}$$

This mechanism introduced by ω is important for us due to two different reasons: First, we can introduce formulas that correspond to partial assignments to "neutralize" a universal quantifier, which gives our logic the non-monotonic layer we need. Second, we can construct knowledge bases such that supersets correspond to assignments. As strong inconsistency considers *all* supersets of a given $H \subseteq \mathcal{K}$, this facilitates simulation of an additional universal quantifier.

We now show that $L_{\Pi_m^p}$ has the properties we claim, i. e., $\text{SAT}_{L_{\Pi_m^p}}$ is in Π_m^p, S-INC$_{L_{\Pi_m^p}}$ is Π_{m+1}^p-complete and MIN-S-INC$_{L_{\Pi_m^p}}$ is D_{m+1}^p-complete.

Membership: Since we consider m quantifiers, where the first one is universal, it is rather easy to see that $\text{SAT}_{L_{\Pi_m^p}}$ is in Π_m^p. Then S-INC$_{L_{\Pi_m^p}} \in \Pi_{m+1}^p$ and MIN-S-INC$_{L_{\Pi_m^p}} \in \mathsf{D}_{m+1}^p$ follow from Theorems 6.2.4 and 6.2.5.

Hardness: Assume we are given a formula

$$\Phi = \exists Z \forall X_2 Q_3 X_3 \ldots Q_{m+1} X_{m+1}\, \phi. \tag{A.1}$$

Note that Φ contains $m+1$ quantifiers, i. e., deciding whether it is true is Σ_{m+1}^p-complete in general. We assume ϕ is an arbitrary conjunction, i. e., $\phi = C_1 \wedge \ldots \wedge C_r$. We consider the logic

$$L_{\Pi_m^p} = L_{\Pi_m^p}(\forall, Q_3, \ldots, Q_{m+1}, Z \cup X_2, X_3, \ldots, X_{m+1}).$$

For (a), we show that there is a knowledge base $\mathcal{K}$ with a subset $\mathcal{H}$ such that $(\mathcal{K}, \mathcal{H})$ is a "yes" instance of S-INC$_{L_{\Pi_m^p}}$ if and only if Φ is false. The statement (b) follows from (a) utilizing a formula similar to Θ we constructed in the proof of Theorem 6.2.2.

(a): We prove the following statement: Given the formula

$$\Phi = \exists Z \forall X_2 Q_3 X_3 \ldots Q_{m+1} X_{m+1}\, \phi \tag{A.1}$$

where $\phi = C_1 \wedge \ldots \wedge C_r$ is an arbitrary conjunction, the pair $(\mathcal{K}, \mathcal{H})$ with

$$\mathcal{K} = \{(C_1, \emptyset), \ldots, (C_r, \emptyset), (\top, z_1), (\top, \neg z_1), \ldots, (\top, z_n), (\top, \neg z_n)\},$$
$$\mathcal{H} = \{(C_1, \emptyset), \ldots, (C_r, \emptyset)\}$$

is a "yes" instance of S-INC$_{L_{\Pi_m^p}}$ if and only if Φ is false.

Note that we can identify $\mathcal{H}$ with $\{\phi, \emptyset\}$ due to the definition of our logic. The subset $\mathcal{H}$ is considered consistent if

$$\forall Z \cup X_2 Q_3 X_3 \ldots Q_{m+1} X_{m+1}\, \phi$$

is valid and hence, inconsistent if

$$\exists\{Z \cup X_2\}\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi \tag{A.2}$$

holds, where $\overline{Q_i}$ is the complementary quantifier. Note that the formulas besides the $(C_i, \emptyset)$ are all of the form $(\top, z_k)$ resp. $(\top, \neg z_k)$. Hence, all they do is fixing z-variables. So, naturally, any set $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ corresponds to the formula Φ with respect to a partial assignment on Z. This motivates the following notation: Given a (partial) assignment $\omega : Z \to \{0, 1\}$, we let $\mathcal{H}_\omega$ be the set of formulas of the form $(\top, z_k)$ resp. $(\top, \neg z_k)$ that naturally corresponds to the assignment, i. e., if $\omega(z_k) = 1$, then $(\top, z_k)$ occurs in $\mathcal{H}_\omega$ and if $\omega(z_k) = 0$, then $(\top, \neg z_k)$ occurs in $\mathcal{H}_\omega$.

Now, we show: Φ is false if and only if $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent.

"$\Leftarrow$": Assume $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent. Consider an assignment $\omega : Z \to \{0, 1\}$ and the set

$$\mathcal{H}' = \mathcal{H} \cup \mathcal{H}_\omega.$$

Since $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent, $\mathcal{H}'$ is inconsistent. The formulas $\mathcal{H}_\omega$ augment ϕ with conjuncts "$\top$" only, which can be ignored. Additionally, the z-variables are fixed according to ω, making the consideration of $\phi[\omega]$ rather than ϕ itself necessary. Thus, $\mathcal{H}'$ being inconsistent means

$$\exists\{Z \cup X_2\}\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega]$$

holds. Since ω fixes the z-variables, this is equivalent to

$$\exists X_2\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega].$$

Since ω was an arbitrary assignment, we obtain that

$$\forall Z\exists X_2\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi$$

holds. Hence,

$$\Phi = \exists Z\forall X_2 Q_3 X_3 \ldots Q_{m+1}X_{m+1}\, \phi \tag{A.1}$$

is false.

"$\Rightarrow$": Now assume Φ is false. Hence,

$$\forall Z\exists X_2\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi \tag{A.3}$$

holds. Again, consider an assignment $\omega : Z \to \{0, 1\}$. Since (A.3) is true, we obtain that in particular,

$$\exists X_2\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega]$$

is true. Since the z-variables are fixed anyway,

$$\exists\{Z \cup X_2\}\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega]$$

holds as well. Due to the definition of our logic, this means that the set

$$\mathcal{H}' = \mathcal{H} \cup \mathcal{H}_\omega$$

is inconsistent. Now consider any set $\mathcal{H}^*$ with $\mathcal{H} \subseteq \mathcal{H}^* \subseteq \mathcal{H}'$. Such $\mathcal{H}^*$ naturally corresponds to an assignment $\omega^* : Z^* \to \{0, 1\}$ with $Z^* \subseteq Z$ and $\omega_{|Z^*} = \omega^*$. It is an easy observation that

$$\exists\{Z \cup X_2\}\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega]$$

being true now implies that

$$\exists\{Z \cup X_2\}\overline{Q_3}X_3 \ldots \overline{Q_{m+1}}X_{m+1}\, \neg\phi[\omega^*]$$

holds as well, because the latter is the same formula with less z-variables being fixed. Again by definition of our logic, $\mathcal{H}^*$ is inconsistent. Since ω and ω^* are arbitrary, we obtain that any

$$\mathcal{H}' = \mathcal{H} \cup \mathcal{H}_\omega$$

where ω is a (partial) assignment is inconsistent. The remaining sets $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ do not correspond to a (partial) assignment. Hence, all of them contain at least one pair of the form $\{(\top, z_k), (\top, \neg z_k)\}$ rendering them inconsistent (recall the definition of $\mathcal{ACC}_{\Pi_m^p}$ if the set L contains a complementary pair of literals). Hence, $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent.

This completes (a).

(b): For (b), the technical work is already done. Assume we are given the generic D^p_{m+1}-complete problem, i. e., two formulas Φ_1 and Φ_2 of the form

$$\Phi_i = \exists Z \forall X_2 Q_3 X_3 \ldots Q_{m+1} X_{m+1}\, \phi_i \tag{A.1}$$

and have to decide whether Φ_1 is false while Φ_2 is true. We construct the formula

$$\Theta = \exists Z \forall X_2 Q_3 X_3 \ldots Q_{m+1} X_{m+1}\, \theta$$

as it is done in the proof of Theorem 6.2.2, in the last step. Thus, θ is a conjunction and for notational convenience, we assume $\theta = C_1 \wedge \ldots \wedge C_r$. Recall that Θ is a "yes" instance of QBF-MU$(Q_1, ..., Q_{m+1})$ if and only if Φ_1 is false while Φ_2 is true. Consider $(\mathcal{K}, \mathcal{H})$ given as in (a), i. e.,

$$\begin{aligned}\mathcal{K} &= \{(C_1, \emptyset), \ldots, (C_r, \emptyset), (\top, z_1), (\top, \neg z_1), \ldots, (\top, z_n), (\top, \neg z_n)\},\\ \mathcal{H} &= \{(C_1, \emptyset), \ldots, (C_r, \emptyset)\}.\end{aligned}$$

Now $(\mathcal{K}, \mathcal{H})$ is a "yes" instance of MIN-S-INC$_{L_{\Pi_m^p}}$ iff $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent, but no proper subset of $\mathcal{H}$. Due to (a), this is the case if and only if Θ is false, but removing any conjunct from θ renders it true. By construction of Θ, this is the case if and only if Φ_1 is false, while Φ_2 is true. This yields hardness in D^p_{m+1}. □

Theorem 6.2.9. *There is a logic* $L_{P_{Strat}} = (\mathcal{WF}_{P_{Strat}}, \mathcal{BS}_{P_{Strat}}, \mathcal{INC}_{P_{Strat}}, \mathcal{ACC}_{P_{Strat}})$ *such that* SAT$_{L_{P_{Strat}}}$ *is in* $\mathsf{P} = \Sigma_0^p$ *and*

- *(a)* S-INC$_{L_{P_{Strat}}}$ *is* **coNP**-*complete and*
- *(b)* MIN-S-INC$_{L_{P_{Strat}}}$ *is* D_1^p-*complete.*

Proof. Our logic

$$L_{P_{Strat}} = (\mathcal{WF}_{P_{Strat}}, \mathcal{BS}_{P_{Strat}}, \mathcal{INC}_{P_{Strat}}, \mathcal{ACC}_{P_{Strat}})$$

is similar to

$$L_{\text{ASP}^*} = (\mathcal{WF}_{\text{ASP}^*}, \mathcal{BS}_{\text{ASP}}, \mathcal{INC}_{\text{ASP}}, \mathcal{ACC}_{\text{ASP}})$$

with some minor modifications: in addition to rules r, $\mathcal{WF}_{P_{Strat}}$ contains tuples $(l, k(l))$ where l is a literal and $k(l)$ a non-negative integer. The integer $k(l)$ is the stratum corresponding to l. Hence, formally a knowledge base $\mathcal{K}$ is of the form $\mathcal{K} = P \cup S$ where P is a set of rules of the form

$$r: \quad l_0 \leftarrow l_1, \dots, l_m, \text{not } l_{m+1}, \dots, \text{not } l_n.$$

and S (S for *strata*) is a set of tuples of the form $(l, k(l))$ with $l \in A$. Now given a knowledge base $\mathcal{K} = P \cup S$ we check in polynomial time that for every literal l occurring in P, exactly one tuple $(l, k(l))$ is contained in S. Then, for every rule (except for the constraints occurring in P)

$$r: \quad l_0 \leftarrow l_1, \dots, l_m, \text{not } l_{m+1}, \dots, \text{not } l_n.$$

we check in polynomial time that

- $\|l_0\| \geq \|l_i\|$ for $i = 1, ..., m$,
- $\|l_0\| > \|l_j\|$ for $j = m+1, ..., n$,

i. e., that the program is stratified and the integers $k(l)$ correspond to appropriate strata for the atoms. If there is a literal l occurring in the program such that $(l, k(l))$ is not contained or occurring twice for two different integers $k_1(l)$ and $k_2(l)$ in S, then $\mathcal{K}$ is considered inconsistent. Otherwise, $\mathcal{K}$ is consistent if and only if the (stratified) program P has an answer set, i. e., all constraints are satisfied by the unique answer set of P. This check is done in polynomial time as well [6] and hence $\text{SAT}_{L_{P_{Strat}}}$ is in $\mathsf{P} = \Sigma_0^p$.

Now we utilize the construction given in the proof of Theorem 6.2.7 to see the hardness results. □

Theorem 6.2.10. *For disjunctive logic programs,*

(a) the problem S-INC$_{L_{ASP}}$ *is* Π_2^p*-complete,*

(b) the problem MIN-S-INC$_{L_{ASP}}$ *is* D_2^p*-complete.*

Proof. Membership: The membership statements follow from Theorems 6.2.4 and 6.2.5 since $\text{SAT}_{L_{\text{ASP}}}$ is in Σ_2^p.

Hardness: In [52], it has been shown that deciding whether a disjunctive logic program is consistent is Σ_2^p-complete. We sketch the proof since we are going to make use of the construction.

Assume we are given a QBF

$$\Phi = \exists X \forall Y \, \phi$$

where ϕ is in 3-DNF, i. e., $\phi = C_1 \vee \ldots \vee C_r$ with $C_k = l_{k,1} \wedge \ldots \wedge l_{k,3}$. Let $X = \{x, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$. We introduce fresh atoms $\{x'_1, \dots, x'_n\}$ and $\{y'_1, \dots, y'_m\}$ corresponding to the classical negation of the atoms. We let σ be the appropriate mapping, i. e.,

$$\sigma(l) = \begin{cases} l & \text{if } l \text{ is of the form } x_i \text{ or } y_j, \\ x'_i & \text{if } l \text{ is of the form } \neg x_i, \\ y'_j & \text{if } l \text{ is of the form } \neg y_j. \end{cases}$$

We consider the following program P.

$$
\begin{array}{ll}
P: & \\
x_i \vee x_i'. & i = 1, ..., n \\
y_j \vee y_j'. & j = 1, ..., m \\
y_j \leftarrow w. & j = 1, ..., m \\
y_j' \leftarrow w. & j = 1, ..., m \\
w \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}). & k = 1, ..., r \\
\leftarrow \text{not } w. &
\end{array}
$$

Now, in order for a set M to be an answer set of P, w needs to be contained in M. Since $w \in M$, all y-variables have to be in M as well. Thus, in order for M to be a minimal model, it needs to be possible to entail w for *any* choice of the y-variables (that is the translation of the universal quantifier). We refer the reader to [52] for more details.

Now, for (b), i. e., the hardness of MIN-S-INC$_{L_{\text{ASP}}}$ in D_2^p, we proceed as in the proof of Theorem 6.2.2. We assume we are given an instance of the generic D_2^p-complete problem, i. e., two formulas

$$\Phi_i = \exists X_i \forall Y_i \, \phi_i, \qquad i = 1, 2$$

where we have to decide whether it holds that Φ_1 is false while Φ_2 is true. We construct programs (P_1, H_1) that are a "yes" instance of MIN-S-INC$_{L_{\text{ASP}}}$ if and only if Φ_1 is false, programs (P_2, H_2) that are a "yes" instance of MIN-S-INC$_{L_{\text{ASP}}}$ if and only if Φ_2 is true and then we show how to combine them to one instance (Q, G) of MIN-S-INC$_{L_{\text{ASP}}}$ which is a positive one if and only if both (P_1, H_1) and (P_2, H_2) are. Utilizing the construction of (P_1, H_1) yields (a).

As in the proof of Theorem 6.2.2, we omit the indices and denote the formula by Φ for both constructions. We assume w. l. o. g. that both formulas are over disjoint sets of variables. So consider

$$\Phi = \exists X \forall Y \, \phi$$

as above. The reason why we have to adjust the construction given in [52] is the translation of the universal quantifier. The rules "$y_j \leftarrow w.$" and "$y_j' \leftarrow w.$" make sure that the formula is true for any choice of the y-variables. However, given H, one might construct H' with $H \subseteq H' \subseteq P$ such that H' does not contain all rules of this form. Then, the universal quantifier is not translated correctly.

To solve this issue, we allow entailment of w only if all y_j and y_j' are true. Thus, we introduce the atom w^* as a tool to obtain the y-variables. They, in turn, allow entailment of

w. So we construct P_1 and H_1 as follows.

P_1 :

$x_i \vee x_i'.$ $\quad i = 1, ..., n$

$y_j \vee y_j'.$ $\quad j = 1, ..., m$

$y_j \leftarrow w^*.$ $\quad j = 1, ..., m$

$y_j' \leftarrow w^*.$ $\quad j = 1, ..., m$

$w^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}).$ $\quad k = 1, ..., r$

$w \leftarrow y_1,\, y_1', \ldots, y_m,\, y_m'.$

$\leftarrow$ not w.

H_1 :

$\leftarrow$ not w.

We can already argue for hardness of S-INC$_{L_{\text{ASP}}}$ in Π_2^p.

(a): Deciding whether Φ is true is Σ_2^p-complete in general. Since P_1 is strongly inconsistent if and only if it is inconsistent itself, we see (as in [52]): Φ is false if and only if (P_1, P_1) is a "yes" instance of S-INC$_{L_{\text{ASP}}}$. Hence, the latter is Π_2^p-hard in general.

(b): We claim that Φ is false if and only if H_1 is minimal strongly P_1-inconsistent.

"$\Rightarrow$": Minimality is clear since H_1 consists of one rule only. Now assume H_1 is *not* strongly inconsistent. Then, there is a program $H_1 \subseteq H_1' \subseteq P_1$ that is consistent. We argue that in this case, P_1 itself is consistent. The program H_1' must contain "$w \leftarrow y_1,\, z_1, \ldots, y_m,\, z_m.$". One can see that any answer set M of H_1' has to contain $y_1, y_1', \ldots, y_m, y_m'$. Hence, the rules of the form "$y_j \leftarrow w.$" and "$y_j' \leftarrow w.$" can all assumed to be contained in H_1'. It is easy to see that the remaining rules do not introduce inconsistency. Hence, P_1 is consistent. Thus, Φ is true (cf. [52]), a contradiction.

"$\Leftarrow$": Assume H_1 is minimal strongly inconsistent. Then, P_1 itself is inconsistent. Hence, Φ is false.

As already mentioned, we construct a second program as well. As stipulated above, we denote the given QBF by Φ again. We make use of the same notations as above. However, recall that we assume the formulas to be over disjoint sets of variables. Also note that our goal is now to obtain programs (P_2, H_2) that are a positive instance of MIN-S-INC$_{L_{\text{ASP}}}$ if and only if Φ is true. Consider:

P_2 :

$x_i \vee x_i'.$ $\quad i = 1, ..., n$

$y_j \vee y_j'.$ $\quad j = 1, ..., m$

$y_j \leftarrow v^*.$ $\quad j = 1, ..., m$

$y_j' \leftarrow v^*.$ $\quad j = 1, ..., m$

$v^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}).$ $\quad k = 1, ..., r$

$v \leftarrow y_1,\, y_1', \ldots, y_m,\, y_m'.$

$\leftarrow$ not v.

$\leftarrow v$.

and

$$\begin{aligned} &H_2: \\ &\leftarrow \text{not } v. \\ &\leftarrow v. \end{aligned}$$

The situation is similar except that now, H_2 is strongly inconsistent in any case. Furthermore, the subprogram $\{\leftarrow v.\}$ is consistent. So, we have:

$$\begin{aligned} & H_2 \text{ is \textit{minimal} strongly inconsistent} \\ \Leftrightarrow\ & \{\leftarrow \text{not } v.\} \text{ is \textit{not} strongly inconsistent} \\ \Leftrightarrow\ & P_2 \setminus \{\leftarrow v.\} \text{ is consistent} \\ \Leftrightarrow\ & \Phi \text{ is true.} \end{aligned}$$

Combining:
Now assume we are given two formulas Φ_1 and Φ_2 as above over disjoint sets of variables. Construct P_1 and P_2 as above, except the occurring constraints. Let P be the union of both programs, i. e.,

$$P = (P_1 \cup P_2) \setminus \{\leftarrow \text{not } w., \leftarrow \text{not } v., \leftarrow v.\}$$

Set $Q = P \cup \{\leftarrow \text{not } v, \text{not } w., \leftarrow v, \text{not } w.\}$ and $G = \{\leftarrow \text{not } v, \text{not } w., \leftarrow v, \text{not } w.\}$. Using the considerations above, we can easily verify that G is minimal strongly Q-inconsistent if and only if Φ_1 (responsible for entailment of w) is false while Φ_2 (responsible for entailment of v) is true.

To summarize, (Q, G) is a "yes" instance of MIN-S-INC$_{L_{\text{ASP}}}$ if and only if Φ_1 is false while Φ_2 is true. We thus obtain the desired hardness result. □

Lemma 6.3.6. *Given an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$, there is a disjunctive logic program $P(\Phi) \subseteq \mathcal{WF}_{ASP}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\text{MOD}(\Phi)|.$$

Proof. This proof is similar to the one given in Lemma 6.3.3. Roughly speaking, the main difference is that we use the construction in [52, Theorem 3.1], in order to translate the two quantifiers in the formula.

So let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$ and $Z = \{z_1, \ldots, z_t\}$. Consider

$$\text{MOD}(\Phi) = \{X \mid \forall Y \exists Z\, \phi(X, Y, Z)\}.$$

We will construct a program $P = P(\Phi)$, where subsets correspond to assignments to the X-variables as in the proof of Lemma 6.3.3. The subsets that are *not* strongly P-inconsistent will correspond to assignments where

$$\exists Y \forall Z\, \neg\phi(X, Y, Z)$$

holds, and thus, strongly P-inconsistent subsets correspond to assignments in $\text{MOD}(\Phi)$.

We can assume ϕ to be a formula in 3-CNF and thus, $\neg\phi$ is in 3-DNF, i. e., the disjunction of $C_1, \ldots, C_r$ with $C_k = l_{k,1} \wedge \ldots \wedge l_{k,3}$. Let $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m, z'_1, \ldots, z'_t$ be fresh atoms and let σ be the mapping

$$\sigma(a) = \begin{cases} a & \text{if } a \in X \cup Y \cup Z, \\ a' & \text{if } a \in \{\neg x_1, \ldots, \neg x_n\} \cup \{\neg y_1, \ldots, \neg y_m\} \cup \{\neg z_1, \ldots, \neg z_t\}. \end{cases}$$

The construction in [52] works as follows: Given a formula

$$\Psi = \exists Y \forall Z\, \psi(Y, Z)$$

in 3-DNF (with the notations as here), the following program $\overline{P}$ is consistent if and only if Ψ is valid.

$\overline{P}$:

$y_j \vee y'_j.$		$j = 1, ..., m$
$z_l \vee z'_l.$		$l = 1, ..., t$
$z_l \leftarrow w^*.$	$z'_l \leftarrow w^*.$	$l = 1, ..., t$
$w^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}).$		$k = 1, ..., r$
$w \leftarrow z_1, \ldots, z_t, z'_1, \ldots, z'_t.$		
$\leftarrow$ not w.		

After consideration of the proof of Lemma 6.3.3, the following construction should be clear in principle. We give the program and make the few necessary observations afterwards.

P :

$x_1., \ldots, x_n.$		
$x'_1., \ldots, x'_n.$		
$w \leftarrow$ not x_i, not $x'_i, w_{x_1}, ..., w_{x_n}.$		$i = 1, ..., n$
$w_{x_i} \leftarrow$ not $x_i.$	$w_{x_i} \leftarrow$ not $x'_i.$	$i = 1, ..., n$
$y_j \vee y'_j.$		$j = 1, ..., m$
$z_l \vee z'_l.$		$l = 1, ..., t$
$z_l \leftarrow w^*.$	$z'_l \leftarrow w^*.$	$l = 1, ..., t$
$w^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}).$		$k = 1, ..., r$
$w \leftarrow z_1, \ldots z_t, z'_1, \ldots z'_t, w_{x_1}, ..., w_{x_n}$		
$\leftarrow$ not w.		

Consider a program $H \subseteq \overline{P}$ with $\leftarrow$ not $w. \in H$. Now if H is consistent, then so is $\overline{P}$ because the other rules facilitate entailment of w and can thus never be responsible for inconsistency.

Now the following observations can be made similar as in the proof of Lemma 6.3.3:

(a) Any inconsistent subset of P contains the constraint "$\leftarrow$ not w." and the inconsistency stems from it.

(b) Let $H \in SI_{min}(P)$. Then, H only contains "$\leftarrow$ not w." and rules of the form "x_i." or "x'_i.".

(c) Let $H \in SI_{min}(P)$. If H does not contain both "x_i." and "x'_i." for an $i \in \{1, ..., n\}$, then it contains either "x_i." or "x'_i." for *all* $i \in \{1, ..., n\}$.

(d) Let $i \in \{1, ..., n\}$ and let $H_i := \{x_i., \ x'_i., \ \leftarrow \text{not } w.\}$. Then, $H_i \in SI_{min}(P)$.

Again, let

$$H_I = \bigcup_{i \in \{1,...,n\}} H_i.$$

As before, $H \in SI_{min}(P) \setminus H_I$ contains either "x_i." or "x'_i." for all $i \in \{1, ..., n\}$. Let $\mathcal{H}_\Omega \subseteq 2^P$ be the set of all subprograms containing either "x_i." or "x'_i." for all $i \in \{1, ..., n\}$. We define a corresponding assignment $\omega(H) : X \to \{0, 1\}$ as before:

$$\omega(H)(x_i) := \begin{cases} 1 & \text{if } x_i. \in H, \\ 0 & \text{if } x'_i. \in H. \end{cases}$$

And the last step is as above.

(e) Let $H \in \mathcal{H}_\Omega \subseteq 2^P$. Then $H \in SI_{min}(P) \setminus H_I$ if and only if $\exists Y \forall Z \, \neg\phi(X, Y, Z)$ does not hold for the assignment $\omega(H)$ to the X-variables.

Hence, any $H \in SI_{min}(P) \setminus H_I$ corresponds to an assignment where

$$\exists Y \forall Z \, \neg\phi(X, Y, Z)$$

is *not* the case, i. e.,

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y \exists Z \, \phi(X, Y, Z)\}|$$

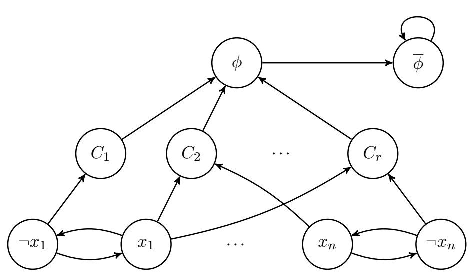
holds. □

Proposition 6.3.7. *The problems* LOWER$_{L_{ASP}}$ *and* UPPER$_{L_{ASP}}$ *are* $\mathbf{C}\Sigma_2^p$*-complete. The problem* VALUE$_{L_{ASP}}$ *is* $\#\cdot\Pi_2^p$*-complete under subtractive reductions.*

Proof. This can be inferred from the construction given in the proof of Lemma 6.3.6 as in the analogous result for normal logic programs. □

Theorem 6.4.8. VER-MIN-REPAIR$_{stb,cred}$ *is* $\mathbf{D}_1^p$*-complete.*

Proof. Membership is due to Proposition 6.4.2. Hardness is as in the proof of Theorem 6.4.5. We utilize a similar construction:

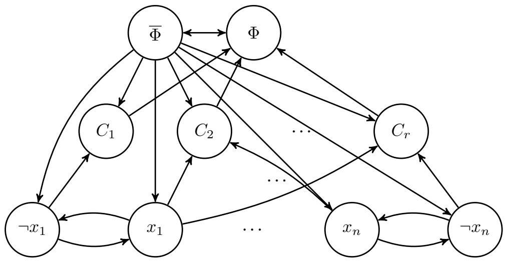


The result can be inferred similarly, reducing the problem MC as above. □

Theorem 6.4.9. VER-MIN-REPAIR$_{stb,skep}$ *is* Π_2^p*-complete.*

Proof. Membership is due to Proposition 6.4.2. For hardness, recall how to prove that skeptical reasoning is coNP-complete for stable semantics. Given a formula of the form $\Phi = \exists X : \phi(X)$ where ϕ is a formula over variables in $X = \{x_1, \ldots, x_n\}$ in 3-CNF with $\phi(x) = C_1 \wedge \ldots \wedge C_r$ recall the following construction $\mathcal{H}_\Phi$ from [93]. It has the property that the AF accepts $\overline{\Phi}$ skeptically wrt. stable semantics if and only if $\Phi = \exists X : \phi(X)$ evaluates to false.

$\mathcal{H}_\Phi$:



We augment this construction in order to show that VER-MIN-REPAIR$_{stb,skep}$ is Π_2^p-hard. We proceed as in the proof of Theorem 6.4.7. We construct an AF $F = (A, R)$ and consider $\mathcal{S} \subseteq A$ as well as the induces subframework $F_\mathcal{S}$. We need to check whether in $F_\mathcal{S}$ any argument is skeptically accepted, while this is not the case for any $F_\mathcal{S}$ with $\mathcal{S}' \subsetneq \mathcal{S}$. Hence, we may also take possible subsets of $\mathcal{S}$ into account. Assume we are given a formula $\Psi = \forall Y \, \exists X : \psi(X, Y)$ with X as above and $Y = \{y_1, \ldots, y_m\}$. Assume $\psi(X, Y) = \psi$ is in CNF with $\psi = C_1 \wedge \ldots \wedge C_r$. We will construct a subframework $F_\mathcal{S}$ with the following properties:

- $F_\mathcal{S}$ itself is consistent, i. e., there is a skeptically accepted argument,
- subsets of $\mathcal{S}$, i. e., the sets $\mathcal{S}' \subseteq \mathcal{S}$ may correspond to assignments to the Y-variables,
- there is no $\mathcal{S}'$ with $\mathcal{S}' \subsetneq \mathcal{S}$ such that the AF $F_{\mathcal{S}'}$ consistent if and only if the formula $\Psi = \forall Y \, \exists X : \psi(X, Y)$ evaluates to true.

The first and the last item together ensure that $(F, \mathcal{S})$ with $\mathcal{S}$ is a positive instance of the decision problem VER-MIN-REPAIR$_{stb,skep}$ if and only if the formula Ψ is true.

Before depicting and explaining our construction we give the arguments A of the AF $F = (A, R)$ as we did in the proof of Theorem 6.4.7. We have

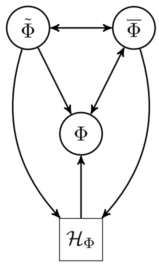
$$A = \{x_1, \neg x_1, \ldots, x_n, \neg x_n, y_1, \neg y_1, \ldots, y_m, \neg y_m,$$
$$C_1, \ldots, C_r, \Phi, \overline{\Phi}, \tilde{\Phi}, y_1?, \ldots, y_m?, all?, \top\}$$

Moreover, our subset $\mathcal{S} \subseteq A$ is

$$\mathcal{S} = \{y_1, \neg y_1, \ldots, y_m, \neg y_m\}.$$

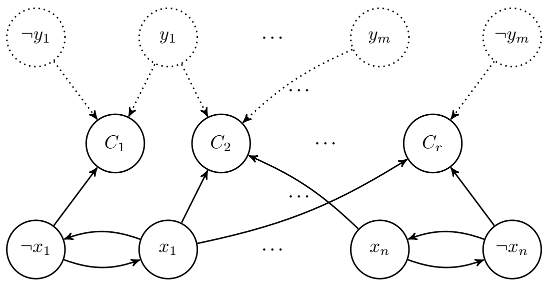
For the moment it suffices to observe that sets $\mathcal{S}'$ with $\mathcal{S}' \subseteq \mathcal{S}$ correspond to choosing y arguments.

Observe that $\mathcal{H}_\Phi$ from above possesses a stable extension containing only $\overline{\Phi}$. Our first step is consideration of a similar argument which will be called $\tilde{\Phi}$. Similar to $\overline{\Phi}$, the argument $\tilde{\Phi}$ attacks all arguments. Moreover, $\tilde{\Phi}$ and $\overline{\Phi}$ attack each other.

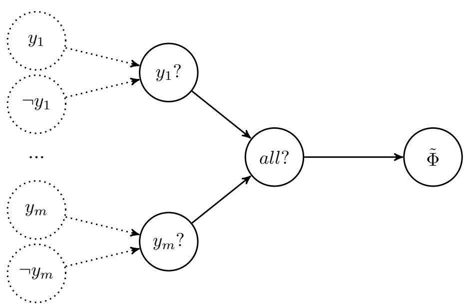


This framework has at least two stable extensions, namely $\overline{\Phi}$ and $\tilde{\Phi}$. Note the intended asymmetry between the two arguments: $\overline{\Phi}$ is attacked by Φ while $\tilde{\Phi}$ is not. The reason is as follows: The purpose of $\overline{\Phi}$ is as in the original construction to control whether there is a satisfying assignment to the given formula or not. This is why it needs to be attacked by Φ. However, $\tilde{\Phi}$ is utilized to render some subframeworks $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ inconsistent as we will see later. This is why there is no attack from Φ to $\tilde{\Phi}$.

As our next step, we consider arguments $y_1, \neg y_1, \ldots, y_m, \neg y_m$ which shall correspond to the Y-variables in the given formula $\Psi = \forall Y\, \exists X : \psi(X, Y)$. As already pointed out, they do not occur in the AF $F_{\mathcal{S}}$ as they belong to $\mathcal{S}$. So, a subset $\mathcal{S}'$ with $\mathcal{S}' \subseteq \mathcal{S}$ somewhat corresponds to a partial assignment $\omega : Y \to \{0, 1\}$. Similar to the X arguments, they attack the arguments $C_1, \ldots, C_r$ in the natural way: We have $(y_j, C_i) \in R$ iff y_j occurs in the clause C_i and $(\neg y_j, C_i) \in R$ iff $\neg y_j$ occurs in the clause C_i. Note that the Y arguments do *not* attack each other.

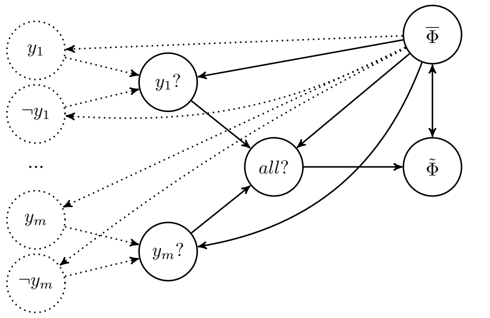


The Y arguments as well as their attacks are depicted with dotted lines to illustrate that they do *not* occur in $F_{\mathcal{S}}$. Now let us ensure only intended subsets of $\{y_1, \neg y_1, \ldots, y_m, \neg y_m\}$ are important. Recall that they shall correspond to assignments $\omega : Y \to \{0, 1\}$. Interestingly, we only need to prune away *partial* assignments, i. e., cases where there is an index j such that neither y_j nor $\neg y_j$ occurs in $F_{\mathcal{S}'}$. The case that both y_j and $\neg y_j$ occur in $F_{\mathcal{S}'}$ –actually not corresponding to a well-defined assignment– does no harm. Consider the following additional arguments and attacks:



Observe that the auxiliary argument "$all?$" attacks $\tilde{\Phi}$ only and not $\overline{\Phi}$. The meaning of this construction is as follows: In case for any j there is y_j or $\neg y_j$ occurring in $F_{\mathcal{S}'}$, each "$y_j?$" is attacked and hence, "$all?$" is defended from $\{y_1?, \ldots, y_m?\}$. It may thus occur in a stable extension. In this case, $\tilde{\Phi}$ does *not* occur in any stable extension. Otherwise, it does.

However, $\overline{\Phi}$ keeps attacking all arguments and is thus still a given possibility to find a stable extension:



We thus see the following: For any $F_{\mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ there is always a stable extension containing $\overline{\Phi}$. There is another stable extension containing $\tilde{\Phi}$ iff for at least one index j neither y_j nor $\neg y_j$ occur in $F_{\mathcal{S}'}$. Then, no argument is skeptically excepted. The AF $F_{\mathcal{S}'}$ is thus inconsistent.

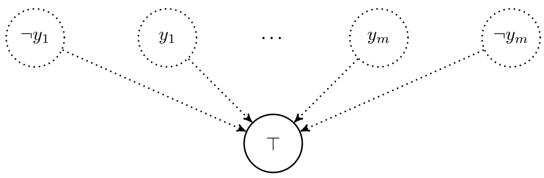
Now let us assume that for any index j there is *either* y_j *or* $\neg y_j$ occurring in $F_{\mathcal{S}'}$. Then we see that there are two cases:

1. The formula $\exists X : \psi(X, Y)$ evaluates to true under the corresponding assignment $\omega : Y \to \{0, 1\}$. Then, as in the construction from [93] there is a stable extension containing the corresponding X arguments as well as Φ, but not $\overline{\Phi}$. Hence, no argument is skeptically accepted.

2. The formula $\exists X : \psi(X, Y)$ evaluates to false under the corresponding assignment $\omega : Y \to \{0, 1\}$. Then, $\overline{\Phi}$ is skeptically accepted.

Hence, the formula $\forall Y \, \exists X : \psi(X, Y)$ is true iff the former case always occurs. Assume this is the case. Now consider a choice of the Y-variables which does not correspond to a well-defined assignment, i. e., there is j such that both y_j and $\neg y_j$ occur in $F_{\mathcal{S}'}$. It is clear that for this AF $F_{\mathcal{S}'}$ we also have the former case, i. e., no argument is skeptically accepted since this was already the case with only y_j or $\neg y_j$ occurring in the AF.

To summarize, we have: $\forall Y \, \exists X : \psi(X, Y)$ is true iff $F_{\mathcal{S}'}$ is inconsistent for all $\mathcal{S}' \subsetneq \mathcal{S}$. The latter *nearly* means that $(F, \mathcal{S})$ is a "yes" instance of VER-MIN-REPAIR$_{stb,skep}$. What we have left to do is to make sure that $F_{\mathcal{S}}$ itself is consistent, i. e., there is at least one skeptically accepted argument. The following final gadget does the job:



There is no other argument attacking $\top$. Hence, as long as no Y argument is chosen, $\top$ is skeptically accepted. As soon as a proper superset of $\mathcal{S}$ is under consideration, $\top$ is rendered pointless. □

Appendix B

Academic History

Education

since 11/2015	Doctoral study of Computer Science at Leipzig University, supervised by Prof. Dr. Gerhard Brewka
10/2009 – 09/2015	Diploma study of Business Mathematics at Leipzig University

Academic Working Experience

since 11/2018	Research associate for the DFG Research Unit “Hybrid Reasoning for Intelligent Systems”
11/2015 – 10/2018	Scholarship holder of the DFG Research Training Group “Quantitative Logics and Automata”
10/2012 – 07/2015	Teaching assistant at Leipzig University

Publications, Talks and Posters

Conference paper:

- [Ulbricht et al., 2016]: *Measuring Inconsistency in Answer Set Programs*, in: Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA’ 16)

- [Brewka et al., 2017]: *Strong Inconsistency in Non-monotonic Reasoning*, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI’17)

- [Ulbricht et al., 2018]: *Measuring Strong Inconsistency*, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI’18)

- [Baumann and Ulbricht, 2018]: *If Nothing Is Accepted - Repairing Argumentation Frameworks*, in: In Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR'18) (nominated for the Ray Reiter Best Paper Award)

- [Ulbricht, 2019]: *Repairing Non-monotonic Knowledge Bases*, accepted for publication in: Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA'19)

Journal paper:

- [Brewka et al., 2019]: *Strong Inconsistency*, in: Artificial Intelligence

Book contribution:

- [Ulbricht et al., 2018]: *Inconsistency Measures for Disjunctive Logic Programs Under Answer Set Semantics*, in: Measuring Inconsistency in Information

Additional talks:

- "Repairing Non-monotonic Knowledge Bases" (QuantLA Seminar 01/29/2019)
- "Inconsistency in Non-monotonic Logics" (QuantLA Workshop 08/31/2018)
- "Inconsistent Argumentation" (QuantLA Seminar 05/29/2018)
- "Strong Inconsistency in Non-monotonic Reasoning" (QuantLA Workshop 09/19/2017)
- "The Complexity of Inconsistency" (QuantLA Seminar 05/02/2017)
- "Measuring Inconsistency in Answer Set Programs" (QuantLA Workshop 09/23/2016)
- "Charging Strategies for Hybrid Cars" (QuantLA Seminar 01/05/2016)

Additional poster:

- "Inconsistency in Non-monotonic Logics" (11th Joint Workshop of the German Research Training Groups in Computer Science 06/12/2017)

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäßaus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

. . . .
(Ort, Datum)

. . . .
(Unterschrift)