KLM-Style Defeasible Reasoning for Datalog

by
Guy Paterson-Jones

A thesis presented in accordance with the requirements for the degree of Masters of Science, in the Faculty of Science.

Declaration

I, Guy Paterson-Jones, hereby declare that the work on which this dissertation/thesis is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

I empower the university to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Date: 29 July 2022

Signed: Signed by candidate

Acknowledgements

To my supervisors, Thomas Meyer and Giovanni Casini, I owe a wealth of gratitude. Without their guidance, encouragement and assistance, I may never have pursued my interest in mathematical logic, and this thesis would still be drifting in the ether of unfinished ideas. Thank you for putting up with my half-baked ideas, and for the gentle prods and pushes that led to innemerable insights during our conversations together. Doing a masters during a global pandemic hasn't been easy, but I can confidently say without you two it would have been impossible.

I am extremely grateful to the Centre for Artificial Intelligence Research (CAIR) and the National Research Foundation (NRF) for the financial support they have given me over the last two years. Writing this thesis without them would have taken four times as long, and it would have been half as legible!

Thank you too to Ivan Varzinczak for his humour, help and ideas during our various paper-writing scrambles. A happy Friday to you, Ivan!

Finally, a warm thank you to my friends and family, who have had to deal with many different versions of Guy during my studies. You're a bunch of nutters, and I love you all.

Abstract

In many problem domains, particularly those related to mathematics and philosophy, classical logic has enjoyed great success as a model of valid reasoning and discourse. For real-world reasoning tasks, however, an agent typically only has partial knowledge of its domain, and at most a statistical understanding of relationships between properties. In this context, classical inference is considered overly restrictive, and many systems for *non-monotonic reasoning* have been proposed in the literature to deal with these tasks. A notable example is the KLM framework, which describes an agent's defeasible knowledge qualitatively in terms of conditionals of the form "if A, then typically B".

The goal of this research project is to investigate KLM-style semantics for defeasible reasoning over Datalog knowledge bases. Datalog is a declarative logic programming language, designed for querying large deductive databases. Syntactically, it can be viewed as a computationally feasible fragment of first-order logic, so this continues a recent line of work in which the KLM framework is lifted to more expressive languages.

Contents

Chapter 1

Introduction

In this chapter we will give a brief overview of what defeasible reasoning is, what the KLM framework has to say about it and where Datalog fits into the story. Our starting point is the observation that in many problem domains, there is a need for AI to perform *reasoning tasks*. By this, we mean the process of drawing new conclusions from a set of established beliefs about the problem domain. A simple example of this is a database system - in the process of answering a complex query, a database will typically have to derive new information from its existing body of facts (in a SQL database, for instance, a query might require the construction of a new table [20]). A more recent example might be something like GPT3, a deep neural network that is able to interpret and answer a broad range of questions in English [6].

There are as many different kinds of reasoning as there as colours of the rainbow. Science, for instance, is heavily dependent on *induction*, which is the process of forming good general principles from an incomplete body of evidence. Economists care about *decision theory*, where one reasons about the different choices one can make in a given situation. Or perhaps one is trying to understand why a computer program has just crashed, in which case one can make use of *abduction*, which is about working out the most likely explanation for a set of observations.

Closer to the topic of this thesis is the study of *mathematical logic*, which at its core is all about reasoning with *absolute*, *a priori* knowledge and assumptions [27]. One perspective is that the role of a mathematician is to take a set of axioms, which are assumed to be true beyond a shadow of doubt, and to apply a sequence of logically sound rules of inference in order to arrive at new conclusions, which by virtue of the axioms are thus *also* true beyond a shadow of doubt. This is called *deductive reasoning*, and an example of a valid deduction is the following:

"Socrates is a man."

"All men are mortal."
"Therefore, Socrates is mortal."

This is a valid deduction because whenever the *premises* are true, the *conclusion* follows logically as well. If Socrates is a man, and we assume that all men are mortal (though some may argue against this!), then it is *necessarily* true that Socrates is mortal. This kind of deductive argument is ubiquitous in mathematics and philosophy - so much so, that a common misconception is that reasoning and deduction are the same thing [31]. A moment's reflection, however, will reveal that things are not so simple in the real world. Consider the following example of an *invalid* deduction:

"Tommie is a bird."
"Birds usually fly."
"Therefore, Tommie can fly."

This is *not* a valid deduction, because it is possible for the conclusion to be false even if the premises are true. Suppose, for example, that Tommie happened to be a penguin. In this case, the first two sentences would be true, but the last sentence would almost certainly be false. Nevertheless, the argument has some intuitive appeal. If all we know about Tommie is that he is a bird, it's reasonable to expect him to be able to fly, since almost every bird we see on a day-to day basis can do so!

Humans perform this kind of heuristic reasoning all the time. We are endowed with a small number of sensory organs, all of which are limited in their precision and often somewhat faulty. As a result, we never have perfect information about the world around us, and the only way to come to any conclusions at all is to generalise beyond that which we know for certain. While it would be technically incorrect to call this *deduction*, it is certainly a form of *reasoning*.

A hallmark of this kind of reasoning is that its conclusion are often *defeasible*, meaning they may have to be withdrawn in light of new information. In the example above, if we were to add the additional premise that *"Tommie is a penguin"*, then we would no longer be justified in concluding that Tommie can fly.

Notice the crucial difference here between defeasible reasoning and deduction. A valid deduction is *always* valid, no matter how many new premises we tack on after the fact. This property is known in the literature as *monotonicity*, and is one of the hallmarks of mathematical reasoning.

Thus, in order to understand the defeasible reasoning process, we need to come to terms with non-monotonicity. Fortunately, constructing non-monotonic models of reasoning is not hard - in the literature, there are a multitude of options available to the intrepid reasoner (see the book by Makinson [26] on the topic, for instance). In a certain sense, the KLM framework is just one of

these options. What distinguishes it from its peers, however, is its focus on the *properties* that a non-monotonic consequence relation might enjoy.

The KLM framework provides a mathematical characterisation of *any* non-monotonic model of reasoning satisfying certain inferential properties [22, 25]. These properties have come to be known as the *rationality postulates*, and provide a useful tool for comparing and delineating the field of non-monotonic logic. Beyond this, the semantics of the KLM framework can also be used to construct some particularly elegant and well-behaved *algorithms* for non-monotonic reasoning, such as the *Rational Closure* construction. For these reasons, the KLM framework has been enormously influential in the study of defeasible reasoning, and has inspired a flurry of further research on the topic (such as this thesis!) that continues to this day. We give an overview of the original KLM framework in Chapter 3.

In their seminal paper on the KLM framework, Kraus et al. [22] study the problem of performing defeasible reasoning when one's background assumptions and knowledge are encoded in the formalism of *propositional logic*. Propositional logic is a simple model of mathematical reasoning in which the basic objects are *propositions*, or complete statements about one's problem domain. For instance, the sentence *"Socrates is a man"* is a proposition, as is the sentence *"Socrates is mortal"*. One of the advantages of propositional logic is that deduction for propositions is *computable*, meaning there exist algorithms for determining whether a given proposition can be deduced from a set of assumptions.

On the other hand, propositional logic is quite restrictive. It offers no way to reason about more nuanced statements such as *"every person has something that they like more than anything else"*, and in particular isn't expressive enough to provide a full account of mathematical reasoning. The gold standard for this is *first-order logic*, which allows one to model statements containing different kinds of *predicates* (such as *"X is red"* or *"X likes Y"*) and *quantifiers* (such as *"there exists some X"* or *"every X is Y"*). First-order logic offers significantly more flexibility in how one describes and reasons about a problem domain. The price one pays for this, however, is that deduction for first-order logic is *incomputable*. There is no algorithm, even in principle, that can tell you how to reason deductively about first-order statements!

Datalog is a database query language that sits somewhere between the two. It is based on first-order logic, but restricts the kinds of quantifiers and predicates that are allowed when describing a statement. This lets it capture a wide range of problem domains, and preserves the property of being computable. Alongside an elegant syntax, this has made Datalog into a versatile tool for working with large deductive databases [12].

Our goal in this thesis, as we will review in the next section, is to extend the KLM framework to the Datalog setting, and provide a mathematical model of defeasible reasoning for Datalog knowledge bases.

Two papers were published based on the work done in this thesis. The first

studies a boolean extension of the propositional KLM framework, and was published in the proceedings of SACAIR 2021 [29]. The second contains an account of KLM-style reasoning for Datalog, and was published in the proceedings of NMR 2021 [8]. A third paper on Defeasible Datalog is currently under review for IJCAI 2022, and we are awaiting notification.

1.1 Research Objectives

There are a number of active lines of research being explored on increasing the scope of the KLM framework. At the moment, the most promising can be summarised as follows:

1. The syntax of the KLM framework can be modified to support more expressive conditionals, such as negated conditionals [2] or conditionals with fine-grained typicality [3].
2. Inferential limitations with Rational Closure can be overcome by considering various extensions, such as Lexicographic Closure [23] or Relevant Closure [11].
3. Analogues of the KLM framework can be defined over more expressive base logics, such as the Description Logic ALC [19, 5].

The aim of this thesis is to investigate and continue this last line of research. Specifically, we investigate analogues of KLM-style inference for *Datalog*, a declarative logic programming language designed for querying deductive databases. Datalog supports rules and predicates of arbitrary arity, which makes it much more expressive than propositional logic, and also slightly more expressive than the variety of Description Logics (such as ALC) for which KLM-style defeasibility has already been studied. As such, it provides an interesting test-case that should be understood before KLM-style defeasibility is applied to *extremely* expressive settings like unrestricted first-order logic.

1.2 Dissertation Outline

In Chapter 2, we review some basic concepts of order theory and mathematical logic that will be used in the rest of the thesis. Readers who have some experience with propositional logic, first-order logic and the notion of a partial order can likely skip this chapter. Chapter 3 introduces KLM-style defeasible reasoning for propositional logic, based on the seminal papers by Kraus, Lehmann and Magidor [22, 25]. We cover the main concepts, namely the *rationality postulates*, *ranked interpretations* and *rational closure*. This lays the groundwork for the more expressive theories in the rest of the thesis. In Chapter 4, we look

at the simple Description Logic ALC, and review the literature on KLM-style defeasible reasoning for ALC. We discuss some of the difficulties that more expressive logics present for defeasible reasoning. In Chapter 5 we introduce Datalog, a computationally-flavoured fragment of first-order logic. We focus on the classical syntax and semantics of Datalog, and present some of the basic non-monotonic extensions of Datalog that exist in the literature already. Chapter 6 is the meat of the thesis, and introduces the building blocks for a KLM-style theory of defeasible reasoning over Datalog knowledge bases. We introduce Datalog analogues of the rationality postulates, cover various possible semantics for Defeasible Datalog formulas, and finish by analysing several viable notions of entailment for Defeasible Datalog knowledge bases. Finally, in Chapter 7, we present our conclusions from this research project and suggest several lines of future research that may be fruitful.

Chapter 2

Mathematical Preliminaries

In this chapter, we collect some definitions and results about basic mathematics that we will need in future chapters. In particular, we will cover the basics of *order theory*, *propositional logic* and *first-order logic*.

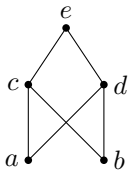
2.1 Order Theory

A binary relation on a set P is a subset $r \subseteq P \times P$. A *strict partial order* on a set P is any binary relation $\prec \subseteq P \times P$ satisfying the following two conditions, where x, y, z refer to any elements of P [13, p. 2]:

1. $x \prec y$ and $y \prec z$ implies $x \prec z$
2. $x \prec y$ implies $y \not\prec x$

Given a strict partial order $\prec$ on P, we say x is *covered* by y iff $x \prec y$ and there is no z such that $x \prec z \prec y$, a fact we denote by $x \lhd y$ [13, p. 11]. If P is finite, then $x \prec y$ iff there exists a sequence of coverings $x \lhd x_1 \lhd \cdots \lhd x_n \lhd y$. Thus, in the finite case, strict partial orderings are determined by their covering relation. This fact gives us a convenient diagrammatic notation for finite strict partial orders:

Example 1. *Let $P = \{a, b, c, d, e\}$, and consider the following diagram:*



The edges in the diagram represent the covering relation, going downwards. Thus in this diagram we have $c \lhd e$, $d \lhd e$, $a \lhd d$, etcetera. Translating between the covering relation and a strict partial order $\prec$, we can see, for instance, that $a \prec e$ and $b \prec d$.

These kinds of diagrams are known as *Hasse diagrams* [13, p. 11]. A given strict partial order may have many possible Hasse diagrams, of course, but we will strive to minimise confusion whenever we draw one.

If $Q \subseteq P$ is a subset of a strict partially ordered set P, then an element $x \in Q$ is a *maximal* element of Q iff there is no $y \in Q$ such that $x \prec y$. Similarly, $x \in Q$ is a *minimal* element of Q iff there is no $y \in Q$ such that $y \prec x$ [13, p. 16]. We denote the set of maximal and minimal elements of Q by $\max_{\prec} Q$ and $\min_{\prec} Q$ respectively. Note that minimal and maximal elements do not always exist:

Example 2. *Consider the set of integers $\mathbb{Z}$, and let $<$ be the usual ordering. Then $\min_< \mathbb{Z} = \max_< \mathbb{Z} = \emptyset$. On the other hand, if we consider the set of naturals $\mathbb{N} \subset \mathbb{Z}$, then $\max_< \mathbb{N} = \emptyset$ and $\min_< \mathbb{N} = \{0\}$.*

In the case that $\max_{\prec} Q = \{x\}$ or $\min_{\prec} Q = \{x\}$ for some singleton $x \in Q$, then we say x is the *maximum* or *minimum* element of Q respectively. We say P is *totally ordered* (or a *chain*) iff for every $x, y \in P$ either $x = y$, $x \prec y$ or $y \prec x$ [13, p. 3]. A totally ordered set P is *well-ordered* (or a *well-order*) iff every subset $Q \subseteq P$ has a minimum element.

Example 3. *While $\mathbb{Z}$ and $\mathbb{N}$ are both totally ordered sets, only $\mathbb{N}$ is a well-order, as $\min_< \mathbb{Z} = \emptyset$ and hence $\mathbb{Z}$ has no minimum element.*

Finally, we note the following definition from a paper by Lehmann et al. [25], which will come in handy later:

Definition 1. *[25, p. 19] Let $\prec$ be a strict partial order on P. Then we say $\prec$ is* modular *iff any of the following equivalent conditions hold, where x, y, z are any elements of P:*

1. *$x \not\prec y$, $y \not\prec x$ and $z \prec x$ implies $z \prec y$*
2. *$x \prec y$ implies $z \prec y$ or $x \prec z$*
3. *$x \not\prec y$ and $y \not\prec z$ implies $x \not\prec z$*
4. *there exists a totally ordered set $\langle \Omega, < \rangle$ and function $r : P \to \Omega$ such that $x \prec y$ iff $r(x) < r(y)$.*

2.2 Propositional Logic

The language of propositional logic is defined over a set of *propositional atoms*, denoted $\mathcal{P}$, which represent the atomic statements that one wants to reason

about. When explicitly writing elements of $\mathcal{P}$, we will use lowercase latin characters such as p, q, r. The set of propositional formulas over $\mathcal{P}$ is denoted $\mathcal{L}^{\mathcal{P}}$, and is defined by the following rules (where we've dropped the superscript $\mathcal{P}$ for convenience) [27, p. 116]:

1. For all $\mathsf{p} \in \mathcal{P}$, $\mathsf{p} \in \mathcal{L}$.
2. For all $\alpha \in \mathcal{L}$, $\neg\alpha \in \mathcal{L}$.
3. For all $\alpha, \beta \in \mathcal{L}$, $\alpha \vee \beta \in \mathcal{L}$.

Other logical connectives can be defined in terms of $\neg$ and $\vee$:

1. $\alpha \wedge \beta$ is an alias for $\neg(\neg\alpha \vee \neg\beta)$.
2. $\alpha \to \beta$ is an alias for $\neg\alpha \vee \beta$.
3. $\alpha \leftrightarrow \beta$ is an alias for $(\alpha \to \beta) \wedge (\beta \to \alpha)$.

A *theory* (or *deductively closed theory*) is a set of propositional formulas that is equal to its closure under the following Hilbert-style deduction rules, where α, β, γ refer to any propositional formulas [27, p. 214]:

1. $\alpha \to (\beta \to \alpha) \in \Gamma$.
2. $(\neg\alpha \to \neg\beta) \to (\beta \to \alpha) \in \Gamma$.
3. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)) \in \Gamma$.
4. if $\alpha \in \Gamma$ and $\alpha \to \beta \in \Gamma$ then $\beta \in \Gamma$.

The closure of a theory Γ under these deduction rules is denoted $\mathrm{Th}(\Gamma)$, and thus Γ is deductively closed iff $\Gamma = \mathrm{Th}(\Gamma)$. A theory Γ is *complete* iff for every formula $\alpha \in \mathcal{L}$, $\alpha \in \mathrm{Th}(\Gamma)$ or $\neg\alpha \in \mathrm{Th}(\Gamma)$ [27, p. 198].

Propositional formulas and theories are interpreted by *valuations*, which are defined to be any atomic truth assignment $u : \mathcal{P} \to \{0, 1\}$. Truth assignments are extended to non-atomic formulas as follows [27, p. 121]:

1. $u(\neg\alpha) = 1$ iff $u(\alpha) = 0$.
2. $u(\alpha \vee \beta) = 1$ iff $u(\alpha) = 1$ or $u(\beta) = 1$.

The set of valuations over $\mathcal{P}$ is denoted $\mathcal{U}^{\mathcal{P}}$, and similarly to $\mathcal{L}$ we will drop the superscript $\mathcal{P}$ where possible. We say a valuation $u \in \mathcal{U}$ *satisfies* a formula $\alpha \in \mathcal{L}$, denoted $u \Vdash \alpha$, iff $u(\alpha) = 1$. A valuation satisfies a theory iff it satisfies every formula in the theory.

We say a formula $\alpha \in \mathcal{L}$ is *classically entailed* (or *logically follows from*) a theory Γ iff every model of Γ satisfies α. The classical entailment relation $\models \subseteq 2^{\mathcal{L}} \times \mathcal{L}$ is defined by $\Gamma \models \alpha$ iff Γ classically entails α [27, p. 121]. As noted by Tarski [34], the classical entailment relation satisfies the following three properties, known as *Inclusion*, *Cumulativity* and *Monotonicity* respectively:

$$\begin{aligned}
&(\textsc{Incl}) && \alpha \in \Gamma \text{ implies } \Gamma \models \alpha \\
&(\textsc{Cumu}) && \Gamma \models \alpha \text{ and } \Gamma \cup \{\alpha\} \models \beta \text{ implies } \Gamma \models \beta \\
&(\textsc{Mono}) && \Gamma \models \alpha \text{ implies } \Gamma \cup \{\beta\} \models \alpha
\end{aligned}$$

The following theorem is fundamental, and expresses the equivalence between deduction and entailment for propositional logic:

Theorem 1. *[27, p. 123] For any theory $\Gamma \subseteq \mathcal{L}$ and formula $\alpha \in \mathcal{L}$, $\Gamma \models \alpha$ iff $\alpha \in \mathrm{Th}(\Gamma)$.*

To every valuation $u \in \mathcal{U}$ we assign a corresponding theory called its *satisfaction set*, defined by $\Gamma_u = \{\alpha \in \mathcal{L} : u \Vdash \alpha\}$. Then the following corollary of Theorem 1 shows that there is an equivalence between valuations and closed or complete theories:

Corollary 1. *A theory $\Gamma \subseteq \mathcal{L}$ is deductively closed iff there exists some set of valuations $U \subseteq \mathcal{U}$ such that $\Gamma = \bigcap_{u \in U} \Gamma_u$. Γ is complete iff there exists some valuation $u \in \mathcal{U}$ such that $\Gamma = \Gamma_u$.*

Lastly, we will describe some non-standard notation for valuations that will come in handy later. In the case that $\mathcal{P}$ is finite, consider the set of strings of the form $p_1 p_2 \dots p_{|\mathcal{P}|}$, where each p_i is either a propositional atom p or its negation, denoted by $\overline{\mathsf{p}}$. Such a string will be considered *admissible* if every propositional atom appears exactly once in the string (possibly as a negation). Each admissible string corresponds to a unique valuation $v : \mathcal{P} \to \{0, 1\}$, defined by setting $v(\mathsf{p}) = 1$ if the string contains p, or $v(\mathsf{p}) = 0$ if the string contains $\overline{\mathsf{p}}$.

Example 4. *Suppose that $\mathcal{P} = \{\mathsf{p}, \mathsf{q}, \mathsf{r}\}$, and consider the valuation v defined by $v(p) = 0$, $v(q) = 1$ and $v(r) = 1$. Then v can be described by the admissible string $\overline{\mathsf{p}}\mathsf{q}\mathsf{r}$. Note that this description is not unique - any permutation of the string would do as well.*

2.3 First-Order Logic

First-order logic expands the language of propositional logic, allowing for the expression of formulas at a finer level of granularity than atomic propositions. A *first-order language* is a tuple $\Sigma = \langle \textsc{pred}, \textsc{func}, \textsc{const}, \textsc{var} \rangle$, which has the following interpretation [36, p. 23]:

1. PRED is a set of *predicate symbols*, which we require to be non-empty.
2. FUNC is a set of *function symbols*, which may be empty.
3. CONST is a set of *constant symbols*, which may be empty.
4. VAR is a set of *variable symbols*, which we require to be non-empty (and which we will typically take to be countably infinite, though this is not a requirement).

These sets are assumed to be disjoint, and furthermore we assume that each predicate and function symbol $\alpha \in \text{PRED} \cup \text{FUNC}$ has an associated *arity*, denoted $\mathfrak{ar}(\alpha) \in \mathbb{N}$. Given a first-order language Σ, the set $\mathcal{T}$ of *terms* over Σ is defined by the following rules [36, p. 24]:

1. If $\mathsf{c} \in \text{CONST}$ is a constant symbol, then $\mathsf{c} \in \mathcal{T}$.
2. If $\mathsf{x} \in \text{VAR}$ is a variable symbol, then $\mathsf{x} \in \mathcal{T}$.
3. If $\mathsf{f} \in \text{FUNC}$ is a function symbol and $t_1, \ldots, t_{\mathfrak{ar}(\mathsf{f})}$ are terms, then $\mathsf{f}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{f})}) \in \mathcal{T}$.

An *atomic formula* over Σ is defined to be anything of the form $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})})$, where $\mathsf{p} \in \text{PRED}$ is a predicate symbol and $t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})} \in \mathcal{T}$ are terms. The set of *formulas* over Σ is denoted $\mathcal{L}^\Sigma$, and is defined inductively in terms of atomic formulas as follows [36, p. 24]:

1. If $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})})$ is an atomic formula, then $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})}) \in \mathcal{L}^\Sigma$.
2. If $\alpha \in \mathcal{L}^\Sigma$ is a formula, then $\neg\alpha \in \mathcal{L}^\Sigma$.
3. If $\alpha, \beta \in \mathcal{L}^\Sigma$ are formulas, then $\alpha \vee \beta \in \mathcal{L}^\Sigma$.
4. If $\alpha \in \mathcal{L}^\Sigma$ is a formula and $\mathsf{x} \in \text{VAR}$ is a variable symbol, then $\forall \mathsf{x}.\alpha \in \mathcal{L}^\Sigma$.

When there is no ambiguity we will drop the superscript and refer to the set of formulas over Σ by $\mathcal{L}$. As in the propositional case, other logical connectives can be defined in terms of $\neg$, $\vee$ and $\forall$:

1. $\alpha \wedge \beta$ is an alias for $\neg(\neg\alpha \vee \neg\beta)$.
2. $\alpha \to \beta$ is an alias for $\neg\alpha \vee \beta$.
3. $\alpha \leftrightarrow \beta$ is an alias for $(\alpha \to \beta) \wedge (\beta \to \alpha)$.
4. $\exists \mathsf{x}.\alpha$ is an alias for $\neg(\forall \mathsf{x}.\neg\alpha)$.

Under the standard semantics, first-order formulas and theories over a language Σ are interpreted by *first-order structures*, which are defined to be tuples $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain*, and $\cdot^{\mathcal{I}}$ is an *interpretation function* that acts as follows [36, p. 167]:

1. Constant symbols $\mathsf{c} \in \textsc{const}$ are interpreted by domain elements $\mathsf{c}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
2. Predicate symbols $\mathsf{p} \in \textsc{pred}$ are interpreted by relations $\mathsf{c}^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^{\mathfrak{ar}(\mathsf{p})}$.
3. Function symbols $\mathsf{f} \in \textsc{func}$ are interpreted by functions $\mathsf{f}^{\mathcal{I}} : (\Delta^{\mathcal{I}})^{\mathfrak{ar}(\mathsf{f})} \to \Delta^{\mathcal{I}}$.

A first-order structure provides an interpretation for all constant, predicate and function symbols in a given first-order language, as well as for all variable symbols bound by a universal or existential quantifier. The missing ingredient before we can talk about satisfaction is an interpretation for the *unbound* variable symbols. This is given by an *assignment*, which is defined to be a function $\upsilon : \textsc{var} \to \Delta^{\mathcal{I}}$ [36, p. 168]. Every such assignment can be extended to the set of all terms of the language as follows:

Lemma 1. *[36, p. 168] Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ be a first-order structure, and $\upsilon : \textsc{var} \to \Delta^{\mathcal{I}}$ some assignment. Then there is a unique function $\hat{\upsilon} : \mathcal{T} \to \Delta^{\mathcal{I}}$ such that:*

1. *If t is a variable symbol $\mathsf{x} \in \textsc{var}$, then $\hat{\upsilon}(t) = \upsilon(\mathsf{x})$.*
2. *If t is a constant symbol $\mathsf{c} \in \textsc{const}$, then $\hat{\upsilon}(t) = \mathsf{c}^{\mathcal{I}}$.*
3. *If t is a term of the form $\mathsf{f}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{f})})$ for some function symbol $\mathsf{f} \in \textsc{func}$, then $\hat{\upsilon}(t) = \mathsf{f}^{\mathcal{I}}(\hat{\upsilon}(t_1), \ldots, \hat{\upsilon}(t_{\mathfrak{ar}(\mathsf{f})}))$,*

For any assignment υ, variable symbol $\mathsf{x} \in \textsc{var}$ and domain element $d \in \Delta^{\mathcal{I}}$, we denote by υ^{x}_d the assignment that agrees with υ everywhere except that x is mapped to d. Given a first-order structure $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ and an assignment $\upsilon : \textsc{var} \to \Delta^{\mathcal{I}}$, we say a formula is *satisfied* by $\mathcal{I}$ under the assignment υ according to the following rules [36, p. 169]:

1. An atom $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})})$ is satisfied by $\mathcal{I}$ under υ iff $(\hat{\upsilon}(t_1), \ldots, \hat{\upsilon}(t_{\mathfrak{ar}(\mathsf{p})})) \in \mathsf{p}^{\mathcal{I}}$.
2. A formula $\neg\alpha$ is satisfied by $\mathcal{I}$ under υ iff α is not.
3. A formula $\alpha \vee \beta$ is satisfied by $\mathcal{I}$ under υ iff α is or β is.
4. A formula $\forall \mathsf{x}.\alpha$ is satisfied by $\mathcal{I}$ under υ iff α is satisfied by $\mathcal{I}$ under υ^{x}_d for every $d \in \Delta^{\mathcal{I}}$.

The fact that α is satisfied by $\mathcal{I}$ under υ is denoted by $\langle \mathcal{I}, \upsilon \rangle \Vdash \alpha$. In the case that α is satisfied by $\mathcal{I}$ under every possible assignment, we simplify the notation and just write $\mathcal{I} \Vdash \alpha$. Whenever $\Gamma \subseteq \mathcal{L}$ is a set of formulas, we write $\langle \mathcal{I}, \upsilon \rangle \Vdash \Gamma$ or $\mathcal{I} \Vdash \Gamma$ to mean $\langle \mathcal{I}, \upsilon \rangle \Vdash \alpha$ or $\mathcal{I} \Vdash \alpha$ for every $\alpha \in \Gamma$ respectively.

Given a set of formulas $\Gamma \subseteq \mathcal{L}$, we say a formula α is *classically entailed* (or *logically follows from*) Γ iff whenever $\langle \mathcal{I}, \upsilon \rangle \Vdash \Gamma$ we have $\langle \mathcal{I}, \upsilon \rangle \Vdash \alpha$. As in the propositional case, we denote this by $\Gamma \models \alpha$. A *theory* is defined to be a set of formulas that is closed under classical entailment, and a *complete theory* is a theory that contains either α or $\neg\alpha$ for every $\alpha \in \mathcal{L}$.

We note that while a first-order proof theory and analogue of Theorem 1 exists, we will not make use of it in this thesis. The interested reader is encouraged to consult Wolf [36] for an overview of first-order logic (inlcuding proof theory) or Monk [27] for a more comprehensive textbook.

While first-order structures are the standard tool for interpreting first-order theories, they are far from the only option. A simpler alternative that is commonplace in logic programming is *Herbrand semantics*, which intuitively is a semantics in which *terms interpret themselves*. Formally, we define the *Herbrand universe* and *Herbrand base* of a first-order language Σ as follows [18, p. 2]:

Definition 2. *Let* $\Sigma = \langle \textsc{pred}, \textsc{func}, \textsc{const}, \textsc{var} \rangle$ *be a first-order language. Then the Herbrand universe of* Σ *is the set of ground terms over* Σ*, i.e. terms without variables in them:*

$$\mathbb{U}^\Sigma = \{t \in \mathcal{T}^\Sigma : t \text{ contains no variables}\}$$

The Herbrand base of Σ *is the set of ground atoms over* Σ*, i.e. atomic formulas containing only terms from* $\mathbb{U}^\Sigma$*:*

$$\mathbb{B}^\Sigma = \{\mathsf{p}(t_1, \ldots, t_{\mathtt{ar}(\mathsf{p})}) \in \mathcal{L}^\Sigma : t_1, \ldots, t_{\mathtt{ar}(\mathsf{p})} \in \mathbb{U}^\Sigma\}$$

As usual, we will omit the superscript Σ when it is obvious from context. With these definitions under our belt, we define a *Herbrand interpretation* to be any subset of the Herbrand base $\mathcal{H} \subseteq \mathbb{B}$ [18, p. 3]. The idea here is that a ground atom is assumed to be satisfied by $\mathcal{H}$ iff it is contained in $\mathcal{H}$. Herbrand interpretations are thus in some sense a first-order generalisation of propositional valuations.

Note that there is an edge-case here when CONST is empty. In this case, both the Herbrand universe and the Herbrand base will be empty as well, and thus the only valid Herbrand interpretation over such a language is the empty set. This is consistent with what follows, but should be kept in mind as a possibility.

The main difficulty is interpreting quantifiers with Herbrand semantics. To this end, we define a *substitution* over a language Σ to be any function $\varphi : \textsc{var} \to \mathcal{T}$ assigning terms to each variable symbol. If the image of a substitution φ contains

only ground terms, i.e. $\mathrm{im}(\varphi) \subseteq \mathbb{U}$, then we say that φ is a *ground substitution*. We denote by $\varphi_{\mathtt{1}}$ the *identity substitution*, i.e. the substitution that sends every variable symbol to itself. Like assignments, the domain of a substitution can be uniquely extended to the set of terms of Σ:

Lemma 2. *Let $\Sigma = \langle \textsc{pred}, \textsc{func}, \textsc{const}, \textsc{var} \rangle$ be a first-order language, and $\varphi : \textsc{var} \to \mathcal{T}$ some substitution. Then there is a unique function $\hat{\varphi} : \mathcal{T} \to \mathcal{T}$ such that:*

1. *If t is a variable symbol $\mathsf{x} \in \textsc{var}$, then $\hat{\varphi}(t) = \varphi(\mathsf{x})$.*
2. *If t is a constant symbol $\mathsf{c} \in \textsc{const}$, then $\hat{\varphi}(t) = \mathsf{c}$.*
3. *If t is a term of the form $\mathsf{f}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{f})})$, then $\hat{\varphi}(t) = \mathsf{f}(\hat{\varphi}(t_1), \ldots, \hat{\varphi}(t_{\mathfrak{ar}(\mathsf{f})}))$.*

Given any substitution φ and formula $\alpha \in \mathcal{L}$, we define the substituted formula $\varphi(\alpha)$ as follows:

1. If α is an atom $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})}) \in \mathcal{L}$, then $\varphi(\alpha) = \mathsf{p}(\hat{\varphi}(t_1), \ldots, \hat{\varphi}(t_{\mathfrak{ar}(\mathsf{p})}))$.
2. If α is of the form $\neg\beta \in \mathcal{L}$, then $\varphi(\alpha) = \neg\varphi(\beta)$.
3. If α is of the form $\beta \vee \gamma \in \mathcal{L}$, then $\varphi(\alpha) = \varphi(\beta) \vee \varphi(\gamma)$.
4. If α is of the form $\forall \mathsf{x}.\beta \in \mathcal{L}$, then $\varphi(\alpha) = \forall \mathsf{x}.\varphi^*(\beta)$, where φ^* is the substitution that agrees with φ everywhere except that x is mapped to x.

Note that substitutions only affect the *free* variables in a formula. Given a Herbrand interpretation $\mathcal{H} \subseteq \mathbb{B}$, we say a formula $\alpha \in \mathcal{L}$ is *satisfied* by $\mathcal{H}$ according to the following rules [18, p. 3]:

1. If α is a ground atom $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})}) \in \mathcal{L}$, then α is satisfied by $\mathcal{H}$ iff $\alpha \in \mathcal{H}$.
2. If α is of the form $\neg\beta \in \mathcal{L}$, then α is satisfied by $\mathcal{H}$ iff β is not satisfied by $\mathcal{H}$.
3. If α is of the form $\beta \vee \gamma \in \mathcal{L}$, then α is satisfied by $\mathcal{H}$ iff β is satisfied by $\mathcal{H}$ or γ is satisfied by $\mathcal{H}$.
4. If α is of the form $\forall \mathsf{x}.\beta \in \mathcal{L}$, then α is satisfied by $\mathcal{H}$ iff $\varphi(\beta)$ is satisfied by $\mathcal{H}$ for every substitution φ that agrees with $\varphi_{\mathtt{1}}$ everywhere except that x is mapped to some constant $\mathsf{c} \in \textsc{const}$.

As before, we denote the fact that $\mathcal{H}$ satisfies α by $\mathcal{H} \Vdash \alpha$, and for any set of formulas $\Gamma \subseteq \mathcal{L}$ we write $\mathcal{H} \Vdash \Gamma$ to mean $\mathcal{H} \Vdash \alpha$ for every $\alpha \in \Gamma$. Note that we have only defined satisfaction for formulas without free variables. This is

enough for our purposes in this thesis, but a fully general definition can easily be given if desired.

The reader may be wondering why Herbrand semantics isn't the standard for interpreting first-order logic. After all, it is significantly simpler, and possibly even more intuitive. The difficulty is that Herbrand semantics is *not* equivalent to the use of first-order structures:

Lemma 3. *[18, p. 5] Consider a first-order language with a unary predicate* p*, a unary function symbol* f*, a constant symbol* c *and a countable set of variable symbols* $\{\mathsf{x}, \mathsf{y}, \dots\}$*. Then the set* $\Gamma = \{\mathsf{p}(\mathsf{c}), \mathsf{p}(\mathsf{f}(\mathsf{c})), \mathsf{p}(\mathsf{f}(\mathsf{f}(\mathsf{c}))), \dots, \exists \mathsf{x}.\neg\mathsf{p}(\mathsf{x})\}$ *is satisfiable by a first-order structure, but not by any Herbrand interpretation.*

The differences between Herbrand semantics and standard first-order semantics manifest in a number of ways. Standard first-order semantics is *compact*, for instance, meaning that an infinite set of formulas is satisfiable by a first-order structure iff every finite subset is. In contrast, the example in Lemma 3 shows that Herbrand semantics is *not* compact. A related difference is the fact that there exists a sound and complete proof theory for standard first-order semantics, whereas no such proof theory can exist for Herbrand semantics [18, p. 5].

On the other hand, these technical limitations are generally not a problem when working with restricted fragments of first-order logic, such as Datalog. In fact, as we will see later, Herbrand semantics is sound, complete and compact for Datalog-like fragments (see Section 6.1, for instance). In cases like this, the simplicity of Herbrand semantics more than makes up for its formal inadequacies, and we will adopt it whenever possible.

The last thing we turn to for first-order logic is a notational convenience that we will make extensive use of later. So far, we have defined universal and existential formulas in terms of a single variable symbol, such as $\forall \mathsf{x}.\alpha$ or $\exists x.\alpha$. Like all first-order connectives, this can be nested to obtain quantification over multiple variables simultaneously, such as $\forall \mathsf{x}.(\forall \mathsf{y}.(\exists \mathsf{z}.\alpha))$. Notationally this becomes cumbersome, however, and is particularly painful when describing inference rules for quantified formulas, a topic we will turn to later when we study Defeasible Datalog.

To make life simpler, we introduce the notation $\forall \vec{\mathsf{x}}.\alpha$, where $\vec{\mathsf{x}} = \langle \mathsf{x}_1, \dots, \mathsf{x}_n \rangle \in \textsc{var}^n$ is any ordered tuple of variable symbols. The formula that this notation stands for is defined inductively as follows:

1. If $\vec{\mathsf{x}} = \langle\rangle$ is a nullary tuple, then $\forall \vec{\mathsf{x}}.\alpha$ stands for α.
2. If $\vec{\mathsf{x}} = \langle \mathsf{x}_1 \rangle$ is a unary tuple, then $\forall \vec{\mathsf{x}}.\alpha$ stands for $\forall \mathsf{x}_1.\alpha$.
3. If $\vec{\mathsf{x}} = \langle \mathsf{x}_1, \dots, \mathsf{x}_n \rangle$ is an n-ary tuple for $n > 1$, then $\forall \vec{\mathsf{x}}.\alpha$ stands for $\forall \mathsf{x}_1.(\forall \vec{\mathsf{x}}_*.\alpha)$, where $\vec{\mathsf{x}}_* = \langle \mathsf{x}_2, \dots, \mathsf{x}_n \rangle$.

We can extend the notation to existential quantifiers in the usual fashion, with $\exists\vec{\mathsf{x}}.\alpha$ standing for $\neg(\forall\vec{\mathsf{x}}.\neg\alpha)$. We also note that this notation allows for formulas in which the tuple $\vec{\mathsf{x}}$ contains duplicate variable symbols, something that is perfectly acceptable in first-order logic (and occasionally very useful, such as when one is working in the decidable 2-variable fragment of first-order logic). In fact, the following basic result shows that whether or not one allows such tuples is simply a matter of taste, and doesn't change the expressiveness of the notation:

Lemma 4. *Let $\vec{\mathsf{x}} \in \textsc{var}^n$ and $\vec{\mathsf{y}} \in \textsc{var}^m$ be two tuples, possible of different arities, containing exactly the same variable symbols, i.e. $\mathsf{z} = \mathsf{x}_i$ for some $1 \leq i \leq n$ iff $\mathsf{z} = \mathsf{y}_j$ for some $1 \leq j \leq m$. Then under both standard first-order semantics and Herbrand semantics, $\forall\vec{\mathsf{x}}.\alpha$ is logically equivalent to $\forall\vec{\mathsf{y}}.\alpha$ for every $\alpha \in \mathcal{L}$.*

Chapter 3

KLM-Style Defeasible Reasoning

3.1 Background

The KLM framework was first proposed as a qualitative system for defeasible reasoning by Kraus, Lehmann and Magidor [22, 25] in the 1990s, to whom the acronym KLM refers. It builds on top of propositional logic by adding an additional logical connective, which is denoted by "$\mathrel{|\!\sim}$" (pronounced *"twiddle"*). In our exposition, this connective expresses a relationship between two propositional formulas that is intended to be read along the lines of *"typically implies"* or *"defeasibly implies"*. This differs from Kraus et al. [22], in which the connective represents a *consequence relation*, and agrees rather with the perspective of Lehmann et al. [25].

For instance, we can use "$\mathrel{|\!\sim}$" to express statements like the following:

"$\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$"
"birds typically/defeasibly fly"

Note the difference here between the reading of "$\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$" and the reading of "$\mathsf{b} \rightarrow \mathsf{f}$", which states that birds *always* or *unconditionally* fly. As one might expect based on this reading, there is a fundamental difference between "$\mathrel{|\!\sim}$" and the standard logical connectives (namely "$\wedge$" "$\vee$", "$\rightarrow$" and "$\neg$").

These standard logical connectives are *truth functional*, which means that the truth value of a compound depends entirely on the truth values of its constituents. For instance, a compound such as "$\alpha \wedge \beta$" is true if and only if both "α" and "β" are true. On the other hand, "$\mathrel{|\!\sim}$" is *not* a truth functional connective. Intuitively, this is because the truth value of a statement such as *"birds*

typically fly" depends on *all* conceivable situations involving birds, not just the situation we happen to be considering.

A *defeasible formula* is defined to be anything of the form $\alpha \mathrel{|\!\sim} \beta$, where $\alpha, \beta \in \mathcal{L}$ are propositional formulas over a specified set $\mathcal{P}$ of atoms. The set of defeasible formulas over $\mathcal{P}$ is denoted $\mathcal{L}_{\mathcal{P}}^{|\sim}$ (we will drop the subscript where it is unambiguous or doesn't matter), and constitutes the set of valid formulas for propositional KLM. Note that this imposes some implicit constraints on the connective "$|\sim$":

1. No nesting of $|\sim$ is allowed - formulas such as $\alpha \mathrel{|\!\sim} (\beta \mathrel{|\!\sim} \gamma)$ are invalid.
2. Negations and conjunctions of formulas containing $|\sim$ are not permitted - formulas such as $\neg(\alpha \mathrel{|\!\sim} \beta)$ and $(\alpha \mathrel{|\!\sim} \beta) \vee (\alpha \mathrel{|\!\sim} \neg\beta)$ are invalid.

3.2 Rationality Postulates

Defeasible reasoning is often characterised in terms of properties that it *lacks* in general, such as monotonicity of entailment. One of the main selling points of the KLM framework is that it characterises several properties that defeasible reasoning *has*, known as the *rationality postulates*. These properties describe the ways in which new facts can be (defeasibly) derived from old facts. Or, phrased differently, they describe the structure of the *implicit* content of a defeasible theory [32].

Consider, for instance, a reasoner who knows the *explicit* facts that birds typically fly ($\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$), and that Tweety is a bird (b). Then the reasoner is justified in concluding the *implicit* fact that Tweety flies (f), at least until new facts are learned that require the revision of this conclusion.

To motivate the rationality postulates, let's consider a few examples of defeasible knowledge bases, which are defined to be sets of defeasible formulas $\mathcal{K} \subseteq \mathcal{L}^{|\sim}$. While this appears to exclude the possibility of classical formulas $\alpha \in \mathcal{L}$ being a part of a knowledge base, we note that in the KLM framework such a formula can be encoded by $\neg\alpha \mathrel{|\!\sim} \bot$ [25, p. 6]. The fact that this encoding is valid is non-trivial, and will be properly justified later. For now, we simply assume it is true to simplify our discussion.

Our first example, as is customary in AI literature, concerns birds:

Example 5 (The Penguin Triangle)**.** *Consider the set of propositional atoms $\mathcal{P} = \{\mathsf{b}, \mathsf{f}, \mathsf{p}\}$, which stand for* b*ird,* f*ly and* p*enguin. Then it is generally known that penguins are birds, that birds typically fly and that penguins typically don't. This can be represented by the knowledge base $\mathcal{K} = \{\mathsf{p} \to \mathsf{b}, \mathsf{b} \mathrel{|\!\sim} \mathsf{f}, \mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}\}$.*

Note that the classical version of the same knowledge base, namely $\Gamma = \{\mathsf{p} \to \mathsf{b}, \mathsf{b} \to \mathsf{f}, \mathsf{p} \to \neg\mathsf{f}\}$, implies the formula $\mathsf{p} \to \bot$, i.e. that there are no penguins!

Our second example concerns mathematicians, and is based off the well-known *Nixon diamond* knowledge base [25]:

Example 6 (The Mathematician Diamond)**.** *Consider the set of propositional atoms* $\mathcal{P} = \{\mathsf{m}, \mathsf{f}, \mathsf{e}, \mathsf{c}\}$*, which stand for* m*athematician,* f*ormal,* e*ccentric and* c*reative. Our claim is that mathematicians typically have the unusual combination of being both formal and eccentric. On the other hand, formal people are typically not very creative, while eccentric people typically are. This paradoxical state of affairs can be represented by the knowledge base* $\mathcal{K} = \{\mathsf{m} \mathrel{|\!\sim} \mathsf{e} \wedge \mathsf{f}, \mathsf{e} \mathrel{|\!\sim} \mathsf{c}, \mathsf{f} \mathrel{|\!\sim} \neg\mathsf{c}\}$.

In propositional logic, deductively closed theories provide a formal model of the outcome of a classical reasoning process, as they are closed under all of the classical inference rules. This is justified by Corollary 1, which connects this observation to the semantic definition of classical entailment.

In the same vein, we wish to introduce a formal model of the outcome of a *defeasible* reasoning process. Like its classical counterpart, this formal model should be closed under a number of *defeasible properties*, which describe permissible patterns of defeasible reasoning, or the structure of the implicit knowledge a reasoner has about its knowledge base. To make these notions precise, suppose that a set of atoms $\mathcal{P}$ has been fixed by the reasoner to represent propositions about some domain. Then we let $\tilde{\mathcal{P}}$ be a set of *meta-atoms*, disjoint from $\mathcal{P}$, which we will use to denote logical schemas over $\mathcal{P}$.

We denote the set of (meta-)propositional formulas over $\tilde{\mathcal{P}}$ by $\mathcal{L}^{\tilde{\mathcal{P}}}$, and define an *inclusion formula* to be anything of the form $\alpha \mathrel{|\!\sim} \beta$, $\alpha \not\mathrel{|\!\sim} \beta$, or $\models \alpha$, where $\alpha, \beta \in \mathcal{L}^{\tilde{\mathcal{P}}}$. The set of such inclusion formulas will be denoted by $\mathbb{I}^{\tilde{\mathcal{P}}}_{\mid\sim}$. An *instance* of an inclusion formula is the result of applying some substitution $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ to it, which assigns a propositional formula over $\mathcal{P}$ to each meta-atom. We denote such an instance by $\varphi(I)$, where $I \in \mathbb{I}^{\tilde{\mathcal{P}}}_{\mid\sim}$ is the inclusion formula.

Example 7. *Suppose that* $\mathcal{P} = \{\mathsf{p}, \mathsf{q}\}$*, and that our set of meta-atoms is given by* $\tilde{\mathcal{P}} = \{\alpha, \beta\}$*. Then a typical example of an inclusion formula is:*

$$I = \alpha \wedge \beta \not\mathrel{|\!\sim} \neg\alpha$$

Consider the substitutions $\varphi_1, \varphi_2 : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ *defined by* $\varphi_1(\alpha) = \mathsf{p}, \varphi_1(\beta) = \neg\mathsf{q}$*, and* $\varphi_2(\alpha) = \mathsf{p} \vee \mathsf{q}, \varphi_2(\beta) = \top$ *respectively. Then we have the instances:*

$$\varphi_1(I) = \mathsf{p} \wedge \neg\mathsf{q} \not\mathrel{|\!\sim} \neg\mathsf{p}$$
$$\varphi_2(I) = (\mathsf{p} \vee \mathsf{q}) \wedge \top \not\mathrel{|\!\sim} \neg(\mathsf{p} \vee \mathsf{q})$$

A *defeasible property* is defined to be anything of the form $I_1, \ldots, I_n \implies I_{n+1}$, where the $I_i \in \mathbb{I}^{\tilde{\mathcal{P}}}_{\mid\sim}$ are all inclusion formulas, and we denote the set of defeasible properties by $\mathbb{P}^{\tilde{\mathcal{P}}}_{\mid\sim}$. Intuitively, a defeasible property states that whenever a

reasoning process sanctions an instance of the inclusion formulas on the left-hand side, it should also sanction the corresponding instance of the right-hand formula.

Example 8. *Let's construct a defeasible property that expresses the* Transitivity *pattern of classical reasoning for defeasible formulas: if we reason that α implies β, and that β implies γ, then classically we would also be willing to reason that α implies γ. This can be formalised by:*

$$(\textsc{Trans}) \quad \alpha \mathrel{|\!\sim} \beta, \beta \mathrel{|\!\sim} \gamma \implies \alpha \mathrel{|\!\sim} \gamma$$

Note that we have left the definition of $\tilde{\mathcal{P}}$ implicit in this example (it contains α, β and γ). Context will always be sufficient to determine whether a symbol like "α" is playing the role of a meta-atom or a propositional formula.

Given a set $\mathcal{S} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$ of defeasible formulas, we say $\mathcal{S}$ *satisfies* an inclusion formula instance $\varphi(\alpha \mathrel{|\!\sim} \beta)$ iff $\varphi(\alpha \mathrel{|\!\sim} \beta) \in \mathcal{S}$. Similarly, $\mathcal{S}$ satisfies $\varphi(\alpha \not\mathrel{|\!\sim} \beta)$ iff $\varphi(\alpha \mathrel{|\!\sim} \beta) \notin \mathcal{S}$, and $\mathcal{S}$ satisfies $\varphi(\models \alpha)$ iff α is a classical tautology.

Definition 3. *Let $P = I_1, \ldots, I_n \implies I_{n+1}$ be a defeasible property, and suppose $\mathcal{S} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$ is a set of defeasible formulas. Then $\mathcal{S}$ is said to be* P-complete *iff for every substitution $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ such that $\mathcal{S}$ satisfies $\varphi(I_1), \ldots, \varphi(I_n)$, we have that $\mathcal{S}$ satisfies $\varphi(I_{n+1})$.*

In other words, a set of formulas is P-complete if it is closed under the property P. Similarly, given a set $D \subseteq \mathbb{P}^{\tilde{\mathcal{P}}}_{\mathrel{|\!\sim}}$ of defeasible properties, we say a set of formulas is D-complete iff it is P-complete for every $P \in D$. As we will see later, D-complete sets of formulas play the same role for us as complete theories play for classical logic, in the sense that they completely characterise a class of semantic structures for $\mathcal{L}^{\mathrel{|\!\sim}}$ [22, p. 31] - at least once we've specified the right set of defeasible properties!

Let us now turn to this latter task. What kinds of defeasible properties should we intuitively expect to hold for our hypothetical defeasible reasoning process? The simplest properties we will consider are the following, which in the literature are called *Reflexivity*, *Right Weakening* and *Left Logical Equivalence* respectively:

$$(\textsc{Refl}) \quad \alpha \mathrel{|\!\sim} \alpha$$
$$(\textsc{Lle}) \quad \alpha \mathrel{|\!\sim} \gamma, \models \alpha \leftrightarrow \beta \implies \beta \mathrel{|\!\sim} \gamma$$
$$(\textsc{Rw}) \quad \alpha \mathrel{|\!\sim} \beta, \models \beta \rightarrow \gamma \implies \alpha \mathrel{|\!\sim} \gamma$$

Reflexivity says that formulas typically imply themselves, a statement that is almost tautologically true. In Example 5, for instance, one would expect to conclude that penguins are typically penguins ($\mathsf{p} \mathrel{|\!\sim} \mathsf{p}$). Similarly, Left Logical Equivalence states that equivalent formulas typically imply the same things. If

we learn that *tori* (t) is simply another term for a bird ($\mathsf{b} \leftrightarrow \mathsf{t}$), then we would presumably claim that *tori* typically fly ($\mathsf{t} \mathrel{|\!\sim} \mathsf{f}$).

Right Weakening is a little more complex, but easy to justify intuitively. Given that non-flying things are always ground-based ($\neg\mathsf{f} \rightarrow \mathsf{g}$), for instance, we would conclude from Example 5 that penguins are typically ground-based ($\mathsf{p} \mathrel{|\!\sim} \mathsf{g}$).

A simple consequence of these properties is *Supraclassicality*, which states that anything which can be deduced classically can also be inferred defeasibly:

Lemma 5. *[26, p. 12] Suppose $\mathcal{S}$ is complete for* (Refl) *and* (Rw)*. Then $\mathcal{S}$ is also complete for the following property:*

$$\text{(Sc)} \quad \models \alpha \rightarrow \beta \implies \alpha \mathrel{|\!\sim} \beta$$

The next two properties relate to the way defeasibility interacts with conjunction and disjunction. The *Or* property states that if a formula $\gamma \in \mathcal{L}$ follows defeasible from two other formulas $\alpha, \beta \in \mathcal{L}$, then it should also follow defeasibly from their disjunction $\alpha \vee \beta$. The *And* property is simply the dual form of this, stating that if both β and γ follow defeasibly from α, then so should their conjunction $\beta \wedge \gamma$.

$$\text{(Or)} \quad \alpha \mathrel{|\!\sim} \gamma, \beta \mathrel{|\!\sim} \gamma \implies \alpha \vee \beta \mathrel{|\!\sim} \gamma$$
$$\text{(And)} \quad \alpha \mathrel{|\!\sim} \beta, \alpha \mathrel{|\!\sim} \gamma \implies \alpha \mathrel{|\!\sim} \beta \wedge \gamma$$

These properties describe ways in which defeasibility interacts with classical facts and connectives, and for the most part they are classically valid as well. On the other hand, there are certain classical inference properties that are definitely *not* defeasibly valid. One of these is *Transitivity*, a property we introduced earlier in this section. In Example 6, for instance, we know that mathematicians are typically eccentric ($\mathsf{m} \mathrel{|\!\sim} \mathsf{e}$) and that eccentric people are typically creative ($\mathsf{e} \mathrel{|\!\sim} \mathsf{c}$). Are we justified in concluding that mathematicians are typically creative ($\mathsf{m} \mathrel{|\!\sim} \mathsf{c}$)?

Going back to Example 6, this property would allow us to conclude that mathematicians are both typically creative *and* typically not creative! By supraclassicality, we would conclude that there are no mathematicians ($\mathsf{m} \mathrel{|\!\sim} \bot$), which is certainly not what we meant to express. In fact, it turns out that in the presence of the other properties we have mentioned, (Trans) implies *Monotonicity*:

Lemma 6. *[22, p. 15] Suppose $\mathcal{S}$ is complete for* (And)*,* (Rw)*,* (Or) *and* (Trans)*. Then $\mathcal{S}$ is complete for the following property:*

$$\text{(Mon)} \quad \alpha \mathrel{|\!\sim} \gamma \implies \alpha \wedge \beta \mathrel{|\!\sim} \gamma$$

This defeats the whole purpose of defeasible conclusions, which are supposed to be able to *change* when additional data comes to light! Rather than including properties like Transitivity and Monotonicity, the seminal papers on the Klm framework by Lehmann et al. [22, 25] consider the following weak forms

of Monotonicity, known as *Cautious Monotonicity* and *Rational Monotonicity* respectively:

$$(\textsc{Cm}) \quad \alpha \mathrel{|\!\sim} \beta, \alpha \mathrel{|\!\sim} \gamma \implies \alpha \wedge \beta \mathrel{|\!\sim} \gamma$$
$$(\textsc{Rm}) \quad \alpha \mathrel{|\!\sim} \gamma, \alpha \not\mathrel{|\!\sim} \neg\beta \implies \alpha \wedge \beta \mathrel{|\!\sim} \gamma$$

Cautious Monotonicity expresses the idea that if $\beta \in \mathcal{L}$ can be defeasibly concluded from $\alpha \in \mathcal{L}$, then learning that β is true should not *change* any of the defeasible consequences of α. After all, we expected β to be true in the first place!

Rational Monotonicity states that adding additional premises to a defeasible argument is always valid, unless there is a good reason to suspect it isn't - namely, that we defeasibly expect one of the premises to be false.

These properties form the cornerstone of the KLM framework, as they provide pragmatic ways to reason defeasibly from a set of assumptions. Other forms of Monotonicity have appeared in the literature, such as *Disjunctive Monotonicity* and *Negation Monotonicity* (see Kraus et al. [22]), but the above properties have received the majority of the attention. In part, this is because they enjoy a particularly nice mathematical semantics.

Definition 4. *A* preferential set $\mathcal{S} \subseteq \mathcal{L}^{\mid\sim}$ *is one that is complete for* (Refl), (Lle), (Rw), (And), (Or) *and* (Cm). *A preferential set is said to be* rational *if in addition it is complete for* (Rm).

The properties in Definition 4 have come to be known as the *rationality posulates* or *rationality properties* in the literature (see Booth et al. [3] for instance), and we denote this collection of properties by $\mathbb{R}_{\mid\sim}$. As noted by Kraus et al. [22], there is generally little to be said about sets of formulas that aren't complete for these properties. There, the notion of a *cumulative* and *loop-cumulative* sets are studied, which are both strictly weaker than preferential sets.

3.3 Preferential, Modular and Ranked Interpretations

Earlier, we claimed that D-complete sets are the defeasible analogue of deductively closed classical theories. We now turn to the goal of justifying this analogy by defining semantic structures that characterise preferential and rational sets, in much the same way that classical valuations characterise closed and complete theories (see Corollary 1). While there are many ways to do this [16], we will stick with the semantics given by Lehmann et al. [22, 25] as it has intuitive appeal. The basic idea is to think of a preferential or rational set as describing which possible worlds are *more likely* or *more typical* than which other possible worlds. To illustrate what we mean by this, consider the following example:

Example 9. *Consider the propositional language in Example 5, and suppose that $\mathcal{S}$ is a preferential set that contains the defeasible formulas* $\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$ *and* $\mathsf{b} \wedge \mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}$. *This states that while birds usually fly, birds that are penguins do* not *usually fly.*

Thinking in terms of possible worlds, consider the valuation $\mathsf{b}\overline{\mathsf{p}}\mathsf{f}$. *According to our preferential set $\mathcal{S}$ this would be considered a* typical *valuation, since when* b *is true $\mathcal{S}$ states that we are typically supposed to have* f *be true as well. For the same reason, the valuation* $\mathsf{b}\overline{\mathsf{p}\mathsf{f}}$ *would be considered* atypical*, as it doesn't conform to the intuition of either of our defeasible formulas.*

Somewhere between the two lies the valuation $\mathsf{b}\mathsf{p}\bar{\mathsf{f}}$. *While it fails to conform to the formula $b \mathrel{|\!\sim} f$, it* does *conform to the formula $b \wedge p \mathrel{|\!\sim} \neg f$.*

The point being made here is that a preferential or rational set provides some information about which possible worlds are more reasonable than others, depending on which defeasible formulas in the set they conform to. This suggests that one way to provide meaning to defeasible formulas is to use some kind of preference ranking over the set of valuations $\mathcal{U}$:

Definition 5. *[22, p. 27] A* preferential interpretation *is a triple $\mathcal{P} = \langle S, l, \prec \rangle$, where S is a set of states, $l : S \to \mathcal{U}$ is a function that labels each state with a valuation and $\prec \subseteq S \times S$ is a strict partial order over S satisfying the following smoothness condition:*

For any $\alpha \in \mathcal{L}$, the set $\widehat{\alpha} = \{s \in S : l(s) \Vdash \alpha\}$ has a minimal element.

The smoothness condition encodes the intuition that if there is *some* world satisfying a given proposition α, then there must be a *typical* world satisfying α. This requirement can be relaxed (see Friedman et al. [16] for a more general definition), but for our purposes the original definition of a preferential interpretation suffices. We also note that we allow different states $s, t \in S$ to have the same label, i.e. $l(s) = l(t)$. A preferential interpretation where every state has a unique label is said to be *injective*.

To complete the proposed semantics, we need to define what it means for a defeasible or classical formula to be *satisfied* by a preferential interpretation $\mathcal{P}$. This can be done quite naturally as follows, where $\min_{\prec} \widehat{\alpha}$ denotes the $\prec$-minimal elements of $\widehat{\alpha}$:

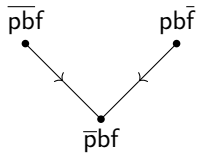
1. For a classical formula $\alpha \in \mathcal{L}$, $\mathcal{P} \Vdash \alpha$ iff $l(s) \Vdash \alpha$ for all $s \in S$.
2. For a defeasible formula $\alpha \mathrel{|\!\sim} \beta \in \mathcal{L}^{\mid\sim}$, $\mathcal{P} \Vdash \alpha \mathrel{|\!\sim} \beta$ iff $l(s) \Vdash \beta$ for all $l \in \min_{\prec} \widehat{\alpha}$.

Classical formulas are satisfied by a preferential interpretation whenever they are satisfied by all of the interpretation's states. In other words, classical formulas represent what is *"always"* the case in one of the interpretation's possible

worlds. A defeasible formula such as $\alpha \mathrel{|\!\sim} \beta$, on the other hand, is satisfied whenever β holds in the *most typical* worlds satisfying α. From this point of view, defeasible formulas represent what is *"usually"* or *"typically"* the case when certain premises are true.

If a preferential interpretation $\mathcal{P}$ satisfies a formula α, then we say $\mathcal{P}$ is a *model* of α. A preferential interpretation is a model of a knowledge base $\mathcal{K} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$ iff it satisfies every formula in the knowledge base.

Example 10. *Consider the knowledge base from Example 5,* $\mathcal{K} = \{\mathsf{p} \to \mathsf{b}, \mathsf{b} \mathrel{|\!\sim} \mathsf{f}, \mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}\}$*. We claim that the following diagram represents a preferential model of* $\mathcal{K}$*:*



Each node in the diagram represents a state, and the valuations written next to each node represent that state's label. The arrowed lines represent the preference ordering and thus in this example we have $\min_{\prec} \widehat{\mathsf{p}} = \{\mathsf{pb}\overline{\mathsf{f}}\}$ *and* $\min_{\prec} \widehat{\mathsf{b}} = \{\overline{\mathsf{p}}\mathsf{bf}\}$*. It's also easy to see that every state satisfies* $\mathsf{p} \to \mathsf{b}$*.*

Let us now prove our earlier statement that classical formulas are equivalent to certain defeasible formulas:

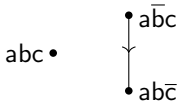
Lemma 7. *[25, p. 6] Let* $\mathcal{P} = \langle S, l, \prec \rangle$ *be a preferential interpretation. Then* $\mathcal{P} \Vdash \alpha$ *iff* $\mathcal{P} \Vdash \neg\alpha \mathrel{|\!\sim} \bot$*.*

Intuitively, this makes some sense. After all, $\alpha \mathrel{|\!\sim} \bot$ states that typical α's satisfy $\bot$, which implies that there cannot be any α's at all! Another consequence of preferential satisfaction, which provides some validation for our choice of semantics, is that preferential interpretations satisfy all of the preferential properties in Definition 4. In fact, preferential interpretations completely characterise preferential sets:

Lemma 8. *[22, p. 28] Denote the set of defeasible formulas satisfied by a preferential interpretation* $\mathcal{P} = \langle S, l, \prec \rangle$ *by* $\mathcal{S}_{\mathcal{P}} = \{\alpha \mathrel{|\!\sim} \beta \in \mathcal{L}^{\mathrel{|\!\sim}} : \mathcal{P} \Vdash \alpha \mathrel{|\!\sim} \beta\}$*. Then a set of formulas* $\mathcal{S} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$ *is a preferential set iff there exists some preferential interpretation* $\mathcal{P}$ *such that* $\mathcal{S} = \mathcal{S}_{\mathcal{P}}$*.*

A proof of Lemma 8 is fairly involved, and can be found in Kraus et al. [22]. Note, however, that not all preferential interpretations satisfy (RM):

Example 11. *Consider the following preferential interpretation* $\mathcal{P}$*:*



First, we note that $\mathcal{P} \Vdash \mathsf{a} \mathrel{|\!\sim} \mathsf{b}$. *On the other hand,* $\mathcal{P} \not\Vdash \mathsf{a} \wedge \mathsf{c} \mathrel{|\!\sim} \mathsf{b}$ *and* $\mathcal{P} \not\Vdash \mathsf{a} \mathrel{|\!\sim} \neg\mathsf{c}$. *Thus* $\mathcal{S}_\mathcal{P}$ *is not* (RM)*-complete.*

This implies that rational sets are a strict subset of preferential sets, and hence a natural question is whether they can be characterised by some well-chosen subset of preferential interpretations. As first shown by Lehmann et al. [25], it turns out that it's enough to require that the preference ordering $\prec$ be modular (see Definition 1).

This leads naturally to the following definition, which in the papers by Lehmann et al. is called a *ranked model* [25, p. 19]. We have used the term "modular interpretation" to avoid overloading the term "ranked" when we start moving to more expressive defeasible logics - see Britz et al. [5] for an example of the usage:

Definition 6. *[25, p. 19] A* modular interpretation *is a preferential interpretation* $\mathcal{M} = \langle S, l, \prec \rangle$ *for which the preference ordering* $\prec$ *is modular (see Definition 1).*

Modular interpretations satisfy *all* of the properties in Definition 4. In fact, like the preferential case, modular interpretations completely characterise rational sets:

Lemma 9. *[25, p. 20] A set of formulas* $\mathcal{S} \subseteq \mathcal{L}^{\mid\sim}$ *is rational iff there is some modular interpretation* $\mathcal{M}$ *such that* $\mathcal{S} = \mathcal{S}_\mathcal{M}$.

Lemmas 8 and 9 together prove that preferential and modular interpretations form an adequate mathematical semantics for two different kinds of defeasible reasoning - namely, those characterised by preferential and rational sets respectively. This gives us a rigorous basis for asserting that a particular defeasible formula is *true*, and with this under our belt we are free to move on to understanding defeasible entailment, or logical consequence.

However, there are some final remarks to make that will simplify our lives in the future. A corollary of Definition 1 is that we can describe a modular interpretation in terms of a ranking function $\mathrm{rk} : S \to \Omega$, which maps states to some totally ordered set Ω. As we will see, in many cases we can even take Ω to be $\mathbb{N}$, the set of natural numbers.

In view of these remarks, it is usually simpler to describe rational sets in terms of the following objects, at least when it is possible to do so:

Definition 7. *A* ranked interpretation $\mathcal{R}$ *is a function* $\mathcal{R} : \mathcal{U} \to \mathbb{N}^\infty$*, where* $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$*, that satisfies the following convexity condition:*

For all $u \in \mathcal{U}$ such that $\mathcal{R}(u) < \infty$, and all $i \in \mathbb{N}$ such that $0 \leq i < \mathcal{R}(u)$, there exists some $u' \in \mathcal{U}$ such that $\mathcal{R}(u') = i$.

Satisfaction for ranked interpretations is simply a translated version of satisfaction for modular and preferential interpretations, where $\mathcal{U}^{\mathcal{R}} = \{u \in \mathcal{U} : \mathcal{R}(u) < \infty\}$ is the set of *possible worlds* for $\mathcal{R}$, and $\widehat{\alpha} = \{u \in \mathcal{U}^{\mathcal{R}} : u \Vdash \alpha\}$ is the set of *possible worlds* for some formula $\alpha \in \mathcal{L}$:

1. For a classical formula $\alpha \in \mathcal{L}$, $\mathcal{R} \Vdash \alpha$ iff $u \Vdash \alpha$ for all $u \in \mathcal{U}^{\mathcal{R}}$.
2. For a defeasible formula $\alpha \mathrel{|\!\sim} \beta \in \mathcal{L}^{\mathrel{|\!\sim}}$, $\mathcal{R} \Vdash \alpha \mathrel{|\!\sim} \beta$ iff $u \Vdash \beta$ for all $u \in \min_{\mathcal{R}} \widehat{\alpha}$.

Let's look at an example of a ranked interpretation, and introduce some diagrammatic notation:

Example 12. *Consider the knowledge base from Example 6, $\mathcal{K} = \{\mathsf{m} \mathrel{|\!\sim} \mathsf{e} \wedge \mathsf{f}, \mathsf{e} \mathrel{|\!\sim} \mathsf{c}, \mathsf{f} \mathrel{|\!\sim} \neg\mathsf{c}\}$. Then the following ranked interpretation is a model of $\mathcal{K}$:*

1	mefc
0	$\overline{\mathsf{m}}\overline{\mathsf{e}}\mathsf{fc}$, $\overline{\mathsf{m}}\overline{\mathsf{e}}\overline{\mathsf{f}}\overline{\mathsf{c}}$

Each valuation $u \in \mathcal{U}$ is permitted to appear in the diagram at most once. The numbers beside each line denote the rank of that line's valuations - for instance, in this example, $\mathcal{R}(\mathsf{mefc}) = 1$. Valuations that do not appear in the diagram are assumed to have rank ∞.

As one would expect from the previous discussion, ranked interpretations generate rational sets. On the other hand, unlike modular interpretations, not all rational sets can be characterised by a ranked interpretation! In particular, a rational set only has a corresponding ranked interpretation if it has a corresponding modular interpretation with a well-founded preference ordering, and this is not always possible:

Lemma 10. *[25, p. 7] There is a rational set $\mathcal{S}$ such that for all modular $\mathcal{M} = \langle S, l, \prec \rangle$ with $\mathcal{S} = \mathcal{S}_{\mathcal{M}}$, $\prec$ is not well-founded.*

In fact, as discussed in Lehmann et al. [25, p. 36], it's an open problem to characterise exactly which rational sets have well-founded modular interpretations. Fortunately for us, however, there is a large class of rational sets for which corresponding ranked interpretations *do* exist:

Lemma 11. *[25, p. 36] Let $\mathcal{S}$ be a rational set over a finite set of propositional atoms $\mathcal{P}$. Then there exists some ranked interpretation $\mathcal{R}$ such that $\mathcal{S} = \mathcal{S}_{\mathcal{R}}$.*

While the infinite case is mathematically interesting, we generally only care about finite sets of atoms in practice. For this reason, we will stick to ranked interpretations as our semantic structure of choice for the rest of this thesis.

3.4 Defeasible Entailment

In this section we turn to the problem of *entailment*; given a set of defeasible and/or classical formulas, under what conditions should we conclude that another formula follows logically from them? Generally speaking, there are many possible answers to this question, in much the same way that there are many possible semantics for any given logic. In order to analyse this properly, let's fix some notation and define a *knowledge base* to be any set of formulas $\mathcal{K} \subseteq \mathcal{L} \cup \mathcal{L}^{\mid\sim}$. Note that a knowledge base represents any set of formulas, and unlike a satisfation set isn't required to represent any kind of interpretation or satisfy any properties.

A traditional account of logical consequence, due to Tarski [33], states that a formula α follows logically from a knowledge base $\mathcal{K}$ iff every *model* of $\mathcal{K}$ satisfies α. By "model", we mean some interpretation in a class of semantic structures that satisfies all the formulas of $\mathcal{K}$. In our case, this would be preferential or ranked interpretations (or others!), depending on what kind of semantics we were interested in.

This suggests that we begin our discussion by looking at the following Tarskian forms of entailment:

Definition 8. *A knowledge base $\mathcal{K}$* preferentially entails *a formula $\alpha \mid\sim \beta$, denoted $\mathcal{K} \mid\approx_{\mathcal{P}} \alpha \mid\sim \beta$, iff $\mathcal{P} \Vdash \alpha \mid\sim \beta$ for every preferential interpretation $\mathcal{P}$ satisfying $\mathcal{K}$. Similarly, $\mathcal{K}$* rank entails *$\alpha \mid\sim \beta$, denoted $\mathcal{K} \mid\approx_{\mathcal{R}} \alpha \mid\sim \beta$, iff $\mathcal{R} \Vdash \alpha \mid\sim \beta$ for every ranked interpretation $\mathcal{R}$ satisfying $\mathcal{K}$.*

To make this clearer, let's look at an example:

Example 13. *Consider the knowledge base $\mathcal{K} = \{\mathsf{r} \rightarrow \mathsf{b}, \mathsf{p} \rightarrow \mathsf{b}, \mathsf{b} \mid\sim \mathsf{f}, \mathsf{p} \mid\sim \neg\mathsf{f}\}$, which extends the knowledge base from Example 5 with an additional atom r, meaning $\mathsf{r}$obin. This knowledge base has a number of models, such as the ranked interpretations $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ below:*

1	$\mathsf{brp}\bar{\mathsf{f}}$
0	$\mathsf{br}\bar{\mathsf{p}}\mathsf{f}$

$\mathcal{R}_1$

1	$\mathsf{brp}\bar{\mathsf{f}}$
0	$\mathsf{b}\bar{\mathsf{r}}\bar{\mathsf{p}}\mathsf{f}$

$\mathcal{R}_2$

1	$\mathsf{b}\bar{\mathsf{r}}\bar{\mathsf{p}}\mathsf{f}$
0	$\mathsf{br}\bar{\mathsf{p}}\mathsf{f}$

$\mathcal{R}_3$

Note that $\mathcal{R}_1 \Vdash \mathsf{r} \mid\sim \mathsf{f}$ and $\mathcal{R}_2 \nVdash \mathsf{r} \mid\sim \mathsf{f}$, and hence $\mathcal{K}$ does not *rank entail $\mathsf{r} \mid\sim \mathsf{f}$ (i.e. that robins typically fly). On the other hand, it can be shown that every ranked interpretation satisfying $\mathcal{K}$ also satisfies the formula $\mathsf{b} \mid\sim \neg\mathsf{p}$, and thus $\mathcal{K} \mid\approx_{\mathcal{R}} \mathsf{b} \mid\sim \neg\mathsf{p}$ (i.e. that birds are typically not penguins).*

From this example, we can deduce a number of general properties of rank entailment (and from similar considerations, preferential entailment). Firstly, even though we knew nothing about atom r other than the fact that $\mathsf{r} \rightarrow \mathsf{b}$ holds, rank entailment does not allow us to conclude that r satisfies the typical properties of b. For instance, $\mathcal{K}$ satisfies $\mathsf{b} \mid\sim \mathsf{f}$ but does *not* satisfy $\mathsf{r} \mid\sim \mathsf{f}$. This is a serious limitation, and it arises because rank entailment is extremely

conservative. A conclusion is sanctioned only if it is true in *every possible extension* of the knowledge base to a rational set.

Secondly, despite being conservative, rank entailment does sanction *some* intuitive conclusions. In the example, the knowledge base contains the formulas $\mathsf{p} \rightarrow \mathsf{b}$ and $\mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}$. Since it also contains $\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$, this tells us that p must be an atypical instance of b. And indeed, the conclusion $b \mathrel{|\!\sim} \neg p$ is rank entailed by $\mathcal{K}$.

Preferential and rank entailment also satisfy some pleasant mathematical properties, which are shared by all Tarskian entailment relations:

Lemma 12. *Preferential and rank entailment satisfy the following properties, where $\mathrel{|\!\approx}_?$ is used as a placeholder for either of the two:*

$$\begin{aligned}
&(\textsc{Incl}) && \alpha \mathrel{|\!\sim} \beta \in \mathcal{K} \textit{ implies } \mathcal{K} \mathrel{|\!\approx}_? \alpha \mathrel{|\!\sim} \beta \\
&(\textsc{Cumu}) && \mathcal{K} \mathrel{|\!\approx}_? \alpha \mathrel{|\!\sim} \beta \textit{ and } \mathcal{K} \cup \{\alpha \mathrel{|\!\sim} \beta\} \mathrel{|\!\approx}_? \gamma \mathrel{|\!\sim} \delta \textit{ implies } \mathcal{K} \mathrel{|\!\approx}_? \gamma \mathrel{|\!\sim} \delta \\
&(\textsc{Mono}) && \mathcal{K} \mathrel{|\!\approx}_? \alpha \mathrel{|\!\sim} \beta \textit{ implies } \mathcal{K} \cup \{\gamma \mathrel{|\!\sim} \delta\} \mathrel{|\!\approx}_? \alpha \mathrel{|\!\sim} \beta
\end{aligned}$$

In the literature, these properties are known as *Inclusion*, *Cumulativity* and *Monotonicity* respectively. Inclusion is the simple statement that a knowledge base entails every formula that it contains. Cumulativity, like the defeasible properties of the same name, states that if a knowledge base entails some formula, then adding that formula into the knowledge base doesn't allow it to entail anything new. In other words, it states that entailment is *transitively closed*.

The last property is really where we start to see problems with rank and preferential entailment, which leads to some subtle considerations. Monotonicity states that if a knowledge base entails a formula, then any *extension* of the knowledge base will entail that formula too. But this is precisely what we were trying to avoid! Earlier, we went to great pains to ensure that the connective "$\mathrel{|\!\sim}$" was non-monotonic, in the sense that knowledge bases like $\mathcal{K} = \{\mathsf{a} \mathrel{|\!\sim} \mathsf{c}, \mathsf{a} \wedge \mathsf{b} \mathrel{|\!\sim} \neg\mathsf{c}\}$ have models. This knowledge base states that "a" typically implies "c", but that adding the premise "b" *changes* the conclusion to "$\neg\mathsf{c}$".

We appear to be in the paradoxical position that our logic is both monotonic *and* non-monotonic! The resolution to this paradox is to realise that we are talking about two different things here. Defeasible formulas involving "$\mathrel{|\!\sim}$" are on the object level, i.e they are simply abstract formulas representing a relationship between *propositions*. And indeed, the relationship that they represent is typically non-monotonic. Preferential and rank entailment, on the other hand, are on the meta level, i.e. they represent a relationship between *abstract defeasible formulas*.

The object level and meta level of a logic often interact - consider the case of the *Deduction Theorem* for classical logic, for instance, which states that a formula $\alpha \rightarrow \beta$ is a tautology iff α classically entails β. It's important to recognise, however, that these interactions are a special feature of the logic, and are *not* universally valid. The Deduction Theorem fails for fuzzy or probabilistic

logics, for instance.

The fact that the object-level relationship represented by "$\mathrel{|\!\sim}$" differs in some ways from the meta-level relationship represented by "$\mathrel{|\!\approx}_{\mathcal{P}}$" or "$\mathrel{|\!\approx}_{\mathcal{R}}$" is not a contradiction. At most, it suggests that we simply haven't found the right notion of entailment for defeasible formulas. And in fact, this isn't the only reason we should consider looking for alternatives. In their seminal paper, Lehmann et al. prove that rank and preferential entailment are *identical*:

Theorem 2. *[25, p. 23] Let $\mathcal{K}$ be any knowledge base. Then $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{P}} \alpha \mathrel{|\!\sim} \beta$ iff $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{R}} \alpha \mathrel{|\!\sim} \beta$.*

Thus, even though preferential and ranked interpretations characterise different kinds of sets of formulas, the entailment relations we have defined do not reflect these differences at all. To deal with these issues, Lehmann and Magidor argue that the correct model of entailment for defeasible formulas should be *non-Tarskian* [25, p. 41]. Rather than saying a formula follows logically from a knowledge base if it is true in *every* model of the knowledge base, they propose considering a privileged *subset* of models. In the context of non-monotonic reasoning, this approach to entailment is known as *default valuation consequence* [26] and the resulting entailment relation depends heavily on the criteria used to pick out the privileged models.

There are many ways to do this (see [9] for a more comprehensive discussion), but we will focus on an elegant solution due to Lehmann and Magidor that is based on the following principle of *Presumption of Typicality* [23]:

> *When it comes to entailment, a formula should be considered as typical as possible with respect to the constraints of the given knowledge base.*

Let's look at an example of how this principle would be applied in practice:

Example 14. *Consider the knowledge base $\mathcal{K} = \{\mathsf{r} \to \mathsf{b}, \mathsf{b} \mathrel{|\!\sim} \mathsf{f}\}$, which we can informally read as* "robins are birds, and birds usually fly"*. As discussed in Example 13, $\mathcal{K}$ does* not *entail $\mathsf{r} \mathrel{|\!\sim} \mathsf{f}$. On the other hand, since the only thing the knowledge base says about robins is that they are birds, the Presumption of Typicality states that we should go further and assume that robins are typical birds. Thus an entailment relation based on the Presumption of Typicality* should *sanction the conclusion $\mathsf{r} \mathrel{|\!\sim} \mathsf{f}$.*

While the principle of Presumption of Typicality is useful as an intuition pump, it isn't particularly easy to formalise. And indeed, providing a mathematical version of the principle was one of the key contributions of Lehmann and Magidor's seminal paper on the KLM framework [25]. Their formulation is defined in terms of a complicated partial ordering on the *rational extensions* of a given knowledge base, i.e. the set of all rational sets containing the knowledge base as a subset. This is mainly for the sake of generality, as they consider knowledge bases over a possibly infinite set of propositional atoms.

We will instead present a simpler characterisation due to Giordano et al. [19]. Their insight is to view the Presumption of Typicality as a statement about the ranked interpretations that model a given knowledge base $\mathcal{K}$. Since ranked interpretations assign a rank to every valuation, which represents how *typical* that valuation is taken to be, the idea is to rephrase the Presumption of Typicality as follows:

> *When it comes to entailment, a valuation should be ranked as low as possible with respect to the constraints of the given knowledge base.*

This leads naturally to the following relation over the set of ranked interpretations:

Definition 9. *[19, p. 8] Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be ranked interpretations. Then we say $\mathcal{R}_1$ is* more typical *than $\mathcal{R}_2$, denoted $\mathcal{R}_1 \leq_G \mathcal{R}_2$, iff $\mathcal{R}_1(u) \leq \mathcal{R}_2(u)$ for every $u \in \mathcal{U}$.*

In other words, a ranked interpretation is considered more typical if it gives lower ranks to every valuation; recall that a valuation has rank 0 if it is maximally typical, and rank ∞ if it is maximally atypical. This relation is reflexive, transitive and antisymmetric, and hence partially orders the set of ranked interpretations. This allows us to give a formal definition of the *priviliged* models we alluded to earlier, based on the Presumption of Typicality:

Definition 10. *[19, p. 11] Let $\mathcal{K}$ be a knowledge base, and Mod($\mathcal{K}$) its set of ranked models. Then a formula $\alpha \mathrel{|\!\sim} \beta$ is in the* rational closure *of $\mathcal{K}$, denoted $\mathcal{K} \mathrel{|\!\approx}_{RC} \alpha \mathrel{|\!\sim} \beta$, iff $\mathcal{R} \Vdash \alpha \mathrel{|\!\sim} \beta$ for every $\mathcal{R} \in \min_{\leq_G} Mod(\mathcal{K})$.*

Rational closure satisfies a number of mathematical properties that make it a much better candidate model for defeasible reasoning than preferential or rank entailment. For one, it is truly defeasible:

Example 15. *Let's consider our earlier example of $\mathcal{K} = \{\mathsf{r} \to \mathsf{b}, \mathsf{b} \mathrel{|\!\sim} \mathsf{f}\}$. We claim that the following ranked interpretation, which we will denote $\mathcal{R}$, is the unique $\leq_G$-minimal model of $\mathcal{K}$:*

1	$\bar{\mathsf{r}}\mathsf{b}\bar{\mathsf{f}}$, $\mathsf{r}\mathsf{b}\bar{\mathsf{f}}$
0	$\overline{\mathsf{r}\mathsf{b}\mathsf{f}}$, $\mathsf{r}\mathsf{b}\mathsf{f}$, $\bar{\mathsf{r}}\mathsf{b}\mathsf{f}$, $\overline{\mathsf{r}\mathsf{b}}\mathsf{f}$

Firstly, $\mathcal{R}$ satisfies all of the formulas in $\mathcal{K}$. Secondly, it is clear that the rank-0 valuations cannot be ranked any lower than they are in $\mathcal{R}$, as they are maximally typical already. Thus it remains to check that there are no models in which the rank-1 or rank-∞ valuations have lower rank than in $\mathcal{R}$.

The rank-∞ valuations are $\mathsf{r}\bar{\mathsf{b}}\mathsf{f}$ and $\mathsf{r}\overline{\mathsf{b}\mathsf{f}}$. Both violate the classical formula $\mathsf{r} \to \mathsf{f}$, however, and thus they must have rank ∞ in every model of $\mathcal{K}$. On the other

*hand, the rank-*1 *valuations both violate the defeasible formula* $\mathsf{b} \mathrel{|\!\sim} \mathsf{f}$*, and thus they cannot have rank* 0 *in any model of* $\mathcal{K}$*.*

We conclude that $\mathcal{K} \not\models_{RC} \mathsf{r} \mathrel{|\!\sim} \mathsf{f}$*. We leave it to the reader to check that* $\mathcal{K} \cup \{r \mathrel{|\!\sim} \neg f\}$ *has ranked models too, however, which implies that* $\mathcal{K} \cup \{r \mathrel{|\!\sim} \neg f\} \not\models_{RC} r \mathrel{|\!\sim} f$*. Adding new facts has* removed *conclusions from the rational closure!*

Rational closure also satisfies some of the basic properties we mentioned earlier:

Lemma 13. *[25, p. 33] The entailment relation* $\models_{RC}$ *satisfies* (Incl) *and* (Cumu)*.*

The fact that $\mathcal{K}$ had a unique minimal model in Example 15 may seem like a coincidence. Remarkably, it turns out that this is always the case for consistent knowledge bases, i.e. knowledge bases with at least one ranked model:

Lemma 14. *[19, p. 15] If* $\mathcal{K}$ *is a consistent knowledge base, then it has a unique* $\leq_G$*-minimal ranked model.*

In fact, this unique minimal model can be constructed in a canonical way by letting valuations sink as low as possible across all models of $\mathcal{K}$:

Lemma 15. *[19, p. 15] Let* $\mathcal{K}$ *be a consistent knowledge base, and consider the ranked interpretation* $\mathcal{R}$ *given by the following construction:*

$$\mathcal{R}(u) = \min\{\mathcal{R}'(u) : \mathcal{R}' \text{ is a ranked interpretation satisfying } \mathcal{K}\}$$

Then $\mathcal{R}$ *satisfies* $\mathcal{K}$*, and is the unique* $\leq_G$*-minimal ranked model of* $\mathcal{K}$*.*

The immediate consequence of this lemma is the fact that the rational closure of a knowledge base can be completely characterised by a single ranked interpretation. This is known in the literature as the *Single Model Property*, and is one of the things that differentiates propositional Klm-style defeasibility from defeasibility in more expressive logics:

(Smp) for all $\mathcal{K}$ there exists some $\mathcal{R}_\mathcal{K}$ such that $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \beta$ iff $\mathcal{R}_\mathcal{K} \Vdash \alpha \mathrel{|\!\sim} \beta$

In conjunction with Lemma 11, this has the interesting consequence that rational closure satisfies "lifted" versions of all of the defeasible properties from Definition 4:

(Refl) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \alpha$

(Lle) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$ and $\models \alpha \leftrightarrow \beta$ implies $\mathcal{K} \models_{RC} \beta \mathrel{|\!\sim} \gamma$

(Rw) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \beta$ and $\models \beta \rightarrow \gamma$ implies $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$

(Or) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$ and $\mathcal{K} \models_{RC} \beta \mathrel{|\!\sim} \gamma$ implies $\mathcal{K} \models_{RC} \alpha \vee \beta \mathrel{|\!\sim} \gamma$

(And) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \beta$ and $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$ implies $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \beta \wedge \gamma$

(Cm) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \beta$ and $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$ implies $\mathcal{K} \models_{RC} \alpha \wedge \beta \mathrel{|\!\sim} \gamma$

(Rm) $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \gamma$ implies $\mathcal{K} \models_{RC} \alpha \wedge \beta \mathrel{|\!\sim} \gamma$ or $\mathcal{K} \models_{RC} \alpha \mathrel{|\!\sim} \neg\beta$

Because it satisfies many pleasant mathematical properties, and because it is based on an intuitive principle, rational closure is often viewed as the "gold standard" against which other defeasible entailment relations can be compared [25, 23]. Indeed, some suggest that any reasonable form of defeasible entailment for KLM-style systems should extend rational closure inferentially [9]. For these reasons, we will use the construction of an appropriate analogue of rational closure as our common goal when it comes to defeasible reasoning for more expressive logics such as Datalog.

3.5 Computing Rational Closure

Alongside its mathematical properties, rational closure also has the desirable property of being easy to compute. In this section we will sketch an efficient algorithm for reducing rational closure entailment checking to classical entailment checking, based on a notion of *typicality rank* for formulas. Our presentation follows that of Morris et al. [28], which is based on the work of Casini et al. [9].

The first concept we will need is that of an *exceptional formula*. A formula α is considered to be exceptional with respect to a knowledge base $\mathcal{K}$ if, roughly speaking, α being considered typical is inconsistent with the defeasible formulas in $\mathcal{K}$:

Definition 11. *[25, p. 11] A formula $\alpha \in \mathcal{L}$ is said to be* exceptional *with respect to a knowledge base $\mathcal{K} \subseteq \mathcal{L}^{\sim}$ iff $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \top \mathrel{|\!\sim} \neg\alpha$.*

Whilst exceptionality is defined in terms of rank entailment, it can easily be characterised in terms of classical entailment:

Lemma 16. *[25, p. 38] A formula $\alpha \in \mathcal{L}$ is exceptional with respect to $\mathcal{K} \subseteq \mathcal{L}^{\sim}$ iff $\mathcal{K}^{\rightarrow} \models \neg\alpha$, where $\mathcal{K}^{\rightarrow}$ is the* materialisation *of $\mathcal{K}$, defined as follows:*

$$\mathcal{K}^{\rightarrow} = \{\alpha \rightarrow \beta : \alpha \mathrel{|\!\sim} \beta \in \mathcal{K}\}$$

Similarly, a defeasible formula $\alpha \mathrel{|\!\sim} \beta$ is said to be exceptional with respect to $\mathcal{K}$ iff α is exceptional with respect to $\mathcal{K}$. We will call a classical or defeasible formula that is not exceptional with respect to $\mathcal{K}$ *typical* instead.

In terms of exceptionality, knowledge bases comes in two flavours. Firstly, it's possible that every formula in a knowledge base $\mathcal{K}$ is exceptional with respect to $\mathcal{K}$. In this case, $\mathcal{K}$ is called *completely exceptional* [25, p. 11]:

Example 16. *Consider the knowledge base $\mathcal{K} = \{\mathsf{a} \mathrel{|\!\sim} \neg\mathsf{a}\}$. If $\mathcal{R}$ is a ranked interpretation satisfying $\mathcal{K}$, then $\widehat{\mathsf{a}}$ must be empty, as every valuation in $\widehat{\mathsf{a}}$ would have to satisfy both a and $\neg\mathsf{a}$, which is impossible. Thus $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \top \mathrel{|\!\sim} \neg\alpha$, and we conclude that $\mathcal{K}$ is completely exceptional.*

Secondly, it's possible that some formulas in the knowledge base are exceptional, and some are typical:

Example 17. *Consider the knowledge base from Example 5,* $\mathcal{K} = \{\mathsf{p} \to \mathsf{b}, \mathsf{b} \mathrel{|\!\sim} \mathsf{f}, \mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}\}$*. It can be checked that* $\mathcal{K}^{\to} \models \neg\mathsf{p}$*, and hence penguins are exceptional with respect to* $\mathcal{K}$*. Intuitively, this is because penguins typically don't fly, and thus they form an atypical subcategory of bird. On the other hand,* $\mathcal{K}^{\to} \not\models \neg\mathsf{b}$*, and hence birds are typical with respect to* $\mathcal{K}$*.*

Here we mention a subtlety that is easily overlooked. By the convention we established earlier, the classical formula $\mathsf{p} \to \mathsf{b}$ *in* $\mathcal{K}$ *actually represents the defeasible formula* $\neg(\mathsf{p} \to \mathsf{b}) \mathrel{|\!\sim} \bot$*. Since* $\mathcal{K}^{\to} \models \mathsf{p} \to \mathsf{b}$*, we therefore conclude that the classical formula* $\mathsf{p} \to \mathsf{b}$ *is exceptional with respect to* $\mathcal{K}$ *as well. In fact, a similar argument shows that* all *classical formulas are exceptional with respect to any knowledge base containing them!*

By iteratively removing the typical formulas from a knowledge base until a completely exceptional knowledge base remains, we can split the knowledge base up into layers, each of which is successively more exceptional than the last.

Example 18. *Consider the knowledge base from the previous example again. From our discussion, it is clear that the first layer would be* $\{\mathsf{b} \mathrel{|\!\sim} \mathsf{f}\}$*, the second layer would be* $\{\mathsf{p} \mathrel{|\!\sim} \neg\mathsf{f}\}$*, and the final, completely exceptional layer would just contain the classical formula* $\{\mathsf{p} \to \mathsf{b}\}$*.*

BaseRank, defined in Algorithm 1, performs this splitting of knowledge bases using the classical characterisation of exceptional formulas [28, p. 3]:

Algorithm 1: BaseRank

Input: a finite knowledge base $\mathcal{K} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$
Output: a partition $(\mathcal{K}_0, \ldots, \mathcal{K}_n, \mathcal{K}_\infty)$ of $\mathcal{K}$
$i := 0$;
$E_0 := \mathcal{K}^{\to}$;
repeat
  $E_{i+1} := \{\alpha \to \beta \in E_i : E_i \models \neg\alpha\}$;
  $\mathcal{K}_i := \{\alpha \mathrel{|\!\sim} \beta : \alpha \to \beta \in E_i \setminus E_{i+1}$;
  $i := i + 1$;
until $E_{i+1} = E_i$;
$\mathcal{K}_\infty := E_i$;
if $E_i = \emptyset$ **then**
  $n := i - 2$;
else
  $n := i - 1$;
return $(\mathcal{K}_0, \ldots, \mathcal{K}_n, \mathcal{K}_\infty)$;

The name BaseRank comes from the fact that this partition can be used to assign a rank to every formula of $\mathcal{L}$, called its *base rank*:

Definition 12. *[9, p. 186] Let* $\mathcal{K} \subseteq \mathcal{L}^{\mathrel{|\!\sim}}$ *be a knowledge base, and suppose that* $(\mathcal{K}_0, \ldots, \mathcal{K}_n, \mathcal{K}_\infty) = \mathsf{BaseRank}(\mathcal{K})$*. Then the base rank of a formula* $\alpha \in \mathcal{L}$ *with*

respect to $\mathcal{K}$, denoted $br(\alpha)$, is defined to be the minimum $N \in \mathbb{N}^\infty$ such that α is typical with respect to $\bigcup_{N \leq i \leq \infty} \mathcal{K}_i$.

Remarkably, it turns out that there is a close connection between the base rank of formulas with respect to $\mathcal{K}$, and the rational closure of $\mathcal{K}$:

Theorem 3. *[19] Let $\mathcal{K} \subseteq \mathcal{L}^{\sim}$ be a knowledge base. Then $\mathcal{K} \mid\!\approx_{RC} \alpha \mid\sim \beta$ iff $br(\alpha) \leq br(\alpha \wedge \neg\beta)$ or $br(\alpha) = \infty$.*

This leads to the following algorithm for rational closure entailment checking[28, p. 3]:

Algorithm 2: RationalClosure

Input: a finite knowledge base $\mathcal{K} \subseteq \mathcal{L}^{\sim}$ and formula $\alpha \mid\sim \beta \in \mathcal{L}^{\sim}$
Output: **true** if $\mathcal{K} \mid\!\approx_{RC} \alpha \mid\sim \beta$, else **false**
$i := 0$;
$R := \mathcal{K}^{\rightarrow}$;
$(\mathcal{K}_0, \ldots, \mathcal{K}_n, \mathcal{K}_\infty) :=$ BaseRank($\mathcal{K}$);
while $R \models \neg\alpha$ ***and*** $R \neq \mathcal{K}_\infty{}^{\rightarrow}$ **do**
  $R := R \setminus K_i{}^{\rightarrow}$;
  $i := i + 1$;
end
return $R \models \alpha \rightarrow \beta$;

Theorem 4. *[9, p. 186] For any knowledge base $\mathcal{K} \subseteq \mathcal{L}^{\sim}$ and formula $\alpha \mid\sim \beta \in \mathcal{L}$, $\mathcal{K} \mid\!\approx_{RC} \alpha \mid\sim \beta$ iff RationalClosure$(\mathcal{K}, \alpha \mid\sim \beta) =$ **true**.*

Finally, we note that BaseRank and RationalClosure reduce rational closure entailment checking to a number of classical entailment checks that is polynomial in the size of the knowledge base. Similarly, by encoding classical formulas α as their defeasible equivalents $\neg\alpha \mid\sim \bot$, we can use RationalClosure to perform classical inference checks as well. Thus the rational closure entailment checking problem, like the propositional satisfaction problem, is NP-Complete [25, p. 39].

Chapter 4

Description Logics

4.1 Background

Description Logics are a family of knowledge representation languages that are generally more expressive than propositional logic, while at the same time being computationally feasible. They provide a logical formalism for reasoning about domain ontologies, and amongst other things are used as a basis for knowledge representation in the Semantic Web. A thorough exposition of the history of DLs, along with their relationships to other knowledge representation formalisms, can be found in the *Description Logic Handbook* [1].

At a mathematical level, Description Logics (DLs) are decidable fragments of first-order logic that allow for the definition of *concepts*, *roles* and *individuals*, as well as relationships between these three primitives [1, p. 49]. A concept represents a property that an individual may or may not have, such as being a $\mathsf{Student}$. A role represents a relationship between two individuals, such as $\mathsf{employedBy}$, and an individual simply represents an element of the domain being described.

Members of the DL family largely differ in the kinds of *concept constructors* and *role constructors* they allow. These constructors, like propositional connectives, allow for the expression of complex concepts and roles in terms of a basic set of *atoms*. Since we are more interested in the defeasible aspects of DLs than in the nuances of varous language constructions, we will focus on a particularly simple DL called ALC in this chapter.

The language of ALC is built up from a finite set $\mathcal{C}$ of *concept names*, a finite set $\mathcal{R}$ of *role names* and a finite set $\mathcal{I}$ of *individual names* or *attributes*. To continue our examples involving birds, consider the following hypothetical language:

Example 19. *Let* $\mathcal{C} = \{\mathsf{Bird}, \mathsf{Pigeon}, \mathsf{Eagle}, \mathsf{Predator}, \mathsf{Prey}\}$*,* $\mathcal{R} = \{\mathsf{eats}\}$*, and* $\mathcal{I} = \{\mathsf{a}\}$*. We will stick to the convention of capitalising concept names, while*

keeping role and individual names in lowercase.

ALC defines a number of concept constructors which allow for the expression of *complex concepts*. These are $\top$ (everything), $\bot$ (nothing), $\neg$ (complement), $\sqcap$ (conjunction), $\sqcup$ (disjunction), $\forall$ (value restriction) and $\exists$ (existential restriction), with their grammar given by the following rules:

1. $\top$ and $\bot$ are atomic complex concepts.
2. If C is a concept name, then C is an atomic complex concept.
3. If C is a complex concept, then $\neg C$ is a complex concept.
4. If C, D are complex concepts, then $C \sqcap D$ and $C \sqcup D$ are complex concepts.
5. If C is a complex concept and r is a role name, then $\forall \mathsf{r}.C$ and $\exists \mathsf{r}.C$ are complex concepts.

The set of all complex concepts definable with these rules constitutes the *language* of ALC, which we denote by $\mathcal{L}^{\text{ALC}}$.

Example 20. *Let $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{I}$ be as they were defined in Example 19. Some examples of complex concepts over these names, along with their intuitive interpretations, are:*

$$\text{“}\mathsf{Eagle} \sqcap \forall \mathsf{eats}.\mathsf{Pigeon}\text{”}$$
“eagle that only eats pigeons”

$$\text{“}\mathsf{Bird} \sqcup \neg\exists \mathsf{eats}.\mathsf{Predators}\text{”}$$
“either a bird or something that doesn’t eat any predators”

To express relationships between (complex) concepts, roles and individuals in ALC, we use *subsumption statements* and *assertion statements*. These are defined by the following rules:

1. If C and D are complex concepts, then $C \sqsubseteq D$ is a subsumption statement.
2. If C is a complex concept and a is an individual name, then $\mathsf{a} : C$ is an assertion statement.
3. If r is a role name and a, b are individual names, then $(\mathsf{a}, \mathsf{b}) : \mathsf{r}$ is an assertion statement.

We denote the set of subsumption statements by $\mathcal{L}_{\mathcal{T}}^{\text{ALC}}$, and the set of assertion statements by $\mathcal{L}_{\mathcal{A}}^{\text{ALC}}$.

Example 21. *Again, let $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{I}$ be as defined in Example 19. Then consider the following examples of subsumption and assertion statements:*

"$\mathsf{Pigeons} \sqsubseteq \mathsf{Bird} \sqcap \neg\exists \mathsf{eats}.\mathsf{Bird}$"
"pigeons are birds that don't eat any other birds"

"$\exists \mathsf{eats}.\mathsf{Bird} \sqsubseteq \mathsf{Predator}$"
"things that eat birds are predators"

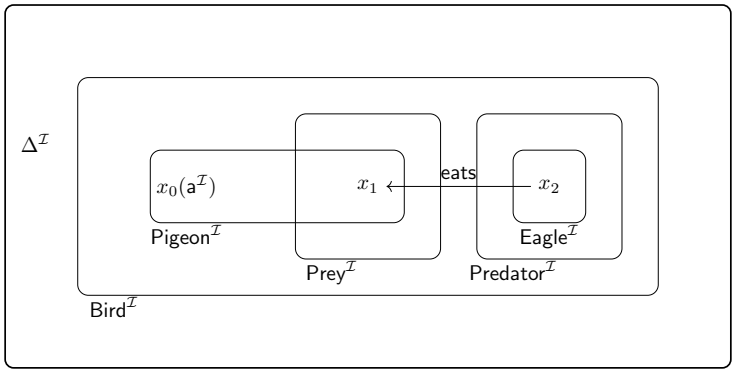
"$\mathsf{a} : \mathsf{Bird} \sqcap \neg \mathsf{Prey}$"
"a *is a bird that isn't preyed on*"

An ALC knowledge base (or *ontology*) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \subseteq \mathcal{L}_{\mathcal{T}}^{\text{ALC}}$ is a set of subsumption statements and $\mathcal{A} \subseteq \mathcal{L}_{\mathcal{A}}^{\text{ALC}}$ is a set of assertion statements. In the DL literature, $\mathcal{T}$ is usually called the *T-Box* (for "terminological box"), and $\mathcal{A}$ is usually called the *A-Box* (for "assertional box").

The semantics for DL knowledge bases is a variant of standard first-order semantics. An *interpretation* is defined to be a tuple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain*, and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps individual names, concept names and role names to elements, subsets and binary relations over the domain respectively. In other words:

1. If a is an individual name, then $\mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
2. If C is a concept name, then $\mathsf{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$.
3. If r is a role name, then $\mathsf{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Example 22. *The following diagram represents an interpretation:*



The domain consists of the elements $\Delta^{\mathcal{I}} = \{x_0, x_1, x_2\}$. *The interpretation of individual names is indicated in brackets next to an element, so in this example* $\mathsf{a}^{\mathcal{I}} = x_0$. *Concept names are interpreted by boxes, so for instance we have* $\mathsf{Pigeon}^{\mathcal{I}} = \{x_0, x_1\}$ *and* $\mathsf{Eagle}^{\mathcal{I}} = \{x_2\}$. *Finally, role names are interpreted by solid arrows, so we have* $\mathsf{eats}^{\mathcal{I}} = \{(x2, x1)\}$.

The interpretation function $\cdot^{\mathcal{I}}$ is extended to the set of complex concepts as follows, where $\mathsf{r}^{\mathcal{I}}(x) = \{y \in \Delta^{\mathcal{I}} : (x, y) \in \mathsf{r}^{\mathcal{I}}\}$ denotes the set of elements related to x by the role r:

1. $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$.
2. $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$.
3. $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$.
4. $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$.
5. $(\forall \mathsf{r}.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : \mathsf{r}^{\mathcal{I}}(x) \subseteq C^{\mathcal{I}}\}$.
6. $(\exists \mathsf{r}.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : \mathsf{r}^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\}$.

The interpretation $\mathcal{I}$ satisfies a subsumption statement $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Similarly, it satisfies an assertion statement $\mathsf{a} : C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and an assertion statement $(\mathsf{a}, \mathsf{b}) : \mathsf{r}$ iff $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in \mathsf{r}^{\mathcal{I}}$. As usual, the satisfaction relation is denoted by $\Vdash$. That is, for any kind of statement α we write $\mathcal{I} \Vdash \alpha$ to mean that $\mathcal{I}$ satisfies α. If $\mathcal{I}$ satisfies every statement in an T-Box $\mathcal{T}$ (or A-Box $\mathcal{A}$), then we say $\mathcal{I}$ is a *model* of $\mathcal{T}$ (or $\mathcal{A}$). Finally, given a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, we say $\mathcal{I}$ is a model of $\mathcal{K}$ iff it is a model of both $\mathcal{T}$ and $\mathcal{A}$.

Example 23. *In the interpretation in Example 22, we have that* $\mathsf{Pigeon}^{\mathcal{I}} = \{x_0, x_1\}$, $\mathsf{Bird}^{\mathcal{I}} = \{x_0, x_1, x_2\}$ *and* $\exists \mathsf{eats}.\mathsf{Bird}^{\mathcal{I}} = \{x_2\}$. *Thus by the above rules we conclude that* $(\mathsf{Bird} \sqcap \neg \exists \mathsf{eats}.\mathsf{Bird})^{\mathcal{I}} = \{x_0, x_1\}$, *and hence the interpretation satisfies the subsumption statement* $\mathsf{Pigeon} \sqsubseteq \mathsf{Bird} \sqcap \neg \exists \mathsf{eats}.\mathsf{Bird}$.

Given an ALC knowledge base $\mathcal{K}$ and a subsumption or assertion statement α, we say $\mathcal{K}$ *classically entails* α (denoted $\mathcal{K} \models \alpha$) iff every model of $\mathcal{K}$ satisfies α. If α is satisfied by every possible interpretation then we say it is a *classical tautology* (denoted $\models \alpha$). While we won't go into the details here, part of the appeal of Description Logics is that the classical entailment problem is decidable for them. While classical entailment checking for ALC is EXPTIME-complete in general (see Donini et al. [15] for an explicit algorithm), optimised algorithms exist that are significantly more efficient for real-world knowledge bases.

This concludes our brief overview of classical Description Logics. There is a wealth of literature on the topic, and interested readers are urged to consult the Description Logic Handbook [1] a comprehensive resource for classical DL theory. For the rest of the chapter, our focus will be on the problem of defeasible entailment for ALC.

Before we describe the KLM formalism for defeasibility in ALC, it's worth making a few remarks about what we should expect. First of all, ALC is a significantly more expressive language than propositional logic, as it supports formulas involving both unary predicates (concepts) and binary predicates (roles), as well

as limited forms of existential and universal quantification (existential and value restriction). It also supports statements about the properties that *particular* individuals and pairs of individuals satisfy (assertional statements). The price we pay for this is that defeasibility becomes a little harder to pin down, and defeasible entailment relations start to become a little less structured mathematically.

For this reason, we will approach defeasibility for ALC in two stages. First, we will look at what happens if we *only* consider T-Boxes, i.e. subsumption statements. It turns out that defeasibility for T-Boxes closely tracks the propositional case, and we will get almost exact analogues of the results from Chapter 3. Second, we will consider defeasibility in the presence of both T-Boxes *and* A-Boxes. Here, the addition of assertional statements starts to interfere with the *Single Model Property* for defeasible entailment relations, and we will show that several other properties have to be given up too.

4.2 Defeasibility for the T-Box

Given the similarities between a subsumption statement and a propositional implication, subsumption is a natural starting point for the study of defeasibility in DLs. Defeasible ALC extends classical ALC with an additional kind of subsumption statement, called a *defeasible subsumption statement*, which is defined by the following rule:

1. If C and D are complex concepts, then $C \mathrel{\underset{\sim}{\sqsubset}} D$ is a defeasible subsumption statement.

Similarly to the propositional "$\mathrel{|\!\sim}$", the defeasible subsumption connective "$\mathrel{\underset{\sim}{\sqsubset}}$" is intended to be read along the lines of "*typically subsumes*" or "*defeasibly subsumes*". For instance, we can express statements such as the following:

$$\text{“}\mathsf{Penguin} \mathrel{\underset{\sim}{\sqsubset}} \neg\mathsf{Fly}\text{”}$$
“penguins typically don't fly”

We denote the set of defeasible subsumption statements by $\mathcal{L}_{\mathcal{D}}^{\text{ALC}}$. A *defeasible T-Box* is defined to be a set of defeasible subsumption statements $\mathcal{D} \subseteq \mathcal{L}_{\mathcal{D}}^{\text{ALC}}$, and a *defeasible knowledge base* (or *defeasible ontology*) is a triple $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}, \mathcal{A} \rangle$, where $\mathcal{T}$, $\mathcal{D}$ and $\mathcal{A}$ are a T-Box, defeasible T-Box and A-Box respectively.

In this section we will focus purely on T-Boxes, and consider defeasible knowledge bases where $\mathcal{A} = \emptyset$, i.e. there are no assertional statements. Like the propositional case (see Lemma 7), we note that in defeasible ALC the classical subsumption statement $C \sqsubseteq D$ is equivalent to the defeasible subsumption statement $C \sqcap \neg D \mathrel{\underset{\sim}{\sqsubset}} \bot$ [5, p. 10]. This means that every model of the classical statement is also a model of the defeasible statement and vice-versa, a fact that can be checked once we introduce ranked interpretations. For now, we will take

it on faith, and note that we can therefore consider the case where $\mathcal{T} = \emptyset$ and we *only* have defeasible statements.

The analysis of defeasible inference for ALC can be carried out in the same way as for propositional logic (see Section 3.2), with concepts taking the place of propositional atoms. Supposing that a set $\mathcal{C}$ of concept names and a set $\mathcal{R}$ of role names have been fixed, we let $\tilde{\mathcal{C}}$ be a set of *meta-concept names*, disjoint from $\mathcal{C} \cup \mathcal{R}$. These meta-concepts will be used to denote logical schemas over the language of ALC, which we will use to express *defeasible properties* for ALC knowledge bases, i.e. permissible patterns of defeasible reasoning.

An *inclusion formula* over $\tilde{\mathcal{C}}$ is defined to be anything of the form $C \mathrel{\sqsubset\!\!\sim} D$, $C \not\mathrel{\sqsubset\!\!\sim} D$, $C \sqsubseteq D$ or $C \equiv D$, where $C, D \in \tilde{\mathcal{C}}$ are meta-concept names. An *instance* of an inclusion formula I is the result of applying a substitution $\varphi : \tilde{\mathcal{C}} \to \mathcal{L}^{\text{ALC}}$, which maps each meta-concept name to a complex concept over $\mathcal{C}$ and $\mathcal{R}$. We denote such an instance by $\varphi(I)$.

Given a set $\mathcal{S} \subseteq \mathcal{L}^{\text{ALC}}_{\mathcal{D}}$ of defeasible subsumption statements, we say $\mathcal{S}$ *satisfies* an instance of an inclusion formula $\varphi(C \mathrel{\sqsubset\!\!\sim} D)$ or $\varphi(C \not\mathrel{\sqsubset\!\!\sim} D)$ iff $\varphi(C \mathrel{\sqsubset\!\!\sim} D) \in \mathcal{S}$ or $\varphi(C \mathrel{\sqsubset\!\!\sim} D) \notin \mathcal{S}$ respectively. Similarly, $\mathcal{S}$ satisfies $\varphi(C \sqsubseteq D)$ iff $\varphi(C \sqsubseteq D)$ is a classical tautology, and $\mathcal{S}$ satisfies $\varphi(C \equiv D)$ iff $\mathcal{S}$ satisfies both $\varphi(C \sqsubseteq D)$ and $\varphi(D \sqsubseteq C)$.

A *defeasible property* is defined to be anything of the form $I_1, \ldots, I_n \implies I_{n+1}$, where the I_i are all inclusion formulas. As in the propositional case, a defeasible property intuitively states that if a defeasible reasoning process sanctions the formulas $I_1, \ldots, I_n$, then it should also sanction the formula I_{n+1}. Formally, this is defined as follows:

Definition 13. *Let $P = I_1, \ldots, I_n \implies I_{n+1}$ be a defeasible property, and suppose $\mathcal{S} \subseteq \mathcal{L}^{\text{ALC}}_{\mathcal{D}}$ is a set of defeasible subsumption statements. Then $\mathcal{S}$ is said to be* P-complete *iff for every substitution $\varphi : \tilde{\mathcal{C}} \to \mathcal{L}^{\text{ALC}}$ such that $\mathcal{S}$ satisfies $\varphi(I_1), \ldots, \varphi(I_n)$, we have that $\mathcal{S}$ satisfies $\varphi(I_{n+1})$.*

Similarly, given a set D of defeasible properties, we say a set $\mathcal{S}$ is D-complete iff $\mathcal{S}$ is P-complete for every $P \in D$. With these definitions under our belt, the propositional rationality postulates can be translated into ALC as follows:

$$
\begin{aligned}
\text{(REFL)}\quad & C \mathrel{\sqsubset\!\!\sim} C \\
\text{(LLE)}\quad & C \equiv D, C \mathrel{\sqsubset\!\!\sim} E \implies D \mathrel{\sqsubset\!\!\sim} E \\
\text{(RW)}\quad & D \sqsubseteq E, C \mathrel{\sqsubset\!\!\sim} D \implies C \mathrel{\sqsubset\!\!\sim} E \\
\text{(OR)}\quad & C \mathrel{\sqsubset\!\!\sim} E, D \mathrel{\sqsubset\!\!\sim} E \implies C \sqcup D \mathrel{\sqsubset\!\!\sim} E \\
\text{(AND)}\quad & C \mathrel{\sqsubset\!\!\sim} D, C \mathrel{\sqsubset\!\!\sim} E \implies C \mathrel{\sqsubset\!\!\sim} D \sqcap E \\
\text{(CM)}\quad & C \mathrel{\sqsubset\!\!\sim} D, C \mathrel{\sqsubset\!\!\sim} E \implies C \sqcap D \mathrel{\sqsubset\!\!\sim} E \\
\text{(RM)}\quad & C \mathrel{\sqsubset\!\!\sim} E, C \not\mathrel{\sqsubset\!\!\sim} \neg D \implies C \sqcap D \mathrel{\sqsubset\!\!\sim} E \\
\text{(CONS)}\quad & \top \not\mathrel{\sqsubset\!\!\sim} \bot
\end{aligned}
$$

As in the propositional case, a *preferential set* is a set of defeasible subsumption

statements that is complete for (Refl) to (Cm) and (Cons). A *rational set* is a preferential set that is also complete for (Rm). Preferential and rational sets attempt to capture what we mean by the outcome of a defeasible reasoning process, at least one that proceeds according to the rationality postulates we have laid out so far. These sets can be characterised exactly in terms of DL versions of preferential interpretations, but the fact that DL interpretations contain many individuals leads to some important differences (as opposed to propositional logic, where interpretations can be viewed as talking about the properties of a *single* individual).

Consider the classical subsumption statement $\mathsf{Mathematician} \sqsubseteq \mathsf{Formal}$. Under classical Alc semantics, this can be interpreted as saying that any individual satisfying the concept Mathematician also satisfies the concept Formal. Similarly, we would expect to interpret a defeasible subsumption statement $\mathsf{Mathematician} \mathrel{\sqsubseteq\!\!\!\sim} \mathsf{Creative}$ as saying that any individual satisfying the concept Mathematician *typically* also satisfies the concept Creative. After all, interpreting these statements for each individual separately is what justifies the validity of the rationality postulates for defeasible Alc.

For this reason, rather than consider preference orderings over Dl *interpretations* (as we did in the propositional case), it is more natural to consider preference orderings over the *individuals* in a DL interpretation in order to interpret defeasible subsumption statements.

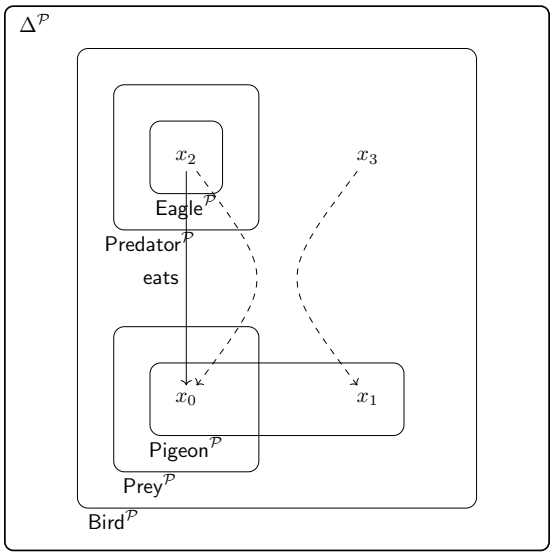
Definition 14. *A* preferential interpretation *is a triple* $\mathcal{P} = \langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}}, \prec^{\mathcal{P}} \rangle$, *where* $\Delta^{\mathcal{P}}$ *and* $\cdot^{\mathcal{P}}$ *constitute an ordinary* Alc *domain and interpretation function, and* $\prec^{\mathcal{P}} \subseteq \Delta^{\mathcal{P}} \times \Delta^{\mathcal{P}}$ *is a strict partial order over the domain satisfying the following smoothness condition:*

$$\textit{For every complex concept } C\textit{, if } C^{\mathcal{P}} \neq \emptyset \textit{ then } \min_{\prec^{\mathcal{P}}} C^{\mathcal{P}} \neq \emptyset.$$

Given a preferential interpretation $\mathcal{P} = \langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}}, \prec^{\mathcal{P}} \rangle$, we denote its underlying Alc interpretation by $\mathcal{I}^{\mathcal{P}} = \langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}} \rangle$. Satisfaction for preferential interpretations is defined as follows:

1. $\mathcal{P} \Vdash C \sqsubseteq D$ iff $\mathcal{I}^{\mathcal{P}} \Vdash C \sqsubseteq D$.
2. $\mathcal{P} \Vdash \mathsf{a} : C$ iff $\mathcal{I}^{\mathcal{P}} \Vdash \mathsf{a} : C$.
3. $\mathcal{P} \Vdash C \mathrel{\sqsubseteq\!\!\!\sim} D$ iff $\min_{\prec^{\mathcal{P}}} C^{\mathcal{P}} \subseteq D^{\mathcal{P}}$.

Example 24. *We can extend our diagrammatic notation for classical interpretations to cover preferential interpretations as well:*



The interpretation of individual, concept and role names remains the same. The only difference is that the preference ordering $\prec^{\mathcal{P}}$ is represented by dashed arrows, with less typical elements being drawn above more typical elements in the style of a Hasse diagram (see Example 1, for instance).

In this example, we have $\min_{\prec^{\mathcal{P}}} \mathsf{Bird}^{\mathcal{P}} = \{x_0, x_1\} = \mathsf{Pigeon}^{\mathcal{P}}$*, and thus the interpretation satisfies* $\mathsf{Bird} \mathrel{\underset{\sim}{\sqsubset}} \mathsf{Pigeon}$ *("birds are typically pigeons").*

In other words, a defeasible subsumption statement $C \mathrel{\underset{\sim}{\sqsubset}} D$ is satisfied by the interpretation if all the most typical individuals satisfying C (with respect to $\prec^{\mathcal{P}}$) also satisfy D. Denoting the set of defeasible formulas a preferential interpretation $\mathcal{P}$ satisfies by $\mathcal{S}_{\mathcal{P}} = \{C \mathrel{\underset{\sim}{\sqsubset}} D : \mathcal{P} \Vdash C \mathrel{\underset{\sim}{\sqsubset}} D\}$, the following theorem proves that preferential interpretations completely characterise preferential sets:

Theorem 5. *[5, p. 11] A set $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{D}}^{\text{ALC}}$ is preferential iff there exists some preferential interpretation $\mathcal{P}$ such that $\mathcal{S} = \mathcal{S}_{\mathcal{P}}$.*

As noted by Britz et al. [5, p. 11] shortly after proving this result, what is surprising here is that no special rationality postulates are necessary for dealing with formulas containing existential ($\exists$) or value ($\forall$) restrictions. In fact, the only postulate that differs between the ALC and propositional cases is (CONS), which expresses the fact that ALC interpretation domains are non-empty and is not critical to a proof of Theorem 5.

Analogously to the propositional case, we can consider the subset of preferential interpretations that have *modular* preference orderings (see Definition 1):

Definition 15. *[5, p. 12] A* modular *interpretation $\mathcal{M} = \langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}}, \prec^{\mathcal{M}} \rangle$ is a preferential interpretation such that the preference ordering $\prec$ is modular.*

And as we would expect, modular interpretations completely characterise *rational* sets:

Theorem 6. *[5, p. 12] A set $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{D}}^{\text{ALC}}$ is rational iff there exists some modular interpretation $\mathcal{M}$ such that $\mathcal{S} = \mathcal{S}_{\mathcal{M}}$.*

Finally, we note that the same simplifications to modular models that we made in the propositional case are similarly valid in the ALC case. In particular, for a large class of defeasible ALC knowledge bases it will suffice to consider the following kinds of modular interpretations:

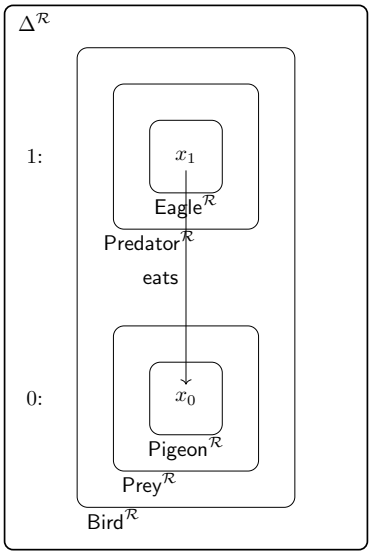
Definition 16. *[5, p. 15] A ranked interpretation is a triple $\mathcal{R} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}}, rk^{\mathcal{R}} \rangle$, where $\Delta^{\mathcal{R}}$ and $\cdot^{\mathcal{R}}$ constitute a classical* ALC *interpretation and $rk^{\mathcal{R}} : \Delta^{\mathcal{R}} \to \mathbb{N}^{\infty}$ is a ranking function satisfying the following convexity condition:*

> *For all $x \in \Delta^{\mathcal{R}}$ such that $rk(x) < \infty$, and all $i \in \mathbb{N}$ such that $0 \leq i < rk(x)$, there exists some $y \in \Delta^{\mathcal{R}}$ such that $rk(y) = i$.*

Satisfaction for ranked interpretations is defined the same way as for preferential interpretations, where $\mathcal{I}^{\mathcal{R}} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}} \rangle$ is the underlying classical interpretation of $\mathcal{R}$:

1. $\mathcal{R} \Vdash C \sqsubseteq D$ iff $\mathcal{I}^{\mathcal{R}} \Vdash C \sqsubseteq D$.
2. $\mathcal{R} \Vdash \mathsf{a} : C$ iff $\mathcal{I}^{\mathcal{R}} \Vdash \mathsf{a} : C$.
3. $\mathcal{R} \Vdash C \mathrel{\sqsubsetneq\!\!\!\sim} D$ iff $\min_{\text{rk}^{\mathcal{R}}} C^{\mathcal{P}} \subseteq D^{\mathcal{P}}$.

Example 25. *A diagrammatic notation for ranked interpretations can be adapted from Example 24 by replacing the dashed arrows (which represent the preference ordering) with numbers on the left indicating the rank of a row's elements:*



Here we have $rk^{\mathcal{R}}(x_0) = 0$ *and* $rk^{\mathcal{R}}(x_1) = 1$*, and thus this ranked interpretation satisfies formulas such as* $\mathsf{Bird} \sqsubseteq \mathsf{Pigeon}$ *("birds are typically pigeons"),* $\mathsf{Eagle} \sqsubseteq \forall \mathsf{eats}.\mathsf{Bird}$ *("eagles typically only eat birds") and* $\mathsf{Prey} \sqcap \mathsf{Birds} \sqsubseteq \neg \mathsf{Eagle}$ *("birds that are preyed on are typically not eagles").*

As we will see shortly, for knowledge bases that admit ranked models we can define an analogue of rational closure in a straightforward way. Fortunately, a large class of ALC knowledge bases admit ranked models:

Lemma 17. *[5, p. 15] Let* $\mathcal{M} = \langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}}, \prec^{\mathcal{M}} \rangle$ *be a modular interpretation such that* $\Delta^{\mathcal{M}}$ *is finite. Then there exists some ranked interpretation* $\mathcal{R}$ *such that* $\mathcal{S}_{\mathcal{M}} = \mathcal{S}_{\mathcal{R}}$*.*

Similarly, it can be shown that finite knowledge bases are equisatisfiable with respect to modular or ranked interpretations:

Lemma 18. *[5, p. 16] Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}, \mathcal{A} \rangle$ *be a finite defeasible knowledge base such that* $\mathcal{A} = \emptyset$*. Then* $\mathcal{K}$ *has a modular model iff it has a ranked model with a finite domain.*

The basic forms of entailment we can consider for defeasible knowledge bases are Tarskian, based on the classical premise that logical consequence corresponds to truth in all models of the given knowledge base:

Definition 17. *[5, p. 13] A knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}, \mathcal{A} \rangle$ preferentially entails *a statement* α *(denoted* $\mathcal{K} \mid\!\!\sim_{\mathcal{P}} \alpha$*) iff* $\mathcal{P} \Vdash \alpha$ *for every preferential model* $\mathcal{P}$ *of* $\mathcal{K}$*. Similarly,* $\mathcal{K}$ rank entails α *(denoted* $\mathcal{K} \mid\!\!\sim_{\mathcal{R}} \alpha$*) iff* $\mathcal{R} \Vdash \alpha$ *for every ranked model* $\mathcal{R}$ *of* $\mathcal{K}$*.*

Preferential and rank entailment suffer from the usual issues with Tarskian entailment relations in a defeasible setting; namely, they are monotonic. For this reason, Britz et al. [5] study analogues of truly defeasible propositional entailment relations, such as rational closure. On the other hand, it's not obvious that analogues even *exist* for a more expressive language like ALC. In a paper on *Propositional Typicality Logic*, for instance, Booth et al. [3] provide an example of an expressive propositional language for which rational closure cannot be defined.

Surprisingly, it turns out that ALC avoids these problems, so long as there are no A-Box assertions. The more complicated case of defeasible entailment in the presence of assertional statements will be covered in a later section. For now, the main technical tool we will need is the notion of *ranked union*:

Definition 18. *[5, p. 16] Let* $\mathbb{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ *be a countable set of ranked interpretations. Then a ranked interpretation* $\mathcal{R} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}}, rk^{\mathcal{R}} \rangle$ *is the* ranked union *of* $\mathbb{R}$ *iff the following holds:*

1. $\Delta^{\mathcal{R}} = \bigsqcup_{\mathcal{R}_i \in \mathbb{R}} \Delta^{\mathcal{R}_i}$*, where each element* $x \in \Delta^{\mathcal{R}_i}$ *is relabelled* $x^{\mathcal{R}_i} \in \Delta^{\mathcal{R}}$*.*

2. $x^{\mathcal{R}_i} \in C^{\mathcal{R}}$ *iff* $x \in C^{\mathcal{R}_i}$.
3. $(x^{\mathcal{R}_i}, y^{\mathcal{R}_j}) \in \mathsf{r}^{\mathcal{R}}$ *iff* $i = j$ *and* $(x, y) \in \mathsf{r}^{\mathcal{R}_i}$.
4. $rk^{\mathcal{R}}(x^{\mathcal{R}_i}) = rk^{\mathcal{R}_i}(x)$.

In other words, the ranked union of a set of ranked interpretations is the result of merging each of their domains together and ranking the elements according to the rankings in their underlying interpretations. Note that we haven't specified the interpretations of individual names, and thus this is only valid when one is considering ALC *without* assertional statements. The main utility of the ranked union is that it lets us construct new *models* of a given knowledge base:

Lemma 19. *[5, p. 16] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base, and $\mathbb{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ a countable set of ranked models of $\mathcal{K}$. Then the ranked union of $\mathbb{R}$ is also a model of $\mathcal{K}$.*

The ranked union of a set of ranked interpretations is, by construction, unique up to isomorphism. In fact, by using the fact that there are only countably many ranked models of a given knowledge base that have a finite domain, we can use the ranked union to construct a *canonical* ranked model for a given knowledge base:

Definition 19. *[5, p. 17] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base, and $\mathbb{R}$ its set of ranked models with finite domain. Then the* big ranked model *of $\mathcal{K}$, denoted $\mathcal{R}(\mathcal{K})$, is defined to be the ranked union of $\mathbb{R}$.*

The rational closure of a defeasible ALC knowledge base has been defined in a number of ways by various authors. Giordano et al. [19] propose a semantic characterisation of rational closure, for instance, using the minimal model construction of Lemma 14. Here, we note that a simple characterisation of rational closure for ALC is given by Britz et al. [5] using the big ranked model construction:

Definition 20. *[5, p. 20] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base. Then a statement $C \sqsubseteq D$ is in the* rational closure *of $\mathcal{K}$, denoted $\mathcal{K} \mid\!\approx_{RC} C \sqsubseteq D$, iff $\mathcal{R}(\mathcal{K}) \Vdash C \sqsubseteq D$.*

There are a number of reasons for considering $\mid\!\approx_{RC}$ to be a reasonable analogue of propositional rational closure. Firstly, the entailment relation $\mid\!\approx_{RC}$ is actually defeasible! Secondly, $\mid\!\approx_{RC}$ satisfies all of the rationality postulates:

Lemma 20. *[5, p. 17] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base. Then the set $\mathcal{S} = \{C \sqsubseteq D : \mathcal{K} \mid\!\approx_{RC} C \sqsubseteq D\}$ is a rational set.*

As we have mentioned, however, there are some limitations with these results. Firstly, ALC is a relatively weak Description Logic, and it's not clear how some of the constructions above would carry over to more expressive DLs. This is an area of active research, which we will not go into. A more pressing limitation, for now, is that we have ignored A-Box statements.

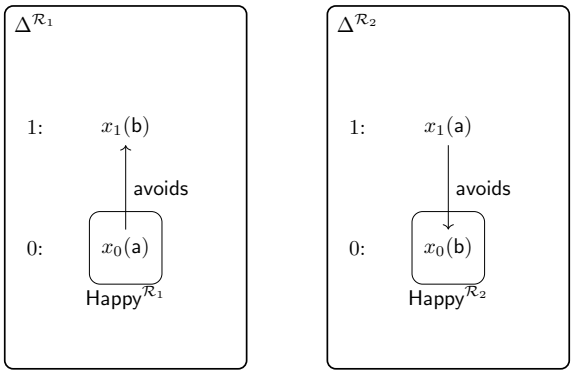
4.3 Defeasibility for the A-Box

To begin our discussion of defeasible reasoning in the presence of assertional statements, let's consider a simple example of the kinds of issues that can occur:

Example 26. *Consider the following defeasible knowledge base, which states that individuals are typically happy and avoid only unhappy people, and that* a *avoids* b*:*

$$\mathcal{K} = \{\top \sqsubseteq \mathsf{Happy}, \top \sqsubseteq \forall \mathsf{avoids}.\neg\mathsf{Happy}, (\mathsf{a},\mathsf{b}) : \mathsf{avoids}\}$$

We can construct at least two ranked models of $\mathcal{K}$ *that are* minimal *in a sense to be defined later. The first thing we could do is decide that* a *is typical, i.e. that* $rk^{\mathcal{R}}(\mathsf{a}^{\mathcal{R}}) = 0$*. From the fact that* $(\mathsf{a},\mathsf{b}) : \mathsf{avoids}$ *holds, we would conclude that* $\mathsf{b} : \neg\mathsf{Happy}$ *must hold and therefore that* $rk^{\mathcal{R}}(\mathsf{b}^{\mathcal{R}}) \geq 1$*. On the other hand, we could also decide that* b *is typical, i.e. that* $rk^{\mathcal{R}}(\mathsf{b}^{\mathcal{R}}) = 0$*. For the same reasons, we would then conclude that* $rk^{\mathcal{R}}(\mathsf{a}^{\mathcal{R}}) \geq 1$*. These two possibilities are represented by the following two ranked models:*



In the above example, it's not clear how we would go about defining the big ranked model of $\mathcal{K}$ along the lines of Definition 19, since $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfy contradictory assertional statements! One way of understanding this is that there is no longer a unique minimal ranked model of $\mathcal{K}$ - we cannot lower the rank of any of the individuals in $\mathcal{R}_1$ or $\mathcal{R}_2$ without violating one of the statements in $\mathcal{K}$.

To continue with our analysis of rational closure, it turns out to be convenient to use the minimal model semantics of Giordano et al. [19], as we did in the propositional case. We will follow the presentation of Casini et al. [10] and work with the notion of *compatibility*, which eases some of the technical details:

Definition 21. *[10, p. 4] Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}\rangle$ *be a defeasible knowledge base, and consider an interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$*. Then an individual* $x \in \Delta^{\mathcal{I}}$ *is* $\langle \mathcal{T}, \mathcal{D}\rangle$-compatible with respect to $\mathcal{I}$ *iff for every complex concept* C *such that* $x \in C^{\mathcal{I}}$*, we have* $\langle \mathcal{T}, \mathcal{D}\rangle \not\models_{\mathcal{R}} \neg C$*.*

In other words, an individual is compatible if it doesn't violate any of the consequences of $\mathcal{K}$. We can extend this notion to an *interpretation* by saying it is compatible if it contains representatives of every compatible individual. This is similar to the notion of a saturated model in classical logic, which contains representatives of every consistent set of formulas:

Definition 22. *[10, p. 4] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base, and consider an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$. Then $\mathcal{I}$ is* $\langle \mathcal{T}, \mathcal{D} \rangle$-compatible *iff for every interpretation $\mathcal{J}$ and individual $x \in \Delta^{\mathcal{J}}$ that is $\langle \mathcal{T}, \mathcal{D} \rangle$-compatible with respect to $\mathcal{J}$, there exists some $y \in \Delta^{\mathcal{I}}$ such that $x \in C^{\mathcal{J}}$ iff $y \in C^{\mathcal{I}}$.*

We say a ranked interpretation $\mathcal{R} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}}, \prec^{\mathcal{R}} \rangle$ is $\langle \mathcal{T}, \mathcal{D} \rangle$-compatible iff $\mathcal{I} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}} \rangle$ is. Given a classical interpretation $\mathcal{I}$, we denote by $\mathbb{R}^{\mathcal{I},\mathcal{T},\mathcal{D}}$ the set of $\langle \mathcal{T}, \mathcal{D} \rangle$-compatible ranked interpretations $\mathcal{R}$ that agree on $\mathcal{I}$. This set of interpretations can be ordered according to how typical they rank elements of the domain, in similar fashion to Definition 9 for propositional logic:

Definition 23. *[10, p. 5] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base, and consider an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$. Then given ranked interpretations $\mathcal{R}_1, \mathcal{R}_2 \in \mathbb{R}^{\mathcal{I},\mathcal{T},\mathcal{D}}$, we define $\mathcal{R}_1 \leq_{\mathcal{I}} \mathcal{R}_2$ iff $rk^{\mathcal{R}_1}(x) \leq rk^{\mathcal{R}_2}(x)$ for all $x \in \Delta^{\mathcal{I}}$.*

A ranked interpretation $\mathcal{R}$ is a *minimal model* of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}, \mathcal{A} \rangle$ iff it satisfies $\mathcal{K}$ and there exists some $\langle \mathcal{T}, \mathcal{D} \rangle$-compatible interpretation $\mathcal{I}$ such that $\mathcal{R} \in \min_{\leq_{\mathcal{I}}} \mathbb{R}^{\mathcal{I},\mathcal{T},\mathcal{D}}$. This gives us a convenient reformulation of rational closure for knowledge bases without assertional statements:

Lemma 21. *[10, p. 5] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a defeasible knowledge base. Then $\mathcal{K} \mid\!\!\approx_{RC} C \sqsubseteq D$ iff $\mathcal{R} \Vdash C \sqsubseteq D$ for some minimal model $\mathcal{R}$ of $\mathcal{K}$.*

Note that the particular choice of minimal model in Lemma 21 is irrelevant, since the minimal models of $\mathcal{K}$ satisfy exactly the same set of defeasible subsumption statements. When it comes to knowledge bases with an A-Box, on the other hand, there is an additional consideration to think about. In Example 26, we saw an example of a knowledge base with minimal models satisfying *different* sets of statements. Thus in order to extend our notion of rational closure, we need to choose between inconsistency or a more skeptical approach to entailment:

Definition 24. *[10, p. 9] Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D}, \mathcal{A} \rangle$ be a defeasible knowledge base. Then for any statement α, $\mathcal{K} \mid\!\!\approx_{RC} \alpha$ iff $\mathcal{R} \Vdash \alpha$ for* every *minimal model $\mathcal{R}$ of $\mathcal{K}$.*

When $\mathcal{A} = \emptyset$, Lemma 21 tells us that Definition 24 is equivalent to the definition of rational closure for knowledge bases without assertional statements, so this is well-defined. In general, however, we should expect rational closure to behave worse in the presence of an A-Box, as we lose the *Single Model Property*. The rational closure of a knowledge base can no longer necessarily be characterised by a single ranked interpretation!

Chapter 5

Datalog

5.1 Background

Datalog is a language for programming and querying databases with *ontologies*, i.e. structured information about the way various concepts and facts in the database relate to each other. Unlike many other paradigmatic database languages, such as SQL, Datalog is based on the *logic programming* paradigm, meaning that its programs are represented as sets of formulas in a formal mathematical logic (in this case first-order logic). Datalog is also fully declarative, in the sense that the order of statements in a Datalog program is irrelevant; programs are described in terms of facts and rules relating them, rather than in terms of the control flow of the program execution.

Unlike other logic programming languages, such as Prolog, Datalog is *not* Turing-complete and hence not suited for general-purpose computation. This weakness is also its strength, however, as program evaluation can exploit Datalog's syntactic restrictions to guarantee both termination and efficiency. Its logical foundations also allow for the expression of novel programming patterns, such as mutually recursive rules, which have only recently been implemented in more mainstream database programming languages (see the introduction by Calì et al. [7], for instance).

Formally, Datalog can be viewed as a decidable (and moreover tractable!) fragment of first-order logic. We will describe the syntax of Datalog theories from this point of view, referring readers who are unfamiliar with first-order logic to Chapter 2.3.

The language of a Datalog theory is a restricted first-order language $\Sigma = \langle \text{PRED}, \text{CONST}, \text{VAR} \rangle$, in which we assume the set of function symbols is empty, and we assume that the set VAR of variable symbols is countably infinite for convenience. The set PRED of predicate symbols is further subdivided into the set PRED_E of *extensional* predicate symbols, and the set PRED_I of *intensional*

predicate symbols.

A *literal* is defined to be any formula $\pm\mathsf{p}(t_1, \ldots, t_{\mathtt{ar}(\mathsf{p})})$ that is either an atom or the negation of an atom, and is said to be extensional or intensional according to whether the predicate symbol p is extensional or intensional respectively. A literal is said to be *ground* if it contains no variable symbols. A *positive literal* is just an atom, and a *negative literal* is the negation of an atom. Variants of Datalog largely differ in terms of which kinds of literals can appear in various positions in a formula, and which kinds of formulas are allowed within a theory.

A *fact* is defined to be any positive, extensional ground literal $\mathsf{p}(\mathsf{c}_1, \ldots, \mathsf{c}_{\mathtt{ar}(\mathsf{p})})$, where the $\mathsf{c}_i \in \textsc{const}$ are all constant symbols. Note that in the absence of function symbols, constant symbols are the only ground terms in the language. A *rule* is defined to be any formula $L_0 \leftarrow L_1, \ldots, L_n$, where the L_i are positive literals satisfying the following *safety conditions*:

1. The literal L_0 is intensional.
2. Every variable symbol occurring in L_0 must occur in L_i for some $1 \leq i \leq n$.

These safety condition ensures that program evaluation always terminates [12, p. 147], a topic we will return to later. The literal on the left-hand side of a rule is called its *head*, while the literals on the right-hand side are called its *body*. We denote the set of facts by $\mathcal{L}_f$, and the set of rules by $\mathcal{L}_{\rightarrow}$. Together, these constitute the language of *Pure Datalog*, which we denote by $\mathcal{L} = \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$.

When describing a Datalog theory, a distinction is typically made between the facts in the theory (which are given in practice by an input database known as the *extensional database*) and the rules (which are computed in the course of program evaluation) [7, p. 2]. We thus define a *Pure Datalog database* to be any set of facts $\mathcal{D} \subseteq \mathcal{L}_f$ and a *Pure Datalog program* to be any set of rules $\mathcal{P} \subseteq \mathcal{L}_{\rightarrow}$. A *Pure Datalog theory* is simply defined to be any combination $\Gamma = \langle \mathcal{D}, \mathcal{P} \rangle$ of the two.

Example 27. *Consider a Datalog language* Σ *where* $\textsc{pred}_E = \{\mathsf{e}(\cdot,\ \cdot)\}$, $\textsc{pred}_I = \{\mathsf{c}(\cdot,\ \cdot)\}$ *and* $\textsc{const} = \{\mathsf{v}_i : i \in \mathbb{N}\}$. *Our goal is to describe a Pure Datalog program that computes the connectedness of a directed graph, where vertices are represented by constants, and edges are represented by the predicate* e*:*

$$\mathsf{c}(X, Y) \leftarrow \mathsf{e}(X, Y).$$
$$\mathsf{c}(X, Z) \leftarrow \mathsf{c}(X, Y), \mathsf{e}(Y, Z).$$

These rules express an inductive definition of connectivity: X *and* Z *are connected if there is either an edge between them, or* X *has an edge to some vertex* Y *such that* Y *and* Z *are connected.*

Datalog theories can be interpreted as describing corresponding first-order theories, a viewpoint which is known as the *logical semantics* for Pure Datalog [12,

p. 148]. These corresponding first-order theories are defined in terms of a *translation operator* Tr, which maps a Datalog rule or fact α to its corresponding first-order formula $\mathrm{Tr}(\alpha)$:

1. If α is a fact $\mathsf{p}(\mathsf{c}_1, \ldots, \mathsf{c}_{\mathtt{ar}(\mathsf{p})})$, then $\mathrm{Tr}(\alpha) = \mathsf{p}(\mathsf{c}_1, \ldots, \mathsf{c}_{\mathtt{ar}(\mathsf{p})})$.
2. If α is a rule $L_0 \leftarrow L_1, \ldots, L_n$, then $\mathrm{Tr}(\alpha) = \forall \vec{\mathsf{x}} \,.\, (L_1 \wedge \cdots \wedge L_n \rightarrow L_0)$, where $\vec{\mathsf{x}}$ is the tuple of variable symbols appearing in α.

In other words, a fact is interpreted by itself as a first-order formula, whereas a rule is interpreted by its universal closure as a clause. This sugggests the possibility of using first-order semantic stuctures, such as Herbrand interpretations, to provide a meaning to Datalog theories, an idea that dates back to Van Emden et al. [35]. Given a Datalog theory $\Gamma = \langle \mathcal{D}, \mathcal{P} \rangle$ and a Herbrand interpretation $\mathcal{H}$ over the theory's language (see Section 2.3 for a definition), we say that $\mathcal{H}$ *satisfies* Γ (or that $\mathcal{H}$ is a *model* of Γ) iff $\mathcal{H} \Vdash \mathrm{Tr}(\alpha)$ for every rule or fact $\alpha \in \Gamma$.

Example 28. *Let $\Gamma = \langle \mathcal{D}, \mathcal{P} \rangle$, where $\mathcal{D} = \{\mathsf{e}(\mathsf{v}_1, \mathsf{v}_2)\}$, and $\mathcal{P}$ is the program from Example 27. Then the following Herbrand interpretations all satisfy Γ:*

$$\mathcal{H}_1 = \{\mathsf{e}(\mathsf{v}_1, \mathsf{v}_2), \mathsf{c}(\mathsf{v}_1, \mathsf{v}_2)\}$$
$$\mathcal{H}_2 = \{\mathsf{e}(\mathsf{v}_1, \mathsf{v}_2), \mathsf{e}(\mathsf{v}_2, \mathsf{v}_3), \mathsf{c}(\mathsf{v}_1, \mathsf{v}_2), \mathsf{c}(\mathsf{v}_1, \mathsf{v}_3), \mathsf{c}(\mathsf{v}_2, \mathsf{v}_3)\}$$
$$\mathcal{H}_3 = \{\mathsf{e}(\mathsf{v}_1, \mathsf{v}_2), \mathsf{c}(\mathsf{v}_1, \mathsf{v}_2), \mathsf{c}(\mathsf{v}_4, \mathsf{v}_5), \mathsf{c}(\mathsf{v}_1, \mathsf{v}_1)\}$$

This lets us give a precise account of what it means for a Pure Datalog theory $\Gamma = \langle \mathcal{D}, \mathcal{P} \rangle$ to *compute* or *entail* a given fact $\alpha \in \mathcal{L}_f$, which we denote by $\Gamma \models \alpha$. In particular, $\Gamma \models \alpha$ iff every Herbrand model $\mathcal{H}$ of Γ satisfies $\mathrm{Tr}(\alpha)$ [12, p. 148]. A common way of phrasing this, which emphasises the computational aspect of Pure Datalog, is that the program $\mathcal{P}$ *computes* the fact α given the input database $\mathcal{D}$.

In fact, there is a convenient way to characterise the set of facts entailed by a theory. One of the properties that sets Pure Datalog apart from general first-order logic, aside from its restricted language and comparatively simple syntax, is that it satisfies the *model intersection property*:

Lemma 22. *[12, p. 149] Let Γ be a Pure Datalog theory, and $\mathbb{H} = \{\mathcal{H}_1, \ldots\}$ a (possibly infinite) set of Herbrand models of Γ. Then $\mathcal{H} = \bigcap \mathbb{H}$ is also a Herbrand model of Γ.*

A consequence of the model intersection property is that every Pure Datalog theory Γ has a *minimal Herbrand model* $\mathcal{H}_\Gamma$, which is defined to be the intersection of the set of all Herbrand models of the theory [12, p. 149]. This minimal model precisely characterises the facts entailed by Γ:

Lemma 23. *[12, p. 149] Let Γ be a Pure Datalog theory, and consider some fact $\alpha \in \mathcal{L}_f$. Then $\Gamma \models \alpha$ iff $\alpha \in \mathcal{H}_\Gamma$.*

The nice thing about the logical semantics for Pure Datalog is that it tells us exactly which facts are computed by a given program, without specifying anything about *how* the computation should be performed. This is what is meant when Datalog is referred to as a declarative language - a given Datalog program's meaning is completely determined by its *set* of rules, and doesn't depend on their order or on any kind of implementation detail.

Something else to note about the semantics we have just presented is that entailment is only defined for *facts*. While it wouldn't be hard to do so by making use of the translation operator, we have refrained from defining a notion of entailment for rules, negative literals or more complicated kinds of formulas. The reason for this is somewhat subtle, and has to do with the way incomplete information is handled in Datalog versus first-order logic.

First-order logic adopts a position known as the *open-world assumption* (OWA), which states that a formula may be true or false irrespective of whether it is *known* to be true or false. Phrased in terms of entailment, if we know that a first-order theory Γ holds, and we have that $\Gamma \not\models \alpha$ and $\Gamma \not\models \neg\alpha$ for some formula α, then we cannot make *any* assumptions about whether α holds or not. After all, both options are a logical possibility, and we wouldn't want an account of logic that draws possibly erroneous conclusions.

On the other hand, Datalog is intended to be a database query language, and thus has different design goals. If we consulted a database of train schedules to determine which trains were leaving at noon, for instance, an answer of "*unknown*" would be as good as no answer at all! Datalog deals with this by adopting a position known as the *closed-world assumption* (CWA), which states that if a formula is not known to be true, then it should be assumed to be false. Phased in terms of entailment, this is the following principle [12, p. 158]:

> *If a fact is not entailed by a Datalog theory, then we should conclude that the negation of the fact is true.*

This principle, combined with the result of Lemma 23, allows us to define what it means for a Datalog theory Γ to entail the *negation* of a fact $\alpha \in \mathcal{L}_f$. In particular, we define $\Gamma \models \neg\alpha$ to hold iff $\Gamma \not\models \alpha$, or equivalently iff $\mathcal{H}_\Gamma \Vdash \neg\alpha$. Note that this is very different to (and in fact incompatible with) the definition of entailment used in first-order logic!

While this lets us deduce negative facts, it's worth noting that in Pure Datalog we cannot use these negative facts in any further deduction, as rules are required to contain only positive literals. Various extensions of Pure Datalog have been proposed in the literature to deal with this problem, which usually requires assuming an approximation to the CWA rather than the full principle. In the next section we will briefly overview some of these extensions of Pure Datalog, and set the stage for a KLM-style form of Defeasible Datalog.

5.2 Extensions of Datalog

Before defining any extensions, let's first discuss why allowing negative literals in Datalog rules can cause problems with the CWA. Consider the program $\mathcal{P} = \{\mathsf{p}(\mathsf{a}) \leftarrow \neg\mathsf{q}(\mathsf{a})\}$, for instance. Computing its set of Herbrand models for an empty input database $\mathcal{D} = \emptyset$, one finds that it has not one minimal model (which we would expect for Pure Datalog theories, by Lemma 23), but *two* minimal models [12, p. 159]:

1. $\mathcal{H}_1 = \{\mathsf{p}(\mathsf{a})\}$
2. $\mathcal{H}_2 = \{\mathsf{q}(\mathsf{a})\}$

Here, it's clear that $\langle \mathcal{P}, \mathcal{D} \rangle \not\models \mathsf{p}(\mathsf{a})$ and $\langle \mathcal{P}, \mathcal{D} \rangle \not\models \mathsf{q}(\mathsf{a})$. Hence, if we were to apply the full CWA principle, we would conclude that we should assume both $\neg\mathsf{p}(\mathsf{a})$ and $\neg\mathsf{q}(\mathsf{a})$ [12, p. 159]. But this is incompatible with both minimal models given above! This leaves us with two choices if we want things to work out consistently:

1. We can restrict the allowed rules so as to avoid contradicting the CWA.
2. We can drop the full CWA principle and assume a weaker version of it instead.

The first option is essentially ruled out by our earlier example, in which we had a Datalog program with a single negative literal in its rule body. It's unclear how one could restrict the syntax further than this without ending up back at Pure Datalog. Allowing negative literals in rule heads, on the other hand, is immediately ruled out for safety (i.e. query termination and efficiency) reasons [12, p. 159].

Thus we are left with the second option, which is to adopt a weaker form of the CWA principle. The two most common approaches in the literature are known as *Stratified Datalog* $^\neg$ [12, p. 159] and *Inflationary Datalog* $^\neg$ [12, p. 160], both of which work by applying the CWA *locally* at each step of a program evaluation. In order to compare the two, let's define their common language. A *Datalog*$^\neg$ *fact* is defined to be any extensional ground literal, and a *Datalog*$^\neg$ *rule* is defined to be any formula $L_0 \leftarrow L_1, \ldots, L_n$, where the L_i are literals satisfying the following safety conditions:

1. The literal L_0 is positive and intensional.
2. Every variable symbol occurring in L_0 must occur in L_i for some $1 \leq i < n$.
3. Every variable symbol occurring in a negative literal L_i must occur in a positive literal L_j for some $1 \leq j < n$.

We denote the set of Datalog$^\neg$ facts by $\mathcal{L}_f^\neg$, and the set of Datalog$^\neg$ rules by $\mathcal{L}_\rightarrow^\neg$. Note that $\mathcal{L}_f \subseteq \mathcal{L}_f^\neg$ and $\mathcal{L}_\rightarrow \subseteq \mathcal{L}_\rightarrow^\neg$, with equality occurring iff the underlying Datalog language has no predicate symbols. Thus the language of Datalog$^\neg$ is (in all non-trivial cases) a strict extension of the language of Pure Datalog.

The *stratified* semantics for Datalog$^\neg$ is based on the following idea: when we evaluate a Datalog$^\neg$ rule with negative literals, we should evaluate the predicates corresponding to these negative literals, and then apply the CWA *locally* to these predicates [12, p. 159]. If we evaluate our earlier program $\mathcal{P} = \{\mathsf{p}(\mathsf{a}) \leftarrow \neg\mathsf{q}(\mathsf{a})\}$ over an empty input database according to this idea, we find that in order to evaluate the predicate p, we first have to evaluate the predicate q, as it appears as a negative literal in the rule for p.

However, since our input database is empty, our program does *not* compute $\mathsf{q}(a)$. Applying the CWA locally, this tells us that we should conclude that $\neg\mathsf{q}(a)$ holds, and hence by the sole rule of $\mathcal{P}$ we conclude that $\mathsf{p}(a)$ holds as well. Thus the *stratified* model of $\mathcal{P}$ that we compute is given by $\mathcal{H} = \{\mathsf{p}(a)\}$, which also happens to be one of the minimal models of $\mathcal{P}$.

This policy clearly introduces an ordering on the evaluation of predicates. In order to evalute a predicate p, we first have to enumerate all the rules that have p in their head, and then evaluate all the predicate symbols occurring in negative literals in those rules. In some cases, this process might never terminate, as there could be *cycles* in the graph of dependencies between predicates. Consider the program $\mathcal{P} = \{\mathsf{p}(\mathsf{a}) \leftarrow \neg\mathsf{q}(\mathsf{a}), \mathsf{q}(\mathsf{a}) \leftarrow \neg\mathsf{p}(\mathsf{a})\}$, for instance, in which evaluating p required evaluating q, and vice-versa.

Datalog programs which do *not* have these cycles are known as *stratified programs*, and *Stratified Datalog*$^\neg$ only permits stratified programs in order to guarantee termination. When $\mathcal{P}$ is a stratified program, evaluating $\mathcal{P}$ over an input database according to the stratified semantics is guaranteed to produce a minimal model of $\mathcal{P}$, known as its *perfect model* [12, p. 160].

The *inflationary* semantics for Datalog$^\neg$ does not have these restrictions, and is applicable to all Datalog$^\neg$ programs, whether they are stratified or not. The idea is to evaluate the program *concurrently* - at each evaluation step, every rule in the program is applied simultaneously to derive new facts, with the CWA being applied *locally* to the set of facts derived in previous steps [12, p. 160].

Let's evaluate the non-stratified program $\mathcal{P} = \{\mathsf{p}(\mathsf{a}) \leftarrow \neg\mathsf{q}(\mathsf{a}), \mathsf{q}(\mathsf{a}) \leftarrow \neg\mathsf{p}(\mathsf{a})\}$ over an empty input database according to this idea. First, we first apply the CWA locally to conclude that $\neg\mathsf{p}(\mathsf{a})$ and $\neg\mathsf{q}(\mathsf{a})$ are applicable in rules (since we have derived no facts so far). Then, we apply both rules concurrently to conclude that $\mathsf{p}(\mathsf{a})$ and $\mathsf{q}(\mathsf{a})$ should both hold. Thus the model of $\mathcal{P}$ computed by the inflationary semantics is given by $\mathcal{H} = \{\mathsf{p}(\mathsf{a}), \mathsf{q}(\mathsf{a})\}$.

The downside of *Inflationary Datalog*$^\neg$ is that the models it computes are not minimal in general. However, it can be shown that given any stratified Datalog$^\neg$ program $\mathcal{P}$, there exists a program $\mathcal{P}'$ computable from $\mathcal{P}$ such that

the inflationary model of $\mathcal{P}'$ is exactly the perfect model of $\mathcal{P}$ [12, p. 161]. Thus, Inflationary Datalog$^\neg$ is strictly more expressive than Stratified Datalog$^\neg$.

5.3 A Methodology for Defeasible Datalog

All of the extensions of Pure Datalog in the previous section have semantics based on local versions of the CWA principle. This has historically been motivated by the need for Datalog programs to have an efficient, declarative semantics, in line with their intended purpose as a database *query* language. In this section, we will lay the groundwork for a different kind of extension of Pure Datalog, one which is motivated instead by the use of Datalog as an *ontology* language.

Our approach follows the lines of Chapters 3 and 4. First, we will define analogues of the *rationality postulates* for Datalog, and extend the notion of a *rational set* to Datalog programs. This gives us a set of defeasible properties, analogous to those proposed by Kraus et al. [22] and Lehmann et al. [25], that should be true of any defeasible reasoning process over Datalog programs.

Preliminary work in this direction has been done by Morris et al. [28] and Harrison et al. [21], who define and implement a syntactic version of Rational Closure for Datalog programs based on analogues of the rationality postulates. One of the challenges highlighted by their work is that some of the rationality postulates can only be expressed in an extension of the language of Pure Datalog, which we will look at in detail in the next chapter.

Second, we will define a *semantics* that completely characterises rational Datalog programs. As we will see, this also introduces some challenges, as the language of Datalog is significantly more expressive than that of propositional logic or the Description Logic ALC. Datalog allows for restricted forms of universal quantification, for instance, and thus we will be required to *extend* the rationality postulates in order to achieve this goal. This is surprising, as the corresponding approach for Description Logics can be carried through in its entirety using nothing more than the standard propositional rationality postulates (see the discussion after Theorem 5, for instance).

Finally, we will define a semantic analogue of *rational closure* for Datalog programs that both characterises and extends the syntactic version proposed by Morris et al. [28] and Harrison et al. [21]. Our approach solves some of the issues with their syntactic algorithm, such as its inability to handle *exceptional individuals*. We will discuss these issues in more detail at the end of the next chapter.

Chapter 6

Defeasible Datalog

Defeasible Datalog is an extension of Pure Datalog first proposed by Morris et al. [28] and Harrison et al. [21], in which one can express *defeasible rules* in addition to the classical rules and facts of Pure Datalog. These defeasible rules express relationships that *typically* hold but which may have exceptions, much like the "$\mathrel{|\!\sim}$" connective of propositional KLM or "$\sqsubseteq\!\!\!\sim$" statements in defeasible ALC. This is an explicit form of defeasibility, different in both style and intention from the implicit forms of defeasibility that feature in extensions like Stratified Datalog$^{\neg}$.

Like Pure Datalog, the underlying language of a Defeasible Datalog theory is a restricted first-order language $\Sigma = \langle \textsc{pred}, \textsc{const}, \textsc{var} \rangle$, where we assume that the sets of predicate and constant symbols are *at most* countably infinite, and that the set VAR of variable symbols *is* countably infinite.

Unlike Pure Datalog, however, we will make no distinction between intensional and extensional predicates in what follows, and we won't concern ourselves with *safety* constraints on the formulas we allow in Defeasible Datalog. These constraints are present in Pure Datalog in order to guarantee query efficiency and termination. This is important for Defeasible Datalog too, of course, but our focus is primarily on providing a robust semantics for defeasible reasoning over Datalog theories, with computational efficiency a topic that can be returned to in future work. The defeasible extensions of propositional logic and ALC that we looked at in Chapters 3 and 4 are both computationally reducible to their classical versions, for instance, so we do not expect this to be a serious shortcoming of our work.

The syntax of Defeasible Datalog is defined in terms of *compounds* [28], which we define inductively as follows:

1. If $\mathsf{p}(t_1, \ldots, t_{\mathfrak{ar}(\mathsf{p})})$ is an atomic formula, then it is also a compound.
2. If A is a compound, then $\neg A$ is a compound.

3. If A and B are compounds, then $A \wedge B$ and $A \vee B$ are compounds.

In other words, compounds are boolean combinations of atoms. The set of all compounds is denoted $\mathcal{L}_c$, and when writing a generic compound we will stick to the convention of denoting them with capital letters. A *fact* is defined to be any ground compound, i.e. any compound $A \in \mathcal{L}_c$ containing no variable symbols, and we denote the set of facts by $\mathcal{L}_f \subseteq \mathcal{L}_c$. An *atom* is defined to be any atomic compound, i.e. any compound of the form $\mathsf{p}(t_1, \ldots, t_{\mathtt{ar}(\mathsf{p})})$, and we denote the set of atoms by $\mathcal{L}_a$.

A *rule* is defined to be any formula $A \leftarrow B$, where A and B are any two compounds. Similarly, a *defeasible rule* is any formula $B \mathrel{\sim\!\!\!<} A$, where A and B are any two compounds. We denote the set of rules by $\mathcal{L}_{\rightarrow}$, and the set of defeasible rules by $\mathcal{L}_{\rightsquigarrow}$.

Like Pure Datalog, we distinguish between the *facts* and *rules* in a given Defeasible Datalog theory. We thus define a *Defeasible Datalog database* to be any set of facts $\mathcal{D} \subseteq \mathcal{L}_f$, and a *Defeasible Datalog program* to be any set of rules $\mathcal{P} \subseteq \mathcal{L}_{\rightarrow} \cup \mathcal{L}_{\rightsquigarrow}$. A *Defeasible Datalog theory* is any combination $\Gamma = \langle \mathcal{D}, \mathcal{P} \rangle$ of the two.

Example 29. *Consider the following Defeasible Datalog knowledge base, based on an example due to Pensel et al. [30], which encodes a basic ontology about workers and their bosses:*

$$\begin{aligned} \mathsf{worker}(X) &\leftarrow \mathsf{boss}(X) \\ \mathsf{responsible}(X) &\mathrel{\sim\!\!\!<} \mathsf{boss}(X) \\ \mathsf{boss}(Y) &\mathrel{\sim\!\!\!<} \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y) \end{aligned}$$

This states that all bosses are workers, that bosses are typically responsible, and that a worker's superior is typically a boss. Intuitively, the use of defeasible rules allows for the presence of exceptional facts in the database, such as a boss who isn't responsible:

$$\mathsf{boss}(\mathsf{Tommie}) \wedge \neg\mathsf{responsible}(\mathsf{Tommie})$$

It's worth pointing out a few things about the language of Defeasible Datalog versus the language of Pure Datalog. Firstly, as we noted earlier, there are no safety restrictions on facts, rules or defeasible rules. We do not view this as a hard requirement - indeed, once we have defined an appropriate semantics for Defeasible Datalog, the next logical step is to restrict the language so as to ensure computational efficiency.

Secondly, Defeasible Datalog facts and rules are significantly more expressive than their Pure Datalog counterparts. We have allowed arbitrary boolean combinations of atoms to form compounds, and thus we can express negation and disjunction in rule heads and bodies. We can also express things like negative

or disjunctive facts. Again, this is not viewed as a hard requirement. As we will see later, this expressiveness is only necessary in order to define appropriate analogues of the rationality postulates. The purpose of these postulates is simply to motivate our choice of semantics, and there is nothing stopping us from restricting the language later for efficiency reasons.

6.1 Semantics for Classical Formulas

We begin our analysis of Defeasible Datalog with a short digression on how we intend to interpret the *classical* part of the Defeasible Datalog language, i.e. facts and (non-defeasible) rules. As we have already stated, our approach to defeasible reasoning over Datalog knowledge bases is motivated by the use of Datalog as an ontology language, and as such we will be ignoring considerations from Chapter 5 such as the Closed-World Assumption.

An implication of this is that we will be interpreting negation and disjunction in a very different way to extensions of Datalog such as Datalog$^\neg$. In fact, we will interpret them *classically*, in the same way that first-order logic does. Given a fixed first-order language $\Sigma = \langle \text{PRED}, \text{CONST}, \text{VAR} \rangle$, we recall from Chapter 2 that the *Herbrand base* of Σ is the set $\mathbb{B}$ of ground atoms over Σ, and that a *Herbrand interpretation* is a subset $\mathcal{H} \subseteq \mathbb{B}$.

We say a Herbrand interpretation $\mathcal{H}$ *satisfies* a fact or rule $\alpha \in \mathcal{L}_f \cup \mathcal{L}_\rightarrow$ iff $\mathcal{H}$ satisfies the first-order formula $\text{Tr}(\alpha)$, where $\text{Tr}(\cdot)$ is the translation operator of Chapter 5. Satisfaction is denoted $\mathcal{H} \Vdash \alpha$, as usual.

Example 30. *Consider a fixed language* Σ *where* PRED $= \{\mathsf{worker}(\cdot),\ \mathsf{boss}(\cdot)\}$ *and* CONST $= \{\mathsf{Josh},\ \mathsf{Kahla}\}$*, and suppose we have the following Herbrand interpretation:*

$$\mathcal{H} = \{\mathsf{worker}(\mathsf{Josh}),\ \mathsf{worker}(\mathsf{Kahla}),\ \mathsf{boss}(\mathsf{Kahla})\}$$

Then $\mathcal{H} \Vdash \mathsf{boss}(\mathsf{Kahla}) \vee \mathsf{boss}(\mathsf{Josh})$*, because* $\mathcal{H}$ *satisfies* $\mathsf{boss}(\mathsf{Kahla})$ *as a first-order formula. We also have* $\mathcal{H} \Vdash \mathsf{worker}(X) \leftarrow \mathsf{boss}(X)$*, because* $\mathcal{H}$ *satisfies the first-order formula* $\forall x.\mathsf{boss}(x) \rightarrow \mathsf{worker}(x)$*.*

We say a fact $A \in \mathcal{L}_f$ or a rule $B \leftarrow A \in \mathcal{L}_\rightarrow$ is a *classical tautology* if it is satisfied by *every* Herbrand interpretation over *every* extension of the language Σ. We will abuse notation slightly and refer to an open compound $A \in \mathcal{L}_c$ as a classical tautology if the rule $A \leftarrow \top$ is a classical tautology.

The fact that we consider extensions of the language here is what we mean by interpreting formulas *classically*. In particular, we are interested in extensions of the language that have more constant symbols, as otherwise we drift back into a closed-world interpretation of formulas, in which whether or not a formula is considered a tautology depends crucially on what one knows:

Example 31. *Consider the first-order language* $\Sigma = \langle \text{PRED}, \text{CONST} \rangle$*, where* $\text{PRED} = \{\mathsf{pigeon}(\cdot)\}$ *and* $\text{CONST} = \{\mathsf{Tim}\}$*. Then every Herbrand interpretation over* Σ *satisfies* $\mathsf{pigeon}(X) \leftarrow \mathsf{pigeon}(\mathsf{Tim})$*, as there are simply no constant symbols other than* Tim*. On the other hand, we do not want to conclude that* $\mathsf{pigeon}(X) \leftarrow \mathsf{pigeon}(\mathsf{Tim})$ *is a classical tautology, as its first-order translation* $\forall x.\mathsf{pigeon}(\mathsf{Tim}) \rightarrow \mathsf{pigeon}(x)$ *is* not *true in every first-order structure.*

We say a classical formula $\alpha \in \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$ is a *classical consequence* of a set of formulas $\mathcal{S} \subseteq \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$ iff α is satisfied by every Herbrand model of $\mathcal{S}$ over every extension of the language Σ. We also say that $\mathcal{S}$ is *deductively closed* if it contains all of its classical consequences.

In Chapter 2 we noted that Herbrand semantics is not equivalent to standard first-order semantics in general (see Lemma 3). This implies that there might be formulas that are classically entailed by a set $\mathcal{S}$ (in the standard first-order sense), but which aren't classical consequences of $\mathcal{S}$ (in the Herbrand sense given above). Fortunately, so long as we restrict our attention to Defeasible Datalog formulas, this is never an issue:

Lemma 24. *Let* $\mathcal{S} \subseteq \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$ *be an arbitrary set of formulas. Then the formula* α *is a classical consequence of* $\mathcal{S}$ *iff* $Tr(\alpha)$ *is classically entailed by* $Tr(\mathcal{S})$*, where Tr is the translation operator of Chapter 5.*

Lemma 24 is useful as it allows us to translate between the language of Herbrand interpretations (classical consequence), and the language of first-order structures (classical entailment). For instance, as a corollary we get a version of the compactness theorem for Herbrand semantics:

Corollary 2. *Suppose* α *is a classical consequence of* $\mathcal{S} \subseteq \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$*. Then there is some finite subset* $\mathcal{S}' \subseteq \mathcal{S}$ *such that* α *is a classical consequence of* $\mathcal{S}'$*.*

6.2 Rationality Postulates

In this section we introduce a formal model of the outcome of a defeasible reasoning process over Defeasible Datalog theories. As we have done before, we will model this in terms of a set of *defeasible properties* based on the KLM rationality postulates [28], which describe permissible patterns of defeasible reasoning (adapted for the case of Datalog rules and facts). For a more in-depth discussion of the model, see the discussion of the propositional case in Section 3.2, and the ALC case in Section 4.2.

In these earlier cases, the technical definition of defeasible properties is in terms of *inclusion formulas*, which are themselves certain kinds of formulas over a set of *meta-atoms* or *meta-concepts*. We will take the same approach here, but due to the expressiveness of Datalog we will need to expand our language for inclusion formulas slightly.

Firstly, Datalog compounds can contain variable symbols. In some cases we might want to express a defeasible property that states certain *substitutions* of a rule should be sanctioned. Thus we need a way of expressing *substitutions* in inclusion formulas. Note that there are a few different kinds of substitutions we might want to talk about, depending on whether the range of the substitution allows for constants, or variables, or both.

Secondly, Datalog theories are invariant under relabelling of variables. For instance, the following theories express exactly the same thing, the difference being merely a permutation of variable symbols:

$$\Gamma_+ = \{B(X,Y) \leftarrow A(X,Y)\}$$
$$\Gamma_- = \{B(Y,X) \leftarrow A(Y,X)\}$$

Thus we need a way of expressing *permutations* in inclusion formulas, which are a special case of substitutions.

The syntax of defeasible properties in our model will be based on a (heavily) restricted form of first-order logic, and we have tried to ensure that it is as simple as possible while still being expressive enough to solve the two problems mentioned above. Let $\mathcal{A} = \{A, B, \dots\}$ be a set of *meta-atom symbols*, and $\mathcal{V} = \{\vec{x}, \vec{y}, \dots\}$ a set of *meta-variable symbols*.

A *meta-argument* is defined to be anything of the form " $\vec{x}$ ", " $\varphi\vec{x}$ ", " $\pi\vec{x}$ " or " $\sigma\vec{x}$ ", where $\vec{x} \in \mathcal{V}$ is a meta-variable. A *atomic meta-term* is then defined to be anything of the form " $A(a_1, \dots, a_n)$ ", where $A \in \mathcal{A}$ is a meta-atom and $a_1, \dots, a_n$ is a (possibly empty) sequence of meta-arguments that contains each meta-variable $\vec{x} \in \mathcal{V}$ at most once. In the case that $n = 0$, we will write the meta-term as " A " rather than " $A()$ " for clarity. A *compound meta-term* is simply a boolean combination of atomic meta-terms. Meta-terms represent placeholders for Defeasible Datalog compounds, with different meta-arguments corresponding to disjoint sets of variables in the compound. We denote the set of meta-terms by $\mathcal{T}$.

An *inclusion schema* is defined to be anything of the form "t", "$t_l \leftarrow t_r$", "$t_l \rightsquigarrow t_r$", "$t_l \not\rightsquigarrow t_r$", "$\models t$", "$\models t_l \leftarrow t_r$", "$\forall\varphi.I$", "$\forall\pi.I$" or "$\forall\sigma.I$", where $t, t_l, t_r \in \mathcal{T}$ are meta-terms and I is itself an inclusion schema (i.e. this definition is recursive). We denote the set of inclusion schemas by $\mathbb{I}_{\rightsquigarrow}$, and make the restriction that inclusion schemas to be finite, as infinitely recursive rules don't seem to have a useful interpretation to us.

Example 32. *The following are all valid inclusion schemas:*

1. *"$A(\vec{x}) \vee \neg A(\vec{x})$", which intuitively states that a reasoner sanctions either the compound "$A(\vec{x})$" or its negation.*
2. *"$\forall\varphi.B(\vec{x}, \varphi\vec{y}) \rightsquigarrow A(\vec{x})$", which intuitively states that for every substitution φ, the reasoner sanctions the defeasible rule "$B(\vec{x}, \varphi(\vec{y})) \rightsquigarrow A(\vec{x})$". The symbols "$\varphi$", "$\pi$" and "$\sigma$" are intended to represent arbitrary substitutions, variable substitutions and variable permutations respectively.*

3. *"$\models B(\vec{y}) \wedge C(\vec{z}) \leftarrow A(\vec{x})$", which intuitively states that the classical rule "$B(\vec{y}) \wedge C(\vec{z}) \leftarrow A(\vec{x})$" is a tautology under the semantics given earlier in this chapter.*

On the other hand, "$A(\vec{x}, \sigma\vec{x}) \leftarrow B$" is not *a valid inclusion schema, as "$A(\vec{x}, \sigma\vec{x})$" contains the meta-variable "$\vec{x}$" twice, and is hence not a valid meta-term.*

An inclusion schema is a blueprint representing a set of Defeasible Datalog rules and facts that a reasoner may or may not sanction. Let us now formalise exactly *which* rules and facts an inclusion schema corresponds to, by providing inclusion schemas with a *logical semantics*. As we have seen in the examples, inclusion schemas like "$\forall\pi.I$" are intended to encode restricted forms of quantification, and thus our language for defeasible properties is significantly more expressive than that of propositional logic or ALC.

A *substitution system* is a tuple $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$, where $\lambda : \mathcal{V} \to 2^{\text{VAR}}$ maps meta-variables to *disjoint* sets of variables, $\tau : \mathcal{A} \to \mathcal{L}_c$ maps meta-atoms to compounds, $\varphi : \text{VAR} \to \text{VAR} \cup \text{CONST}$ is a general substitution, $\pi : \text{VAR} \to \text{VAR}$ is a variable substitution, and $\sigma : \text{VAR} \to \text{VAR}$ is a variable substitution that *permutes* VAR (i.e. it is a bijection).

Let $a_1, \ldots, a_n$ be a sequence of meta-arguments that contains each meta-variable at most once (for example, it can't contain both "$\vec{x}$" and "$\varphi\vec{x}$"). Then the *instance* of $a_1, \ldots, a_n$ with respect to the substitution system Ψ is the substitution $\Psi_{a_1,\ldots,a_n} : \text{VAR} \to \text{VAR} \cup \text{CONST}$ defined as follows:

1. $\Psi_{a_1,\ldots,a_n}(X) = \varphi(X)$ if there's an $i \in [1, n]$ s.t. $a_i =$"$\varphi\vec{x}$" and $X \in \lambda(\vec{x})$
2. $\Psi_{a_1,\ldots,a_n}(X) = \pi(X)$ if there's an $i \in [1, n]$ s.t. $a_i =$"$\pi\vec{x}$" and $X \in \lambda(\vec{x})$
3. $\Psi_{a_1,\ldots,a_n}(X) = \sigma(X)$ if there's an $i \in [1, n]$ s.t. $a_i =$"$\sigma\vec{x}$" and $X \in \lambda(\vec{x})$
4. $\Psi_{a_1,\ldots,a_n}(X) = X$ otherwise.

This may seem rather formal and unmotivated, but we will use instances of meta-arguments to define exactly how a substitution system acts on meta-terms to produce Defeasible Datalog compounds. Note that $\Psi_{a_1,\ldots,a_n}$ is well-defined, as λ is required to map meta-variables to *disjoint* sets of variable symbols, and thus for any variable symbol X precisely one of the above cases holds.

Example 33. *Consider the substitution system $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$, where τ is any map $\mathcal{A} \to \mathcal{L}_c$, and the other components are defined as follows:*

1. $\lambda(\vec{x}) = \{X, W\}$, $\lambda(\vec{y}) = \{Y\}$, $\lambda(\vec{z}) = \{Z\}$
2. $\varphi(X) = \mathsf{Jim}$, $\varphi(Y) = Y$, $\varphi(Z) = W$, $\varphi(W) = \mathsf{Giovanni}$
3. $\pi(X) = Y$, $\pi(Y) = X$, $\pi(Z) = X$, $\pi(W) = Y$

4. $\sigma(X) = Y$, $\sigma(Y) = Z$, $\sigma(Z) = X$, $\sigma(W) = W$

Now let $a =$ "$\vec{x}$", $b =$ "$\varphi\vec{x}$", $c =$ "$\pi\vec{y}$" *and* $d =$ "$\sigma\vec{z}$". *We can construct various instances of these meta-arguments with respect to* Ψ*:*

1. $\Psi_a\Big(\mathsf{friends}(X, Y, Z, W)\Big) = \mathsf{friends}(X, Y, Z, W)$
2. $\Psi_b\Big(\mathsf{friends}(X, Y, Z, W)\Big) = \mathsf{friends}(\mathsf{Jim}, Y, Z, \mathsf{Giovanni})$
3. $\Psi_c\Big(\mathsf{friends}(X, Y, Z, W)\Big) = \mathsf{friends}(X, X, Z, W)$
4. $\Psi_d\Big(\mathsf{friends}(X, Y, Z, W)\Big) = \mathsf{friends}(X, Y, X, W)$
5. $\Psi_{b,c,d}\Big(\mathsf{friends}(X, Y, Z, W)\Big) = \mathsf{friends}(\mathsf{Jim}, X, X, \mathsf{Giovanni})$

Note that an instance like $\Psi_{a,b,c,d}$ *is undefined, as "a" and "b" both contain the meta-variable* $\vec{x}$*.*

Consider now some atomic meta-term $t =$"$A(a_1, \ldots, a_n)$". The idea with the meta-arguments $a_1, \ldots, a_n$ is to constrain the possible compounds that can be substituted for A. Each of the meta-arguments corresponds to a disjoint set of variable symbols, and we want to require that the compound substituted for A uses *only* these variable symbols. With this in mind, we say the substitution system Ψ is *compatible* with t iff every variable symbol in the compound $\tau(A)$ is contained in $\lambda(\vec{x})$ for some meta-variable $\vec{x}$ in the meta-arguments $a_1, \ldots, a_n$.

Example 34. *Let* t *be the meta-term "*$A(\vec{x}, \varphi\vec{y}, \vec{z})$*", and define* $\tau : \mathcal{A} \to \mathcal{L}_c$ *by* $\tau(A) = \mathsf{triplets}(X, Y, Z)$*. Then consider the following two maps from meta-variables to set of variable symbols:*

1. $\lambda_1(\vec{x}) = \{X, Y\}$, $\lambda_1(\vec{y}) = \{Z\}$, $\lambda_1(\vec{z}) = \{W\}$
2. $\lambda_2(\vec{x}) = \{Y\}$, $\lambda_2(\vec{y}) = \{Z\}$, $\lambda_2(\vec{z}) = \{W\}$

The substitution system $\Psi_1 = \langle\lambda_1, \tau, id, id, id\rangle$ *is compatible with* t*, as* $\tau(A)$ *contains the variable symbols* X*,* Y *and* Z*, and these are contained in* $\lambda_1(\vec{x})$*,* $\lambda_1(\vec{x})$ *and* $\lambda_1(\vec{y})$ *respectively.*

On the other hand, the substitution system $\Psi_2 = \langle\lambda_2, \tau, id, id, id\rangle$ *is* not *compatible with* t*, as* $\tau(A)$ *contains the variable symbol* X*, which is not contained in* $\lambda_2(\vec{x})$*,* $\lambda_2(\vec{y})$ *or* $\lambda_2(\vec{z})$*.*

We define the *instance* of the atomic meta-term $t =$"$A(a_1, \ldots, a_n)$" with respect to a compatible substitution system Ψ to be the following compound:

$$\Psi(t) = \Psi_{a_1,\ldots,a_n}\Big(\tau(A)\Big)$$

Once again, we note that $\Psi_{a_1,\ldots,a_n}$ is well-defined, as meta-terms are required to contain each meta-variable $\vec{x} \in \mathcal{V}$ at most once. The instance of a *compound* meta-term $t \in \mathcal{T}$ with respect to Ψ is defined to be the compound that results from replacing each atomic meta-term in t with it's corresponding instance.

Example 35. *Consider the substitution system $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$, where the components are defined as follows:*

1. $\tau(A) = \mathsf{square}(X, Y, Z, W)$
2. $\lambda(\vec{x}) = \{X, Y\}$, $\lambda(\vec{y}) = \{Z\}$, $\lambda(\vec{z}) = \{W\}$
3. $\varphi(X) = \mathsf{Bob}$, $\varphi(Y) = Y$, $\varphi(Z) = W$, $\varphi(W) = \mathsf{Alice}$
4. $\pi(X) = Y$, $\pi(Y) = X$, $\pi(Z) = X$, $\pi(W) = W$
5. $\sigma(X) = Y$, $\sigma(Y) = Z$, $\sigma(Z) = W$, $\sigma(W) = X$

Then the meta-term $t =$ "$A(\varphi\vec{x}, \pi\vec{y}, \sigma\vec{z})$" is compatible with respect to Ψ, and has the following instance:

$$\begin{aligned}\Psi(t) &= \Psi_{\varphi\vec{x},\pi\vec{y},\sigma\vec{z}}\Big(\tau(A)\Big) \\ &= \Psi_{\varphi\vec{x},\pi\vec{y},\sigma\vec{z}}\Big(\mathsf{square}(X, Y, Z, W)\Big) \\ &= \mathsf{square}(\mathsf{Bob}, Y, X, X)\end{aligned}$$

We are now in a position to interpret inclusion schemas using sets of Defeasible Datalog formulas (this is the *logical semantics* we alluded to earlier). Given a set of Defeasible Datalog formulas $\mathcal{S} \subseteq \mathcal{L}$, we say $\mathcal{S}$ *satisfies* an inclusion schema $I \in \mathbb{I}_{\rightsquigarrow}$ under the substitution system $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$ iff Ψ is compatible with every meta-term contained in I, and the following conditions hold:

1. If $I =$ "t", then $\varphi \circ \Psi(t) \in \mathcal{S}$ for every ground substitution φ.
2. If $I =$ "$t_l \leftarrow t_r$", then $\Psi(t_l) \leftarrow \Psi(t_r) \in \mathcal{S}$.
3. If $I =$ "$t_l \rightsquigarrow t_r$", then $\Psi(t_l) \rightsquigarrow \Psi(t_r) \in \mathcal{S}$.
4. If $I =$ "$t_l \not\rightsquigarrow t_r$", then $\Psi(t_l) \rightsquigarrow \Psi(t_r) \notin \mathcal{S}$.
5. If $I =$ "$\models t$", then $\Psi(t)$ is a classical tautology.
6. If $I =$ "$\models t_l \leftarrow t_r$", then $\Psi(t_l \leftarrow t_r)$ is a classical tautology.
7. If $I =$ "$\forall\varphi.I^*$", then for every substitution $\varphi' : \textsc{var} \to \textsc{var} \cup \textsc{const}$, $\mathcal{S}$ satisfies I^* under $\Psi' = \langle \lambda, \tau, \varphi', \pi, \sigma \rangle$.
8. If $I =$ "$\forall\pi.I^*$", then for every substitution $\pi' : \textsc{var} \to \textsc{var}$, $\mathcal{S}$ satisfies I^* under $\Psi' = \langle \lambda, \tau, \varphi, \pi', \sigma \rangle$.

9. If $I =$ "$\forall\sigma.I^*$", then for every permutation $\sigma' : \text{VAR} \to \text{VAR}$, $\mathcal{S}$ satisfies I^* under $\Psi' = \langle\lambda, \tau, \varphi, \pi, \sigma'\rangle$.

Note that the conditions for quantified substitutions are similar in spirit to the satisfaction conditions for quantifiers in first-order logic (see Chapter 2, Section 2.3). We denote the fact that $\mathcal{S}$ satisfies the inclusion schema I under Ψ by $\mathcal{S} \Vdash_\Psi I$.

Example 36. *Consider the inclusion schemas* $I_1 =$"$B(\vec{x}, \varphi\vec{y}) \rightsquigarrow A(\vec{x})$" *and* $I_2 =$"$\forall\varphi.B(\vec{x}, \varphi\vec{y}) \rightsquigarrow A(\vec{x})$", *and let* $\mathcal{S} = \{\neg\mathsf{likes}(\mathsf{Alice}, \mathsf{Bob}) \rightsquigarrow \mathsf{unhappy}(\mathsf{Alice})\}$. *To illustrate our definition of satisfaction, we construct a substitution system* $\Psi = \langle\lambda, \tau, \varphi, \pi, \sigma\rangle$ *as follows:*

1. $\lambda(\vec{x}) = \{X\}$, $\lambda(\vec{y}) = \{Y\}$
2. $\tau(A) = \mathsf{unhappy}(\mathsf{Alice})$, $\tau(B) = \neg\mathsf{likes}(\mathsf{Alice}, Y)$
3. $\varphi(X) = X$, $\varphi(Y) = \mathsf{Bob}$
4. $\pi = \sigma = id$

The system Ψ *is compatible with every meta-term in* I_1 *and* I_2, *and by definition* $\mathcal{S}$ *satisfies* I_1 *under* Ψ *iff* $\mathcal{S}$ *contains the following formula:*

$$\begin{aligned}\Psi\Big(B(\vec{x}, \varphi\vec{y})\Big) \rightsquigarrow \Psi\Big(A(\vec{x})\Big) &= \Psi_{\vec{x},\varphi\vec{y}}\Big(\neg\mathsf{likes}(\mathsf{Alice}, Y)\Big) \rightsquigarrow \Psi_{\vec{x}}\Big(\mathsf{unhappy}(\mathsf{Alice})\Big) \\ &= \neg\mathsf{likes}(\mathsf{Alice}, \mathsf{Bob}) \rightsquigarrow \mathsf{unhappy}(\mathsf{Alice})\end{aligned}$$

Thus $\mathcal{S} \Vdash_\Psi I_1$. *On the other hand,* $\mathcal{S}$ *does* not *satisfy* I_2 *under* Ψ. *To show this, it suffices to construct a substitution* $\varphi' : \text{VAR} \to \text{VAR} \cup \text{CONST}$ *such that* $\mathcal{S}$ *fails to satisfy* $B(\vec{x}, \varphi\vec{y}) \rightsquigarrow A(\vec{x})$ *under* $\Psi' = \langle\lambda, \tau, \varphi', \pi, \sigma\rangle$. *One such choice of* φ' *is the following:*

1. $\varphi'(X) = X$, $\varphi'(Y) = \mathsf{Alice}$

In general, satisfying a quantified inclusion schema may require the set of formulas $\mathcal{S}$ *to be infinite. For this reason we will avoid writing such sets out explicitly, and rather construct them with mathematical tools developed later in the chapter.*

We define a *defeasible property* to be anything of the form "$I_1, \ldots, I_n \implies I_{n+1}$", where the $I_i \in \mathbb{I}_\rightsquigarrow$ are all inclusion schemas, satisfying the following consistency requirement:

1. Whenever two meta-terms $t_1, t_2 \in \mathcal{T}$ in the defeasible property contain the same meta-atom, they must also contain exactly the same meta-variables in exactly the same order.

We denote the set of defeasible properties by $\mathbb{P}_{\rightsquigarrow}$, and in the case that P is a defeasible property with $n = 0$, we will write it as "I_1" rather than the technically correct "$\implies I_1$". The point of the consistency requirement is simply to disallow certain pathological defeasible properties:

Example 37. *Consider the inconsistent defeasible property $A(\vec{x}) \nsim A(\vec{y})$. Then from the definitions it follows that a substitution system $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$ is compatible with both of the meta-terms $A(\vec{x})$ and $A(\vec{y})$ iff $\tau(A)$ is a ground compound, as $\lambda(\vec{x})$ and $\lambda(\vec{y})$ are required to be disjoint sets of variables. Since this can already be expressed by the simpler defeasible property $A \nsim A$, we see no reason to allow such complications.*

As we've seen before in the propositional and ALC cases, a defeasible property represents a particular pattern of defeasible reasoning. It states that whenever a reasoner satisfies the schemas $I_1, \ldots, I_n$ under some substitution system, the reasoner should also sanction the schema I_{n+1}. This can be formalised as follows, where we say that a substitution system Ψ is *compatible* with a defeasible property P iff it is compatible with every meta-term in P:

Definition 25. *Let $P =$"$I_1, \ldots, I_n \implies I_{n+1}$" be a defeasible property, and suppose $\mathcal{S} \subseteq \mathcal{L}$ is a set of Defeasible Datalog formulas. Then $\mathcal{S}$ is said to be* P-complete *iff for every compatible substitution system Ψ such that $\mathcal{S} \Vdash_\Psi I_1, \ldots, I_n$, we have that $\mathcal{S} \Vdash_\Psi I_{n+1}$.*

Similarly, if $D \subseteq \mathbb{P}_{\rightsquigarrow}$ is a *set* of defeasible properties, we say $\mathcal{S}$ is D-complete iff it is complete for every property in the set. The notion of a defeasible property allows us to express Defeasible Datalog analogues of the KLM rationality postulates. These are based on the postulates given by Morris et al. [28], and have simply been translated into our formalism:

(REFL) $A(\vec{x}) \nsim A(\vec{x})$

(LLE) $C(\vec{x}) \nsim A(\vec{x}),\ \models A(\vec{x}) \leftarrow B(\vec{x}),\ \models B(\vec{x}) \leftarrow A(\vec{x}) \implies C(\vec{x}) \nsim B(\vec{x})$

(RW) $B(\vec{x}) \nsim A(\vec{x}),\ \models C(\vec{x}) \leftarrow B(\vec{x}) \implies C(\vec{x}) \nsim A(\vec{x})$

(OR) $C(\vec{x}) \nsim A(\vec{x}),\ C(\vec{x}) \nsim B(\vec{x}) \implies C(\vec{x}) \nsim A(\vec{x}) \vee B(\vec{x})$

(AND) $B(\vec{x}) \nsim A(\vec{x}),\ C(\vec{x}) \nsim A(\vec{x}) \implies B(\vec{x}) \wedge C(\vec{x}) \nsim A(\vec{x})$

(RM) $C(\vec{x}) \nsim A(\vec{x}),\ \neg B(\vec{x}) \not\nsim A(\vec{x}) \implies C(\vec{x}) \nsim A(\vec{x}) \wedge B(\vec{x})$

These properties say exactly the same thing as the propositional and ALC rationality postulates, the only difference being that they are generalised to formulas of arbitrary arity. The propositional rationality postulates apply to propositional formulas, which are analogous to ground facts, and the ALC rationality postulates apply to concepts, which are analogous to unary predicates. We are thus taking one more step up the ladder of generalisation.

Definition 26. *A set of formulas $\mathcal{S} \subseteq \mathcal{L}$ is said to be* rational *iff it is complete for the properties* (REFL)-(RM)*. We refer to these properties as the* Defeasible Datalog rationality postulates*, and denote them by $\mathbb{R}_{\rightsquigarrow}$.*

In fact, not only is $\mathbb{R}_{\rightsquigarrow}$ a *syntactic* generalisation of the propositional or ALC rationality postulates, but it also has the same *consequences*, in the following sense:

Definition 27. *Let $D \subseteq \mathbb{P}_{\rightsquigarrow}$ be a set of defeasible properties. Then we say a property $P \in \mathbb{P}_{\rightsquigarrow}$ is a* consequence *of D iff every D-complete set of formulas is P-complete as well.*

6.3 Consequences of the Rationality Postulates

Our goal in this section is to prove that every consequence of the *propositional* rationality postulates corresponds to a consequence of the *Defeasible Datalog* rationality postulates. This provides us with formal evidence that they express the same thing, and also provides us with a convenient tool for deriving *new* properties that hold for all rational sets of Defeasible Datalog formulas. Proofs for all lemmas and theorems are given in Appendix B.

The main technical ingredient we need is the construction of a *translation operator* that takes propositional defeasible properties and converts them into Defeasible Datalog properties. We recall from Section 3.2 that propositional defeasible properties are described in terms of a set $\tilde{\mathcal{P}}$ of *propositional meta-atoms*.

Definition 28. *A* translation base *is defined to be an injective map $B : \tilde{\mathcal{P}} \to \mathcal{A}$ which assigns a unique Defeasible Datalog meta-atom to each propositional meta-atom.*

Given a fixed translation base B, we can extend it uniquely to a mapping $B_* : \mathcal{L}^{\tilde{\mathcal{P}}} \to \mathcal{T}$ as follows:

1. If $p \in \tilde{\mathcal{P}}$ is atomic, then $B_*(p) = B(p)(\vec{x})$.
2. $B_*(\alpha \wedge \beta) = B_*(\alpha) \wedge B_*(\beta)$
3. $B_*(\neg\alpha) = \neg B_*(\alpha)$

In fact, we can go a little further extend B_* to a *translation operator* $\mathrm{Tr}_B : \mathbb{I}_{\mid\sim} \to \mathbb{I}_{\rightsquigarrow}$, between propositional inclusion formulas and Defeasible Datalog inclusion schemas:

1. $\mathrm{Tr}_B(\alpha \mid\sim \beta) = \ B_*(\beta) \leftsquigarrow B_*(\alpha)$
2. $\mathrm{Tr}_B(\alpha \not\mid\sim \beta) = \ B_*(\beta) \not\leftsquigarrow B_*(\alpha)$
3. $\mathrm{Tr}_B(\models \alpha) = \ \models B_*(\alpha)$

This allows us to extend translation operators to defeasible properties in the obvious way:

Definition 29. *Let Tr_B be a translation operator and $P = I_1, \ldots, I_n \implies I_{n+1}$ some propositional defeasible property. Then we define $Tr_B(P)$ to be the Defeasible Datalog property $Tr_B(I_1), \ldots, Tr_B(I_n) \implies Tr_B(I_{n+1})$.*

A nice property of translation operators is that they are all essentially equivalent, as different translation bases can be mapped onto one another by a permutation of meta-atom symbols. Thus, while we won't do this explicitly, we are always free to choose a *convenient* translation base. This observation can be formalised as follows:

Lemma 25. *Let B, B' be two translation bases. Then for every propositional defeasible property $P \in \mathbb{P}_{\mid\sim}$, $Tr_B(P)$ is a consequence of $Tr_{B'}(P)$ and vice-versa.*

By translating between Defeasible Datalog properties and propositional properties, we are able to prove the following correspondence theorem:

Theorem 7. *Suppose that $D \subseteq \mathbb{P}_{\mid\sim}$ is a set of propositional defeasible properties, and that $P \in \mathbb{P}_{\mid\sim}$ is a consequence of D. Then for any translation base B, $Tr_B(P)$ is a consequence of $Tr_B(D)$.*

This has the following nice corollary:

Corollary 3. *Let B be a translation base. Then if $P \in \mathbb{P}_{\mid\sim}$ is a consequence of the propositional rationality postulates, $Tr_B(P)$ is a consequence of the Defeasible Datalog rationality postulates.*

These results give us a number of nice consequences of the Defeasible Datalog rationality postulates for free, which we will make use of later when we discuss semantics:

Lemma 26. *$\mathbb{R}_{\rightsquigarrow}$ has as consequence the following defeasible properties:*

1. $\bot \not\sim A(\vec{x}) \implies \neg A(\vec{x}) \not\sim A(\vec{x})$
2. $\neg A(\vec{x}) \not\sim A(\vec{x}) \vee B(\vec{x}),\ \neg B(\vec{x}) \not\sim B(\vec{x}) \vee C(\vec{x}) \implies \neg A(\vec{x}) \not\sim A(\vec{x}) \vee C(\vec{x})$
3. $\bot \not\sim A(\vec{x}),\ \neg A(\vec{x}) \rightsquigarrow A(\vec{x}) \vee B(\vec{x}) \implies \neg B(\vec{x}) \not\sim A(\vec{x}) \vee B(\vec{x})$

6.4 Beyond the Rationality Postulates

So far we have discussed analogues of the propositional rationality postulates for Defeasible Datalog, and shown (by Corollary 3) that they faithfully reproduce the same defeasible properties, which is strong evidence that they express the same intuition. We have also introduced the notion of a *rational set* of Defeasible Datalog formulas, with the intention of using such sets as a model of the outcome of a defeasible reasoning process over Defeasible Datalog knowledge bases.

On the other hand, a little thought makes it clear that the rationality postulates leave a lot to be desired in such a process! For instance, it is possible to construct a rational set $\mathcal{S} \subseteq \mathcal{L}$ that contains both $\mathsf{fly}(X) \mathrel{\sim\!\!\!\!\!\leftarrow} \mathsf{bird}(X)$ and $\neg\mathsf{fly}(Y) \mathrel{\sim\!\!\!\!\!\leftarrow} \mathsf{bird}(Y)$, and which contains neither $\bot \mathrel{\sim\!\!\!\!\!\leftarrow} \mathsf{bird}(X)$ nor $\bot \mathrel{\sim\!\!\!\!\!\leftarrow} \mathsf{bird}(Y)$. Such a set appears to state two contradictory things - that birds typically fly, and that birds typically don't fly. While we won't go into details here, pathological rational sets like this can be constructed using the semantics in Lehmann et al. [24], for instance.

The reason such pathologies exist is that we haven't enforced any other constraints on rational sets, such as the requirement that rational sets should be invariant under relabelling of variables. This can be formalised as a defeasible property in the following way:

$$(\textsc{Per}) \quad B(\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\vec{x}) \implies B(\sigma\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\sigma\vec{x})$$

Intuitively, this states that whenever a reasoner sanctions a defeasible rule, it should also sanction any defeasible rule that differs only by a permutation of variable symbols (recall that "σ" is the symbol for *variable permutations* in the formalism of Section 6.2). And indeed, we can prove the following lemma directly from the definitions:

Lemma 27. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a* (Per)*-complete set of formulas, and suppose that $B \mathrel{\sim\!\!\!\!\!\leftarrow} A \in \mathcal{S}$. Then for any permutation $\sigma : \textsc{var} \to \textsc{var}$, we have that $\sigma(B) \mathrel{\sim\!\!\!\!\!\leftarrow} \sigma(A) \in \mathcal{S}$.*

Note that proofs of all lemmas in this section can be found in Appendix C. In a similar vein, consider a set $\mathcal{S} \subseteq \mathcal{L}$ containing the following defeasible rule:

$$\mathsf{contentWith}(X, Y) \mathrel{\sim\!\!\!\!\!\leftarrow} \mathsf{enlightened}(X)$$

This states something along the lines of "enlightened individuals are typically content with anything". Note that the classical Datalog version of this rule (i.e. replacing " $\mathrel{\sim\!\!\!\!\!\leftarrow}$ " with " $\leftarrow$ ") would be considered *unsafe*, as the variable symbol Y is not bound by any predicate in the body of the rule. Thus, if the extensional database contained *any* fact of the form $\mathsf{enlightened}(a)$, we would be able to infer the (generally infinite) set of all possible ground instances of $\mathsf{contentWith}(a, Y)$. In other words, as a *classical, unsafe rule*, we would be able to infer every possible instance $\mathsf{contentWith}(X, b) \leftarrow \mathsf{enlightened}(X)$ as a consequence. Indeed, the reason unsafe rules are banned in classical Datalog is precisely because they can lead to programs with an infinite set of consequences.

As discussed at the start of this chapter, we make no safety restrictions on Defeasible Datalog rules. Nevertheless, it seems reasonable to keep the intuition that unbound variables can be substituted for any other term. For a pithy example, if a reasoner sanctions "enlightened individuals are typically content with anything", it also seems reasonable to conclude that the reasoner sanctions "enlightened individuals are typically content with suffering". This substitution

principle can be formalised as the following defeasible property:

$$(\textsc{Irr})\quad B(\vec{x},\vec{y}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x}) \implies B(\vec{x},\varphi\vec{y}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x})$$

To understand why this formalisation is correct, recall that in the formalism of Section 6.2 different meta-variable symbols $\vec{x}$ and $\vec{y}$ are interpreted by *disjoint* sets of variables, and that "φ" is the symbol for *general substitutions*.

Lemma 28. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a* (Irr)*-complete set of formulas, and suppose that $B \mathrel{\sim\!\!\!\!\!\!\leftarrow} A \in \mathcal{S}$, with $\textsc{var}_A$ and $\textsc{var}_B$ the free variables in A and B respectively. Then for any substitution $\varphi : \textsc{var} \to \textsc{var} \cup \textsc{const}$ that is constant on $\textsc{var}_A$, we have that $\varphi(B) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A \in \mathcal{S}$.*

Let us also mention the following defeasible property, which is a slight weakening of (Irr). Whereas (Irr) deals with *arbitrary* substitutions, the following property only allows variable substitutions (recall that "π" is the symbol for *variable substitutions* in the formalism of Section 6.2):

$$(\textsc{WkIrr})\quad B(\vec{x},\vec{y}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x}) \implies B(\vec{x},\pi\vec{y}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x})$$

Lemma 29. (WkIrr) *is a consequence of* (Irr)*.*

We mention this weaker property as an aside mainly for theoretical reasons. In Section 6.6 we prove a representation result showing that the rationality postulates, along with number of additional defeasible properties from the present section, can be completely characterised by a semantic structure based on the preferential semantics of Kraus et al. [22]. Unfortunately, we were unable to characterise (Irr), and had to settle for the weaker property (WkIrr) instead. Fixing this deficiency would be an interesting avenue for future research.

The last property to do with substitutions we discuss involves the concept of *universal substitution*. In classical Datalog, a rule such as $\mathsf{animal}(X) \leftarrow \mathsf{dog}(X)$ *always* has as a consequence any groundings of the rule, such as:

$$\mathsf{animal}(\mathsf{Fido}) \leftarrow \mathsf{dog}(\mathsf{Fido})$$
$$\mathsf{animal}(\mathsf{Foobar}) \leftarrow \mathsf{dog}(\mathsf{Foobar})$$
$$\ldots$$

This makes sense, as classical Datalog rules are implicitly universally quantified. Thus a natural question to ask is whether a similar principle should hold for *defeasible* rules. Or in other words, is the following defeasible property reasonable?

$$(\textsc{UI})\quad B(\vec{x}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x}) \implies B(\varphi\vec{x}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\varphi\vec{x})$$

We might also wish to consider a slightly weaker version of the principle, in which we permit arbitrary *variable* substitutions of defeasible rules:

$$(\textsc{WkUI})\quad B(\vec{x}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\vec{x}) \implies B(\pi\vec{x}) \mathrel{\sim\!\!\!\!\!\!\leftarrow} A(\pi\vec{x})$$

Note that both of these "universal instantiation" properties are stronger versions of the (Per) property, which asserts the same thing but for *permutations* of variables:

Lemma 30. (Per) *is a consequence of* (WkUI)*, which is in turn a consequence of* (UI)*.*

Unlike the (Per) property, however, we feel that neither (UI) nor (WkUI) is a principle that should hold generally for a defeasible reasoning process. To explain why, we turn to an example originally due to Delgrande [14]. Consider the following set of Defeasible Datalog formulas:

$$\begin{aligned}\mathcal{S} = \{\ &\mathsf{feeds}(X, Y) \mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(X) \wedge \mathsf{elephant}(Y),\\ &\neg\mathsf{feeds}(\mathsf{Fred}, Y) \mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(Y),\\ &\mathsf{feeds}(\mathsf{Fred}, \mathsf{Clyde}) \mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(\mathsf{Clyde})\ \}\end{aligned}$$

Intuitively, this states that keepers typically feed elephants, that the keeper Fred typically *doesn't* feed elephants, and that the keeper Fred typically *does* feed the elephant Clyde. Note that there doesn't appear to be anything inherently contradictory in these statements. After all, Fred might simply have a soft spot for Clyde, even though his daily duties involve feeding the penguins rather than the elephants.

Now suppose we extend $\mathcal{S}$ to a rational, (UI)-complete set of formulas $\mathcal{S}'$, which would represent the outcome of some hypothetical defeasible reasoning process over $\mathcal{S}$. Then by (UI)-completeness we infer that $\mathcal{S}'$ must contain every grounding of every defeasible rule in $\mathcal{S}$. In particular, $\mathcal{S}'$ contains both of the following defeasible rules:

$$\begin{aligned}\mathsf{feeds}(\mathsf{Fred}, \mathsf{Clyde}) &\mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(\mathsf{Clyde})\\ \neg\mathsf{feeds}(\mathsf{Fred}, \mathsf{Clyde}) &\mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(\mathsf{Clyde})\end{aligned}$$

The first of these rules comes directly from $\mathcal{S}$, and the other from applying (UI) to the second rule in $\mathcal{S}$ under the substitution $Y \mapsto \mathsf{Clyde}$. But these two rules are mutually contradictory! Since $\mathcal{S}'$ is assumed to be rational, an application of (And) and (Rw) allows us to conclude that $\mathcal{S}'$ must contain the following rule:

$$\bot \mathrel{\sim\!\!\!\leftarrow} \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(\mathsf{Clyde})$$

This states that, if Fred is a keeper and Clyde is an elephant, then they typically don't exist. This is an unintuitive conclusion to draw from $\mathcal{S}$, to say the least! Given that the rationality postulates are widely accepted in the literature, it seems that the only logical thing to do here is to reject (UI) as a candidate property. How about the somewhat weaker property (WkUI), then? Unfortunately, this also fails for similar reasons. Consider the following set of Defeasible Datalog formulas:

$$\begin{aligned}\mathcal{S} = \{\ &\mathsf{eats}(X, Y) \mathrel{\sim\!\!\!\leftarrow} \mathsf{predator}(X) \wedge \mathsf{prey}(Y),\\ &\neg\mathsf{eats}(X, X) \mathrel{\sim\!\!\!\leftarrow} \mathsf{predator}(X) \wedge \mathsf{prey}(X)\ \}\end{aligned}$$

This states that predators typically eat prey, and that things which are both predators *and* prey typically don't eat themselves. While somewhat artificial, this example also doesn't seem to contain an inherent contradiction. However, if we follow the lead of the previous example and extend $\mathcal{S}$ to a rational, (WkUI)-complete set of formulas $\mathcal{S}'$, we find that $\mathcal{S}'$ necessarily contains the following formula:

$$\bot \mathrel{\sim\!\!\!\!\leftarrow} \mathsf{predator}(X) \wedge \mathsf{prey}(X)$$

In other words, there typically doesn't exist anything that is both a predator and prey. This again seems like an unintuitive conclusion to draw from $\mathcal{S}$, and thus we are forced to reject (WkUI) as a candidate property as well.

Let us now turn to a different class of defeasible properties, namely those that relate to the interplay between classical and defeasible formulas. First of all, consider the standard interpretation of a classical Datalog rule. It states that if an instance of the body holds, then the corresponding instance of the head *always* holds as well. Thus it certainly *usually* holds, which makes the following property seem quite reasonable:

$$\text{(Sup)} \quad B(\vec{x}) \leftarrow A(\vec{x}) \implies B(\vec{x}) \mathrel{\sim\!\!\!\!\leftarrow} A(\vec{x})$$

This property is commonly called *supraclassicality* in the literature, and because we separate rules and facts in Datalog we will also consider the following *fact-based* version of (Sup):

$$\text{(SupF)} \quad A(\vec{x}) \implies \bot \mathrel{\sim\!\!\!\!\leftarrow} \neg A(\vec{x})$$

In the case of propositional logic, a converse to the (SupF) property is valid. We recall Lemma 7, which states that the propositional formula "α" is in fact *equivalent* to the defeasible propositional formula "$\neg\alpha \mathrel{|\!\sim} \bot$". We might imagine that a converse also holds for Defeasible Datalog, which we could formalise as follows:

$$\text{(Neg)} \quad \bot \mathrel{\sim\!\!\!\!\leftarrow} \neg A(\vec{x}) \implies A(\vec{x})$$

This property is related to the following principle, where a "typical instance of $A(\vec{x})$" is anything satisfying both $A(\vec{x})$ and all of its defeasible consequences. For instance, a "typical instance of $\mathsf{bird}(X)$" is anything that is a bird, can fly, builds nests etc:

If it's possible that some *instance of* $A(\vec{x})$ *exists, then it's possible that a* typical *instance of* $A(\vec{x})$ *exists.*

To see why these things are related, suppose that a reasoner sanctions both the defeasible rule "$\bot \mathrel{\sim\!\!\!\!\leftarrow} \neg A(\vec{x})$" and the principle above. This rule essentially states that there are *no* typical instances of $\neg A(\vec{x})$, as such an instance would have to satisfy the falsehood $\bot$. But then, by the above principle, we conclude

that there can be no instances of $\neg A(\vec{x})$ whatsoever. In other words, the reasoner satisfies the (NEG) property!

Whether or not this principle should be adopted is a matter of how one wants to interpret the defeasible implication symbol " $\hookleftarrow$ ". For instance, consider the following two possible interpretations of the defeasible rule "$B(\vec{x}) \hookleftarrow A(\vec{x})$":

1. "the most typical instances of $A(\vec{x})$ all satisfy $B(\vec{x})$"
2. "an instance of $A(\vec{x})$ typically satisfies $B(\vec{x})$"

The first case is similar to the intuition underlying defeasible ALC statements, and in this case it *does* seem reasonable to adopt the principle, as if there exists *some* instance of $A(\vec{x})$ then there has to be a *most typical* such instance. And indeed, a version of (NEG) does hold in defeasible ALC. In the second case, on the other hand, it's possible that every instance of $A(\vec{x})$ is typical in some ways and atypical in others, and thus the principle seems less justified.

Somewhere in between these two intuitions lies the following weaker version of the principle:

$$\text{(WKNEG)} \quad \forall\varphi.\bot \hookleftarrow \neg A(\varphi\vec{x}) \implies A(\vec{x})$$

Lemma 31. (WKNEG) *is a consequence of* (NEG).

The (WKNEG) principle captures the idea that simply knowing that every instance of a concept is atypical is not enough to conclude that the concept is inconsistent. One has to know that every instance of a concept is atypical with respect to every *specialisation* of the concept too. For instance, if one knows that there are no typical birds, no typical red birds, no typical birds with broken wings, and so forth, *only then* can one conclude that there are no birds at all. This is a significant weakening of (NEG), and feels much more reasonable to us.

Finally, the last properties we will explicitly discuss are those relating only to classical rules. A very reasonable requirement of a defeasible reasoning process is that it respects all valid classical inferences, wherever it can make them. Or, to put it differently, if the reasoner believes that a particular inference is *logically necessary*, then the reasoner should make that inference!

This is essentially saying that reasoners should sanction classical Modus Ponens. Because of the nature of Datalog syntax, in which rules are differentiated from facts, we also have to insist on properties that identify classical rules with their equivalent factual forms:

$$\begin{aligned}
\text{(MP)} \quad & A(\varphi\vec{x}),\ \models B(\vec{x}) \leftarrow A(\vec{x}) \implies B(\varphi\vec{x}) \\
\text{(EQ)} \quad & B(\vec{x}) \leftarrow A(\vec{x}) \implies B(\vec{x}) \vee \neg A(\vec{x}) \\
\text{(EQF)} \quad & B(\vec{x}) \vee \neg A(\vec{x}) \implies B(\vec{x}) \leftarrow A(\vec{x})
\end{aligned}$$

Lemma 32. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a set of Defeasible Datalog formulas, and let $\mathcal{S}_C$ denote its classical subset. Then $\mathcal{S}$ is complete for* (Mp), (Eq) *and* (EqF) *iff $\mathcal{S}_C$ is deductively closed.*

This concludes our discussion of defeasible properties in generality. We have certainly not covered such properties exhaustively, and in many places our analysis is no doubt incomplete. What we have so far, however, is enough for us to start looking at what Defeasible Datalog formulas actually *mean*. In the next few sections we will look at two possibilities for *interpreting* Defeasible Datalog formulas, and discuss their relative benefits and shortcomings.

6.5 Preferential Semantics

We start our discussion by presenting a semantics that directly generalises the ranked interpretations of Chapters 3. The basic idea is to take the *propositional* intuition for defeasible formulas in terms of a preference relation on possible worlds.

Recall from Section 6.1 that a *Herbrand interpretation* is a subset $\mathcal{H} \subseteq \mathbb{B}$ of the Herbrand base, defined over a fixed first-order language, and that we can interpret the classical fragment of Defeasible Datalog using such interpretations. If we denote the set of Herbrand interpretations by $\mathbb{H}$, we can define an analogue of a propositional *ranked interpretation* (see Definition 7) as follows:

Definition 30. *A* ranked interpretation *$\mathcal{R}$ is a function $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ satisfying the following convexity condition:*

$$\begin{array}{c}\textit{For all } \mathcal{H} \in \mathbb{H} \textit{ such that } \mathcal{R}(\mathcal{H}) < \infty\textit{, and all } i \in \mathbb{N} \textit{ such that} \\ 0 \leq i < \mathcal{R}(\mathcal{H})\textit{, there exists some } \mathcal{H}' \in \mathbb{H} \textit{ such that } \mathcal{R}(\mathcal{H}') = i.\end{array}$$

A ranked interpretation is thus nothing more than a particular kind of preference ranking on Herbrand interpretations, with lower ranks denoting interpretations that the reasoner considers more *typical*, and infinite ranks denoting interpretations that the reasoner considers *impossible*. A classical formula, which is a rule or fact that is supposed to hold unconditionally, would be sanctioned by the reasoner if it holds in all the *possible* interpretations. So far, this is identical to the propositional case.

What about defeasible formulas? The approach we consider is to interpret a defeasible formula such as $B(\vec{X}) \sim A(\vec{X})$ as follows:

"In the most typical worlds containing A-instances,
all the A-instances are B-instances."

This is a direct translation of the way propositional defeasible formulas are interpreted, and we formalise it in the following way. Let $\mathbb{H}^{\mathcal{R}} = \{\mathcal{H} \in \mathbb{H} :$

$\mathcal{R}(\mathcal{H}) < \infty\}$ denote the set of possible interpretations, and for any compound $A(\vec{X}) \in \mathcal{L}_c$, let $\widehat{A} = \{\mathcal{H} \in \mathbb{H}^{\mathcal{R}} : \mathcal{H} \Vdash A(\varphi(\vec{X}))$ for some $\varphi : \text{VAR} \to \text{CONST}\}$ denote the set of possible interpretations that satisfy some ground instance of $A(\vec{X})$. Then satisfaction for the ranked interpretation $\mathcal{R}$, denoted as usual with the symbol "$\Vdash$", is defined as follows:

1. For a fact $A(\vec{c}) \in \mathcal{L}_f$, $\mathcal{R} \Vdash A(\vec{c})$ iff $\mathcal{H} \Vdash A(\vec{c})$ for every $\mathcal{H} \in \mathbb{H}^{\mathcal{R}}$.
2. For a rule $B(\vec{X}) \leftarrow A(\vec{X})$, $\mathcal{R} \Vdash B(\vec{X}) \leftarrow A(\vec{X})$ iff $\mathcal{H} \Vdash B(\vec{X}) \leftarrow A(\vec{X})$ for every $\mathcal{H} \in \mathbb{H}^{\mathcal{R}}$.
3. For a defeasible rule $B(\vec{X}) \mathrel{\sim\!\!\!\leftarrow} A(\vec{X})$, $\mathcal{R} \Vdash B(\vec{X}) \mathrel{\sim\!\!\!\leftarrow} A(\vec{X})$ iff $\mathcal{H} \Vdash B(\vec{X}) \leftarrow A(\vec{X})$ for every $\mathcal{H} \in \min_{\mathcal{R}} \widehat{A}$.

Example 38. *Consider a fixed language Σ, where we have* PRED $= \{\mathsf{bird}(\cdot), \mathsf{penguin}(\cdot), \mathsf{fly}(\cdot)\}$ *and* CONST $= \{\mathsf{Roxy}, \mathsf{Tom}\}$. *Let $\mathcal{R}$ be the following ranked interpretation, where we use the tabular notation of Chapter 3:*

2	$\mathcal{H}_2 = \{$ bird(Roxy), penguin(Roxy), fly(Roxy) $\}$
1	$\mathcal{H}_1 = \{$ bird(Tom), penguin(Tom), bird(Roxy), fly(Roxy) $\}$
0	$\mathcal{H}_0 = \{$ bird(Roxy), fly(Roxy) $\}$

In this example, $\widehat{\mathsf{bird}} = \{\mathcal{H}_2, \mathcal{H}_1, \mathcal{H}_0\}$, and hence $\min_{\mathcal{R}} \widehat{\mathsf{bird}} = \{\mathcal{H}_0\}$. Since $\mathcal{H}_0 \Vdash \mathsf{fly}(X) \leftarrow \mathsf{bird}(X)$, we conclude from the definitions above that $\mathcal{R} \Vdash \mathsf{fly}(X) \mathrel{\sim\!\!\!\leftarrow} \mathsf{bird}(X)$. Intuitively, this ranked interpretation sanctions the defeasible assertion "birds typically fly".

Since bird(Roxy) *is satisfied by every Herbrand interpretation in $\mathbb{H}^{\mathcal{R}} = \{\mathcal{H}_2, \mathcal{H}_1, \mathcal{H}_0\}$, we can also conclude that $\mathcal{R} \Vdash \mathsf{bird}(\mathsf{Roxy})$, for instance.*

Given a ranked interpretation $\mathcal{R}$, we denote the set of Defeasible Datalog formulas it satisfies by $\mathcal{S}_{\mathcal{R}} \subseteq \mathcal{L}$. One of the attractive features of this semantics, which we will refer to as the *preferential semantics* for Defeasible Datalog, is that these sets are always *rational*:

Lemma 33. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^{\infty}$ be a ranked interpretation. Then $\mathcal{S}_{\mathcal{R}}$ is $\mathbb{R}_{\mathrel{\sim}}$-complete.*

These sets are also complete for a number of the other properties we considered in the previous section:

Lemma 34. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^{\infty}$ be a ranked interpretation. Then $\mathcal{S}_{\mathcal{R}}$ is complete for* (PER), (IRR), (NEG), (SUP), (SUPF), (MP), (EQ) *and* (EQF).

In fact, the only properties from the previous section that don't hold for this semantics are the ones that we explicitly ruled out as being undesirable, namely the properties (UI) and (WKUI) of *universal instantiation*. In short, as far as these kinds of basic properties go, preferential semantics is quite promising.

A slightly less obvious property of preferential semantics is the following, which we mention because it appears to be a common property of *any* preferential-style semantics for Defeasible Datalog, in which formulas are interpreted by minimising the rank of Herbrand interpretations:

Lemma 35. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ be a ranked interpretation. Then $\mathcal{S}_\mathcal{R}$ is complete for the following defeasible property:*

$$(\textsc{Pref}) \quad \bot \mathrel{\not\sim} A(\vec{x}) \implies \neg A(\vec{x}) \mathrel{\not\sim} A(\vec{x}) \vee A(\pi\vec{x})$$

Under the reading we gave for defeasible formulas, this is essentially saying that if a world contains either an instance of $A(\vec{x})$, or an instance of some special case $A(\pi\vec{x})$, then that world cannot typically fail to contain an instance of $A(\vec{x})$. This makes some sense, as a special case of a compound is always an instance of that compound.

Aside from properties, another axis on which to judge a semantics is its *representational capabilities*. In other words, what kinds of knowledge bases can preferential semantics model or not model, and why? One example of what we mean by this is the idea that defeasible reasoning should always allow for *exceptions* to general rules, as this is the entire point of reasoning defeasibly in the first place! For instance, we would expect a defeasible reasoner to find the following knowledge base consistent:

$$\mathcal{K}_{\mathsf{tweety}} = \left\{\mathsf{bird}(\mathsf{Tweety}), \neg\mathsf{fly}(\mathsf{Tweety}), \mathsf{fly}(X) \mathrel{\sim\!\!\!\!\!\!\leftarrow} \mathsf{bird}(X)\right\}$$

This states that Tweety is a bird that cannot fly, as well as the assertion that birds *typically* fly. Intuitively, there is no contradiction here, as we are simply stating that Tweety is an atypical bird. Hence any semantics worth pursuing for Defeasible Datalog should at minimum be able to model $\mathcal{K}_{\mathsf{tweety}}$, a criterion we will refer to as the *"Tweety test"*. Unfortunately, it turns out that preferential semantics cannot handle this test-case:

Lemma 36. *There is no non-trivial ranked interpretation $\mathcal{R}$ over any first-order language satisfying $\mathcal{K}_{\mathsf{tweety}}$.*

The reason for this provides a hint as to how we might go about circumventing this kind of representational problem. In preferential semantics, a defeasible rule is interpreted as a constraint on the *most typical worlds* satisfying an instance of the rule body. Something like $\mathsf{fly}(X) \rightsquigarrow \mathsf{bird}(X)$, for instance, is interpreted as saying that *all* birds fly in the most typical worlds containing birds. This leaves no room for exceptions in these worlds!

In the next section, we describe a modification of preferential semantics that patches this problem, by *enriching* Herbrand interpretations with additional structure that distinguishes typical instances of a predicate from atypical instances.

6.6 Enriched Preferential Semantics

To begin with, let discuss something that was mentioned in Section 6.1. In some knowledge bases, such as the $\mathcal{K}_{\mathsf{tweety}}$ given above, it is clear that the individuals mentioned in the knowledge base are insufficient to produce a model of that knowledge base. For instance, the only individual mentioned in $\mathcal{K}_{\mathsf{tweety}}$ is Tweety, whom we know to be an atypical bird. Thus, in order for the formula $\mathsf{fly}(X) \mathrel{\sim\!\!\leftarrow} \mathsf{bird}(X)$ to hold, any model of $\mathcal{K}_{\mathsf{tweety}}$ *must* contain other individuals.

To put it differently, if Tweety is the only bird in the world, and Tweety doesn't fly, on what grounds can we claim that birds typically fly?

This is essentially the *closed-world assumption* (CWA) versus the *open-world assumption* (OWA). As we discussed in Chapter 5, Datalog traditionally embraces the CWA, and takes the position that the only individuals assumed to exist are those explicitly mentioned in a given Datalog program. Here, we break from that tradition and choose to model the OWA instead, giving out semantics a more Description Logic flavour. From a technical perspective, this sorts out the problem we have just mentioned about incomplete knowledge bases. We also note that one generally only *wants* to reason defeasibly when one has incomplete knowledge, as otherwise classical reasoning is a more suitable tool.

Let $\Sigma = \langle \textsc{pred}, \textsc{const}, \textsc{var} \rangle$ be a fixed first order language. Then we recall from Section 6.1 that $\mathbb{U}$ denotes the *Herbrand universe* (the set of constants in the language), and $\mathbb{B}$ the *Herbrand base* (the set of ground atoms over the Herbrand universe). Our first step is to model the OWA, as mentioned above, by enriching the set of constants in the language with an additional set $\mathbb{E}$ of *enrichment constants*, which we assume to be disjoint from $\mathbb{U} = \textsc{const}$. This gives rise to an *enriched Herbrand universe* $\tilde{\mathbb{U}} = \mathbb{U} \cup \mathbb{E}$, containing both the original universe as well as the enrichment constants. Similarly, we denote by $\tilde{\mathbb{B}}$ the *enriched Herbrand base*, containing every ground atom over $\tilde{\mathbb{U}}$.

As mentioned in the last chapter, our second step is to enrich Herbrand interpretations with additional structure that lets us distinguish the *typical* from the *atypical*:

Definition 31. *An* enriched Herbrand interpretation *is a pair* $\mathcal{E} = \langle \tilde{\mathcal{H}}_{\mathcal{E}}, T_{\mathcal{E}} \rangle$, *where* $\tilde{\mathcal{H}}_{\mathcal{E}} \subseteq \tilde{\mathbb{B}}$ *is a Herbrand interpretation over the enriched universe and* $T_{\mathcal{E}} \subseteq \tilde{\mathbb{U}}$ *is a set of* typical constants.

The intuition here is that $\tilde{\mathcal{H}}_{\mathcal{E}}$ tells us what facts hold in the interpretation, and $T_{\mathcal{E}}$ tells us which of these facts are *typical* for the interpretation, and which are unusual or *atypical* in some way. For instance, if $\mathsf{Tweety} \in T_{\mathcal{E}}$ and $\tilde{\mathcal{H}}_{\mathcal{E}} \Vdash \mathsf{bird}(\mathsf{Tweety})$, then we would consider Tweety to be a typical bird. We denote the set of all enriched Herbrand interpretations by $\tilde{\mathbb{H}}$.

For clarity, we denote classical satisfaction by $\mathcal{E} \Vdash A$ or $\mathcal{E} \Vdash B \leftarrow A$ whenever $\tilde{\mathcal{H}}_{\mathcal{E}} \Vdash A$ or $\tilde{\mathcal{H}}_{\mathcal{E}} \Vdash B \leftarrow A$ for a fact or (non-defeasible) rule respectively. For a given rule $B \leftarrow A \in \mathcal{L}_{\rightarrow}$, we also define a notion of *typical satisfaction* as

follows:

1. $\mathcal{E} \mid\!\sim B \leftarrow A$ iff $\mathcal{E} \Vdash \varphi(B) \leftarrow \varphi(A)$ for every substitution $\varphi : \text{VAR} \to T_{\mathcal{E}}$.

We say that an enriched Herbrand interpretation $\mathcal{E}$ *typically satisfies* the rule $B \leftarrow A$ whenever $\mathcal{E} \mid\!\sim B \leftarrow A$ holds. This means that the rule is valid when restricted to the subset of *typical constants* in the interpretation, but not necessarily across the entire domain. In other words, $\mathcal{E} \Vdash B \leftarrow A$ implies $\mathcal{E} \mid\!\sim B \leftarrow A$, but the converse is not necessarily true.

Example 39. *Consider a first-order language with* $\text{PRED} = \{\mathsf{worker}(\cdot), \mathsf{boss}(\cdot), \mathsf{worksFor}(\cdot,\cdot)\}$ *and* $\text{CONST} = \{\mathsf{Toby}, \mathsf{Todd}\}$*, and let* $\mathbb{E} = \{\mathsf{E}_1, \mathsf{E}_2, \dots\}$ *be our set of enrichment constants. An example of an enriched Herbrand interpretation is* $\mathcal{E} = \langle \tilde{\mathcal{H}}_{\mathcal{E}}, T_{\mathcal{E}} \rangle$*, where:*

1. $\tilde{\mathcal{H}}_{\mathcal{E}} = \{\mathsf{worker}(\mathsf{Toby}), \mathsf{boss}(\mathsf{E}_1), \mathsf{boss}(\mathsf{Todd}), \mathsf{worksFor}(\mathsf{Toby}, \mathsf{E}_1)\}$
2. $T_{\mathcal{E}} = \{\mathsf{Toby}, \mathsf{E}_1\}$

Then $\mathcal{E} \not\Vdash \mathsf{worksFor}(X, Y) \leftarrow \mathsf{worker}(X) \wedge \mathsf{boss}(Y)$*, as Toby is a worker and Todd is a boss, yet Toby does not work for Todd. Nevertheless,* $\mathcal{E} \mid\!\sim \mathsf{worksFor}(X, Y) \leftarrow \mathsf{worker}(X) \wedge \mathsf{boss}(Y)$ *holds, as the only* typical constants *satisfying* $\mathsf{worker}(X)$ *and* $\mathsf{boss}(Y)$ *are* Toby *and* E_1*, and* $\mathsf{worksFor}(\mathsf{Toby}, \mathsf{E}_1)$ *holds.*

In order to interpret *defeasible* formulas, we follow the same route as preferential semantics and define an analogue of a propositional *ranked interpretation* (see Definition 7) as follows:

Definition 32. *An* enriched ranked interpretation $\tilde{\mathcal{R}}$ *is a function* $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^\infty$*, where* $\Omega^\infty = \Omega \cup \{\infty\}$ *for some totally ordered set* Ω*, satisfying the following convexity condition:*

$$\begin{gathered}\textit{For all } \mathcal{E} \in \tilde{\mathbb{H}} \textit{ such that } \tilde{\mathcal{R}}(\mathcal{E}) < \infty\textit{, and all } i \in \mathbb{N} \textit{ such that} \\ 0 \leq i < \tilde{\mathcal{R}}(\mathcal{E})\textit{, there exists some } \mathcal{E}' \in \tilde{\mathbb{H}} \textit{ such that } \tilde{\mathcal{R}}(\mathcal{E}') = i.\end{gathered}$$

Note that we have slightly relaxed the propositional definition and allowed for an *arbitrary* totally ordered set Ω in the domain, rather than fixing it to be the naturals $\mathbb{N}$. This gives us a bit more flexibility, and it also solves a technical problem that enrichment introduces. Generally speaking, the knowledge bases we want to model are finite. In propositional logic, there are only a finite number of possible valuations over a finite language, and thus $\mathbb{N}$ always suffices to rank them. Here, even if we have a finite knowledge base, we may have an infinite set of enrichment constants $\mathbb{E}$, and thus there are in general infinitely many possible enriched Herbrand interpretations. This means that me may need a slightly larger total order to rank them properly.

Classical satisfation for an enriched Herbrand interpretation $\mathcal{E}$ is defined exactly as for their non-enriched counterparts. Denoting the set of *consistent worlds* with respect to $\mathcal{R}$ by $\tilde{\mathbb{H}}^{\tilde{\mathcal{R}}} = \{\mathcal{E} \in \tilde{\mathbb{H}} : \tilde{\mathcal{R}}(\mathcal{E}) < \infty\}$, we have:

1. For a fact $A \in \mathcal{L}_f$, $\tilde{\mathcal{R}} \Vdash A$ iff $\mathcal{E} \Vdash A$ for every $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}$.
2. For a rule $B \leftarrow A \in \mathcal{L}_{\rightarrow}$, $\tilde{\mathcal{R}} \Vdash B \leftarrow A$ iff $\mathcal{E} \Vdash B \leftarrow A$ for every $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}$.

Satisfaction for *defeasible* formulas, on the other hand, is quite different. For any compound $A \in \mathcal{L}_c$, let $\tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A) = \{\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}} : \mathcal{E} \Vdash \varphi(A) \text{ for some } \varphi : \text{VAR} \rightarrow T_{\mathcal{E}}\}$ denote the set of consistent worlds satsfying some *typical* instance of A. Then we have:

1. For a defeasible rule $B \mathrel{\reflectbox{\rightsquigarrow}} A \in \mathcal{L}_{\rightsquigarrow}$, $\tilde{\mathcal{R}} \Vdash B \mathrel{\reflectbox{\rightsquigarrow}} A$ iff $\mathcal{E} \mid\!\sim B \leftarrow A$ for every $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$. Note the use of *typical satisfation*.

The intuitive reading of this is to interpret a defeasible rule $B \mathrel{\reflectbox{\rightsquigarrow}} A$ as follows:

> *"In the most typical worlds containing A-instances,*
> *all the typical A-instances are B-instances."*

One way to look at this is that an enriched ranked interpretation is making use of a *double* ranking, as can be seen from the two (distinct) uses of the word *typical* in the reading. The first ranking is simply that of the enriched Herbrand interpretations, each of which gets assigned a rank $\tilde{\mathcal{R}}(\mathcal{E}) \in \mathbb{N}^{\infty}$. The second ranking is *within* each enriched Herbrand interpretation $\mathcal{E}$, in which constants are partitioned into *typical* ($T_{\mathcal{E}}$) and *atypical* ($\tilde{\mathbb{U}} \setminus T_{\mathcal{E}}$). While we stick with the simpler option in this thesis, an interesting avenue for future research would be to extend this two-layer partition into a proper ranking function over $\tilde{\mathbb{U}}$, at which point the semantics starts to resemble that of Friedman et al.[17] for first-order logic.

Example 40. *Consider a first-order language with* $\text{PRED} = \{\mathsf{bird}(\cdot), \mathsf{fly}(\cdot)\}$ *and* $\text{CONST} = \{\mathsf{Tweety}\}$*, and let* $\mathbb{E} = \{E_1, E_2, \dots\}$ *be the usual set of enrichment constants. Then consider the enriched Herbrand interpretation* $\mathcal{E} = \langle \tilde{\mathcal{H}}_{\mathcal{E}}, T_{\mathcal{E}} \rangle$*, where:*

1. $\tilde{\mathcal{H}}_{\mathcal{E}} = \{\mathsf{bird}(\mathsf{Tweety}), \mathsf{bird}(\mathsf{E}_1), \mathsf{fly}(\mathsf{E}_1)\}$
2. $T_{\mathcal{E}} = \{\mathsf{E}_1\}$

Now consider the following enriched ranked interpretation $\tilde{\mathcal{R}}$*, in which there is only a single possible world:*

0	$\mathcal{E}$

Then $\tilde{\mathcal{R}} \Vdash \mathsf{bird}(\mathsf{Tweety})$ *and* $\tilde{\mathcal{R}} \Vdash \neg\mathsf{fly}(\mathsf{Tweety})$*, and since* $\min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(\mathsf{bird}(X)) = \{\mathcal{E}\}$ *and* $\mathcal{E} \mid\!\sim \mathsf{fly}(X) \leftarrow \mathsf{bird}(X)$*, we have that* $\tilde{\mathcal{R}} \Vdash \mathsf{fly}(X) \mathrel{\reflectbox{\rightsquigarrow}} \mathsf{bird}(X)$ *as well. Though this example is near-trivial, it illustrates the fact that enriched Herbrand interpretations can model the "*Tweety test*" knowledge base from the last section.*

Our first step in the analysis of *enriched preferential semantics* is to work out which defeasible properties it enjoys. Unsurprisingly, the answer turns out to be similar to last section's preferential semantics. Letting $\mathcal{S}_{\tilde{\mathcal{R}}}$ denote the set of Defeasible Datalog formulas satisfies by $\tilde{\mathcal{R}}$, we have:

Theorem 8. *Let $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^{\infty}$ be an enriched ranked interpretation. Then $\mathcal{S}_{\tilde{\mathcal{R}}}$ is $\mathbb{R}_{\rightsquigarrow}$-complete, and also complete for* (Pref)*,* (Sup)*,* (SupF)*,* (Per)*,* (WkIrr)*,* (WkNeg)*,* (Mp)*,* (Eq) *and* (EqF)*.*

We will refer to this collection of defeasible properties as the *enriched rationality postulates*, and denote it by $\tilde{\mathbb{R}}_{\rightsquigarrow}$. The enriched postulates are almost identical to the properties satisfied by preferential semantics, the difference being that (Irr) and (Neg) have been substituted for their weak counterparts. As discussed in Section 6.4, the (WkNeg) property is probably more desirable than (Neg), so this is a point in favour of enriched preferential semantics. On the other hand, having to accept the weaker (WkIrr) property is a point against it, and finding ways to fix this would be another interesting avenue for future research.

Something that sets enriched preferential semantics apart quite strongly, however, is that we have been able to obtain a *converse* to Theorem 8. This allows us to characterise $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete sets of formulas as precisely those sets satisfied by enriched ranked interpretations:

Theorem 9. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set of formulas. Then there exists some enriched ranked interpretation $\tilde{\mathcal{R}}$ such that $\mathcal{S} = \mathcal{S}_{\tilde{\mathcal{R}}}$.*

Together, Theorems 8 and 9 provide a Defeasible Datalog analogue of the characterisation of preferential and rational sets in propositional defeasible reasoning (see Lemmas 8 and 9 in Chapter 3).

6.7 Rank Entailment and Minimal Model Entailment

In this section we define and discuss two basic kinds of entailment relations for Defeasible Datalog, making use of enriched preferential semantics. These are *rank entailment*, which is based on propositional rank entailment, and *minimal-model entailment*, which is based on propositional Rational Closure. As we will see, both of these have some shortcomings in a Defeasible Datalog context.

Rank entailment makes use of the Tarskian account of logical consequence [33], in which a formula follows from a knowledge base iff it true in *all models* of that knowledge base:

Definition 33. *A knowledge base $\mathcal{K} \subseteq \mathcal{L}$* rank entails *a formula $\alpha \in \mathcal{L}$, denoted $\mathcal{K} \mid\!\approx_{\mathcal{R}} \alpha$, iff $\tilde{\mathcal{R}} \Vdash \alpha$ for every enriched ranked interpretation $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^{\infty}$ satisfying $\mathcal{K}$.*

Rank entailment is thus extremely conservative, as it considers even those models that are intuitively quite unlikely (they may contain worlds that are ranked much higher than they need to be, for instance). As discussed in Chapters 3 and 4, however, rank entailment enjoys a number of meta-properties that make it useful as a *lower bound* on what we would expect of an entailment relation:

Lemma 37. *Rank entailment satisfies the following three properties:*

$$\begin{aligned}
&(\textsc{Incl}) \quad \alpha \in \mathcal{K} \textit{ implies } \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \\
&(\textsc{Cumu}) \quad \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \textit{ and } \mathcal{K} \cup \{\alpha\} \mathrel{|\!\approx_{\mathcal{R}}} \beta \textit{ implies } \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \beta \\
&(\textsc{Mono}) \quad \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \textit{ implies } \mathcal{K} \cup \{\beta\} \mathrel{|\!\approx_{\mathcal{R}}} \alpha
\end{aligned}$$

Another nice property of rank entailment is that, much like the underlying enriched ranked interpretations, it satisfies a number of the defeasible properties we mentioned in the last section:

Lemma 38. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base, and consider the set $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha\}$. Then $\mathcal{S}$ is complete for every enriched postulate $\tilde{\mathbb{R}}_{\rightsquigarrow}$ except* (Rm).

Let us now look at an example of rank entailment in action, as a benchmark we can compare other entailment relations to:

Example 41. *Consider a first-order language with* $\textsc{pred} = \{\mathsf{worker}(\cdot), \mathsf{boss}(\cdot), \mathsf{responsible}(\cdot), \mathsf{hasSuperior}(\cdot,\cdot)\}$ *and* $\textsc{const} = \{\mathsf{Guy}, \mathsf{Tommie}, \mathsf{Giovanni}\}$*, and let $\mathbb{E} = \{E_1, E_2, \dots\}$ be the usual set of enrichment constants. Then let $\mathcal{K}$ be the knowledge base from Example 29:*

$$\begin{aligned}
\mathsf{worker}(X) &\leftarrow \mathsf{boss}(X) \\
\mathsf{responsible}(X) &\leftsquigarrow \mathsf{boss}(X) \\
\mathsf{boss}(Y) &\leftsquigarrow \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y)
\end{aligned}$$

An application of (Incl) *and* (Rm) *(see Lemmas 37 and 38) shows that $\mathcal{K} \mid\!\approx_{\mathcal{R}} \mathsf{worker}(Y) \leftsquigarrow \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y)$, and so rank entailment allows us to infer at least some non-trivial consequences.*

On the other hand, since worker's superiors are typically bosses, and bosses are typically responsible, a consequence we might expect intuitively is that worker's superiors are typically responsible. After all, there's nothing in the knowledge base to suggest that worker's bosses are atypical in any way. Unfortunately, rank entailment does not *sanction this consequence, a fact we will now prove.*

Let $\mathcal{E}_1 = \langle \tilde{\mathcal{H}}_{\mathcal{E}_1}, \{\mathsf{Guy}, \mathsf{Tommie}\}\rangle$ and $\mathcal{E}_2 = \langle \tilde{\mathcal{H}}_{\mathcal{E}_2}, \{\mathsf{Giovanni}\}\rangle$, where:

1. $\tilde{\mathcal{H}}_{\mathcal{E}_1} = \{\mathsf{worker}(\mathsf{Guy}), \mathsf{boss}(\mathsf{Tommie}), \mathsf{worker}(\mathsf{Tommie}), \mathsf{hasSuperior}(\mathsf{Guy}, \mathsf{Tommie})\}$
2. $\tilde{\mathcal{H}}_{\mathcal{E}_2} = \{\mathsf{boss}(\mathsf{Giovanni}), \mathsf{worker}(\mathsf{Giovanni}), \mathsf{responsible}(\mathsf{Giovanni})\}$

Now consider the enriched ranked interpretation $\tilde{\mathcal{R}}$ defined as follows:

1	$\mathcal{E}_1$
0	$\mathcal{E}_0$

Then $\tilde{\mathcal{R}} \Vdash \mathcal{K}$, and thus $\tilde{\mathcal{R}}$ is a model that must be taken into consideration in rank entailment. However, $\tilde{\mathcal{R}} \nVdash \mathsf{responsible}(Y) \leftsquigarrow \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y)$, and thus that conclusion is not rank entailed by $\mathcal{K}$ either.

An entailment relation that correctly handles Example 41 must necessarily be *non-monotonic*. If we were to later learn that worker's superiors are always atypical in some way, for instance, then we would want to retract the conclusion that they are typically responsible. And in fact, a non-monotonic entailment relation that has precisely this behaviour for the propositional case is the *minimal-model* entailment relation of Lemma 14, Chapter 3.

The idea is to construct an enriched ranked model of a given knowledge base in which the ranks of each enriched Herbrand interpretation are *as low as possible*, while still satisfying every formula in the knowledge base. Surprisingly, for consistent knowledge bases (i.e. those with at least one enriched ranked model) there is always a unique minimal model:

Lemma 39. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a consistent knowledge base, and define the* minimal model *of $\mathcal{K}$ to be the following enriched ranked interpretation $\tilde{\mathcal{R}}_{\mathcal{K}} : \tilde{\mathbb{H}} \to \mathbb{N}^{\infty}$:*

$$\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = \min \left\{ \tilde{\mathcal{R}}(\mathcal{E}) : \tilde{\mathcal{R}} \Vdash \mathcal{K} \right\}$$

Then this minimal model always satisfies $\mathcal{K}$.

The fact that this produces a model of $\mathcal{K}$ may seem surprising in light of the discussion surrounding Lemma 21 in Chapter 4. In the ALC case, adding assertional statements to a knowledge base can result in it having multiple non-equivalent minimal models. Given that Defeasible Datalog allows for ground facts, which are analogous to assertional statements, why don't we see uniqueness breaking down here as well?

The answer is that, while Defeasible Datalog is more expressive than ALC in some ways, it is actually *less* expressive in others. Defeasible Datalog has no equivalent to the existential restrictions of ALC, and it is these that cause the uniqueness failures. Exploring what happens when Defeasible Datalog is extended with existential quantifiers is an important task for future research.

In any case, we can use these minimal models to define a non-monotonic entailment relation that correctly handles Example 41:

Definition 34. *A knowledge base $\mathcal{K} \subseteq \mathcal{L}$* minimal-model entails *a formula $\alpha \in \mathcal{L}$, denoted $\mathcal{K} \mid\!\approx_M \alpha$, iff either $\mathcal{K}$ is inconsistent or $\tilde{\mathcal{R}}_{\mathcal{K}} \Vdash \alpha$.*

Example 42. *Let $\mathcal{K}$ be the knowledge base from example 41, over the same*

first-order language and set of enrichment constants:

$$\begin{aligned} \mathsf{worker}(X) &\leftarrow \mathsf{boss}(X) \\ \mathsf{responsible}(X) &\hookleftarrow \mathsf{boss}(X) \\ \mathsf{boss}(Y) &\hookleftarrow \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y) \end{aligned}$$

Let $A = \mathsf{worker}(X) \wedge \mathsf{hasSuperior}(X, Y)$ *and* $B = \mathsf{responsible}(Y)$*. We claim that, unlike rank entailment,* $\mathcal{K} \mid\!\approx_M B \hookleftarrow A$*. First, we show that there is at least one* $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_\mathcal{K}}(A)$ *with* $\tilde{\mathcal{R}}_\mathcal{K}(\mathcal{E}) = 0$*, and hence* $\tilde{\mathcal{R}}(\mathcal{E}) = 0$ *for* all $\mathcal{E} \in \min_{\tilde{\mathcal{R}}_\mathcal{K}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_\mathcal{K}}(A)$*. Let* $\tilde{\mathcal{R}} = \langle \{\tilde{\mathcal{H}}_\mathcal{E}, \{\mathsf{E}_1\}\rangle$*, where:*

$$\tilde{\mathcal{H}}_\mathcal{E} = \{\mathsf{worker}(\mathsf{E}_1), \mathsf{boss}(\mathsf{E}_1), \mathsf{responsible}(\mathsf{E}_1), \mathsf{hasSuperior}(E_1, E_1)\}$$

Then consider the enriched ranked interpretation $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^\infty$ *where* $\tilde{\mathcal{R}}(\mathcal{E}) = 0$ *and everything else gets rank* ∞*. Then since* $\tilde{\mathcal{R}} \Vdash \mathcal{K}$*, we conclude from the definitions that* $\tilde{\mathcal{R}}_\mathcal{K}(\mathcal{E}) = 0$*. Since* $\tilde{\mathcal{R}} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$*, we also conclude that* $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_\mathcal{K}}(A)$*.*

Finally, consider any $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_\mathcal{K}}(A)$ *and substitution* $\varphi : \textsc{var} \to T_\mathcal{E}$ *such that* $\mathcal{E} \Vdash \varphi(A)$*. As we have already shown,* $\tilde{\mathcal{R}}_\mathcal{K}(\mathcal{E}) = 0$ *and thus* $\mathcal{E}$ *typically satisfies every rule of* $\mathcal{K}$*. In particular, we conclude that* $\mathcal{E} \Vdash \varphi(\mathsf{boss}(Y))$ *from the third rule of* $\mathcal{K}$*, and hence* $\mathcal{E} \Vdash \varphi(\mathsf{responsible}(Y))$ *from the second rule. But then this implies that* $\mathcal{E} \mid\!\sim B \hookleftarrow A$*, which in turn implies that* $\tilde{\mathcal{R}}_\mathcal{K} \Vdash B \hookleftarrow A$ *as required.*

We note that minimal-model entailment, unlike rank entailment, is defeasible on both the object-level *and* the meta-level. Another nice property of minimal-model entailment, which follows immediately from the fact that it is defined in terms of a single enriched ranked interpretation, is that it satisfies all of the enriched rationality postulates:

Lemma 40. *Let* $\mathcal{K} \subseteq \mathcal{L}$ *be a knowledge base, and consider the set* $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mid\!\approx_M \alpha\}$*. Then* $\mathcal{S}$ *is complete for* $\tilde{\mathbb{R}}_{\rightsquigarrow}$*.*

The way we have defined it, minimal-model entailment for Defeasible Datalog can be viewed as a strict generalisation of propositional rational closure (see Definition 10) and ALC rational closure (see Definitions 20 and 24).

This suggest that minimal-model entailment for Defeasible Datalog should have similar strengths and weaknesses to rational closure for propositional logic or ALC. An example of a shared strength, illustrated by Example 42, is that minimal-model entailment can "propagate" defeasible information across rule boundaries in a sensible way (i.e. whenever it doesn't contradict other rules entailed by the knowledge base). On the other hand, a *weakness* we might expect it to share is that rational closure behaves somewhat poorly in the presence of assertional statements in defeasible ALC.

One aspect of this is that rational closure for ALC doesn't have the *Single Model Property*, as knowledge bases can in general have multiple minimal models (see

Example 26, for instance). As we discussed earlier, minimal-model entailment for Defeasible Datalog evades this problem due to its lack of existential quantification. Nevertheless, there *is* a sense in which minimal-model entailment handles ground facts poorly, which we illustrate in the following example:

Example 43. *Consider a first-order language with* PRED $= \{\mathsf{bird}(\cdot), \mathsf{fly}(\cdot)\}$ *and* CONST $= \{\mathsf{Fred}\}$*, and let* $\mathbb{E} = \{\mathsf{E}_1\}$ *be the only enrichment constant. We take* $\mathbb{E}$ *to be finite like this for reasons we will go into in the next section. Now let* $\mathcal{K}$ *be the following knowledge base:*

$$\mathsf{bird}(\mathsf{Fred})$$
$$\mathsf{fly}(X) \hookleftarrow \mathsf{bird}(X)$$

Should $\mathcal{K}$ *entail* $\mathsf{fly}(\mathsf{Fred}) \hookleftarrow \top$*? From a Tarskian point of view, the answer is* no*, because there are non-trivial models of* $\mathcal{K}$ *that satisfy* $\neg\mathsf{fly}(\mathsf{Fred})$*. In the spirit of defeasible reasoning, on the other hand, the answer is probably* yes*. Since there is no explicit reason to suspect that Fred is an* atypical *bird, it seems reasonable to assume that Fred should satisfy the properties of a* typical *one, at least until we obtain evidence to the contrary.*

However, $\mathcal{K} \not\approx_M \mathsf{fly}(\mathsf{Fred}) \hookleftarrow \top$*. In other words, minimal-model entailment does* not *sanction this conclusion. To see this, consider the enriched Herbrand interpretation* $\mathcal{E} = \langle \tilde{\mathcal{H}}_{\mathcal{E}}, \{\mathsf{E}_1\}\rangle$*, where:*

$$\tilde{\mathcal{H}}_{\mathcal{E}} = \{\mathsf{bird}(\mathsf{Fred}), \mathsf{bird}(\mathsf{E}_1), \mathsf{fly}(\mathsf{E}_1)\}$$

Now consider the enriched ranked interpretation $\tilde{\mathcal{R}}$*, where* $\tilde{\mathcal{R}}(\mathcal{E}) = 0$ *and everything else has rank* ∞*. Then* $\tilde{\mathcal{R}} \Vdash \mathcal{K}$*, which implies in the minimal model of* $\mathcal{K}$ *that* $\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = 0$*. But this implies directly that* $\tilde{\mathcal{R}}_{\mathcal{K}} \not\Vdash \mathsf{fly}(\mathsf{Fred}) \hookleftarrow \top$*.*

6.8 Refinement for Ground Fact Inference

In this section we introduce *refinement*, which is a procedure for taking an enriched ranked interpretation and producing an *improved* version of it that satisfies the following two properties:

1. The improved version satisfies all of the formulas satisfies by the original.
2. The improved version propagates defeasible rules to ground facts in a consistent, non-monotonic fashion.

Combined with minimal-model entailment, refinement will allow us to solve the problem of ground fact entailment that was explored in Example 43. And in fact, refinement can be applied to any entailment relation based on enriched ranked interpretations in order to improve the sets of consequences it sanctions.

To understand how it works, let's unravel the example a little bit and try to understand what's going wrong in the minimal model:

Example 44. *Consider the knowledge base $\mathcal{K}$ from Example 43. Since $\tilde{\mathcal{R}}_{\mathcal{K}} \not\Vdash \mathsf{fly}(\mathsf{Fred}) \mathrel{\sim\!\!\!\!\!\sim} \top$, there must be some enriched Herbrand interpretations $\mathcal{E} \in \min_{\tilde{\mathcal{R}}_{\mathcal{K}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_{\mathcal{K}}}(\top)$ such that $\mathcal{E} \not\Vdash \mathsf{fly}(\mathsf{Fred})$.*

$$\begin{aligned} \mathcal{E}_1 &= \langle \tilde{\mathcal{H}}_1, \emptyset \rangle, & \tilde{\mathcal{H}}_1 &= \{\} \\ \mathcal{E}_2 &= \langle \tilde{\mathcal{H}}_2, \{\mathsf{E}_1\} \rangle, & \tilde{\mathcal{H}}_2 &= \{\} \\ \mathcal{E}_3 &= \langle \tilde{\mathcal{H}}_3, \{\mathsf{E}_1, \mathsf{Fred}\} \rangle, & \tilde{\mathcal{H}}_3 &= \{\} \end{aligned}$$

Similar reasoning to Example 43 shows that $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\} \subseteq \min_{\tilde{\mathcal{R}}_{\mathcal{K}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_{\mathcal{K}}}(\top)$, and note that $\mathcal{E}_1 \not\Vdash \mathsf{fly}(\mathsf{Fred})$ and $\mathcal{E}_2 \not\Vdash \mathsf{fly}(\mathsf{Fred})$. Interpretations like these are essentially the reason that $\mathcal{K} \not\approx_M \mathsf{fly}(\mathsf{Fred}) \mathrel{\sim\!\!\!\!\!\sim} \top$.

However, notice that $T_{\mathcal{E}_1} \subset T_{\mathcal{E}_2} \subset T_{\mathcal{E}_3}$. $\mathcal{E}_1$ has fewer typical constants than $\mathcal{E}_2$, which in turn has fewer typical constants than $\mathcal{E}_3$. Since $\mathcal{E}_3 \Vdash \mathsf{fly}(\mathsf{Fred})$, this suggests that the Herbrand interpretations we really *want to be considering minimal are those with the greatest number of typical constants.*

Refinement does exactly this - given an enriched ranked interpretation $\tilde{\mathcal{R}}$, refinement takes each rank-layer of enriched Herbrand interpretations and splits them into multiple layers, based on the size of their typicality sets.

Defining a refinement operator for arbitrary enriched ranked interpretations is no doubt possible, but requires some careful mathematics to deal with infinite first-order languages. To simplify matters, we will consider only the case of *logically finite* languages, i.e. those with finite sets of predicates, constants and enrichment constants. The importance of logical finiteness is that there are only finitely many enriched Herbrand interpretations over a logically finite language. This implies that we need only consider enriched ranked interpretations $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^\infty$ where $\Omega = \mathbb{N}$, of which there are only finitely many.

Definition 35. *Let $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \mathbb{N}^\infty$ be an enriched ranked interpretation over a logically finite language. Then we define* $\mathrm{ref}(\tilde{\mathcal{R}}) : \tilde{\mathbb{H}} \to (\mathbb{N} \times \mathbb{N})^\infty$*, called the* refinement *of $\tilde{\mathcal{R}}$, via the following induction:*

1. *$\mathcal{E} \mapsto (i, 0)$ iff $\tilde{\mathcal{R}}(\mathcal{E}) = i$ and there's no $\mathcal{E}' \in \tilde{\mathbb{H}}$ s.t. $\tilde{\mathcal{R}}(\mathcal{E}') = i$ and $T_{\mathcal{E}} \subset T_{\mathcal{E}'}$.*
2. *$\mathcal{E} \mapsto (i, j)$ iff $\tilde{\mathcal{R}}(\mathcal{E}) = i$ and for all $\mathcal{E}' \in \tilde{\mathbb{H}}$ with $\tilde{\mathcal{R}}(\mathcal{E}') = i$, either $\mathcal{E}' \mapsto (i, k)$ for some $k < j$ or $T_{\mathcal{E}} \not\subset T_{\mathcal{E}'}$.*

Strictly speaking, the refinement of an enriched ranked interpretation $\tilde{\mathcal{R}}$ isn't another ranked interpretation, as the range of $\mathrm{ref}(\tilde{\mathcal{R}})$ is $(\mathbb{N} \times \mathbb{N})^\infty$ rather than the $\mathbb{N}^\infty$ that Definition 32 requires. Because of logical finiteness, however, we can always rearrange $\mathrm{ref}(\tilde{\mathcal{R}})$ into a true ranked interpretation.

For this reason, we will abuse the definition a bit and treat $\mathrm{ref}(\tilde{\mathcal{R}})$ as an enriched ranked interpretation. Definition 35 is quite involved, so let's look at an example to clarify things:

Example 45. *Consider a finitary first-order language with* PRED $= \{\mathsf{elephant}(\cdot), \mathsf{keeper}(\cdot), \mathsf{likes}(\cdot,\cdot)\}$, CONST $= \{\mathsf{Fred}, \mathsf{Clive}\}$ *and* $\mathbb{E} = \{\mathsf{E}_1\}$. *Then we can construct some enriched Herbrand interpretations as follows:*

$$\mathcal{E}_1 = \langle \tilde{\mathcal{H}}_1, \{\mathsf{Fred}\}\rangle, \qquad \tilde{\mathcal{H}}_1 = \{\mathsf{keeper}(\mathsf{Fred}), \mathsf{elephant}(\mathsf{Clive})\}$$
$$\mathcal{E}_2 = \langle \tilde{\mathcal{H}}_2, \{\mathsf{Fred}, \mathsf{Clive}\}\rangle, \quad \tilde{\mathcal{H}}_2 = \{\mathsf{keeper}(\mathsf{Fred}), \mathsf{elephant}(\mathsf{Clive}), \mathsf{likes}(\mathsf{Fred}, \mathsf{Clive})\}$$
$$\mathcal{E}_3 = \langle \tilde{\mathcal{H}}_3, \{\mathsf{E}_1\}\rangle, \qquad \tilde{\mathcal{H}}_3 = \{\mathsf{elephant}(\mathsf{E}_1)\}$$
$$\mathcal{E}_4 = \langle \tilde{\mathcal{H}}_4, \{\mathsf{Fred}\}\rangle, \qquad \tilde{\mathcal{H}}_4 = \{\mathsf{keeper}(\mathsf{E}_1), \mathsf{likes}(\mathsf{E}_1, ,)\mathsf{keeper}(\mathsf{Clive})\}$$
$$\mathcal{E}_5 = \langle \tilde{\mathcal{H}}_5, \{\mathsf{Clive}\}\rangle, \qquad \tilde{\mathcal{H}}_5 = \{\mathsf{keeper}(\mathsf{Clive}), \mathsf{elephant}(\mathsf{Clive})\}$$
$$\mathcal{E}_6 = \langle \tilde{\mathcal{H}}_6, \emptyset\rangle, \qquad \tilde{\mathcal{H}}_6 = \{\mathsf{likes}(\mathsf{E}_1, \mathsf{Fred}), \mathsf{elephant}(\mathsf{Fred})\}$$

These can be arranged into the following enriched ranked interpretation $\tilde{\mathcal{R}}$*:*

2	$\mathcal{E}_5,\ \mathcal{E}_6$
1	$\mathcal{E}_3,\ \mathcal{E}_4$
0	$\mathcal{E}_1,\ \mathcal{E}_2$

Note that $T_{\mathcal{E}_1} \subset T_{\mathcal{E}_2}$, $T_{\mathcal{E}_6} \subset T_{\mathcal{E}_5}$ *and that* $T_{\mathcal{E}_3}$ *and* $T_{\mathcal{E}_4}$ *are incomparable. Taking the refinement according to Definition 35,* $\mathrm{ref}(\tilde{\mathcal{R}})$ *thus separates the various ranks as follows:*

$(2,1)$	$\mathcal{E}_6$
$(2,0)$	$\mathcal{E}_5$
$(1,0)$	$\mathcal{E}_3,\ \mathcal{E}_4$
$(0,1)$	$\mathcal{E}_1$
$(0,0)$	$\mathcal{E}_2$

Finally, by logical finiteness, this can be relabelled into the following enriched ranked interpretation:

4	$\mathcal{E}_6$
3	$\mathcal{E}_5$
2	$\mathcal{E}_3,\ \mathcal{E}_4$
1	$\mathcal{E}_1$
0	$\mathcal{E}_2$

Refinment has some interesting effects on $\tilde{\mathcal{R}}$*. Note, for instance, that* $\tilde{\mathcal{R}}$ *satisfies* $\mathsf{likes}(X,Y) \leftsquigarrow \mathsf{keeper}(X) \wedge \mathsf{elephant}(Y)$ *but not* $\mathsf{likes}(\mathsf{Fred}, \mathsf{Clive}) \leftsquigarrow \mathsf{keeper}(\mathsf{Fred}) \wedge \mathsf{elephant}(\mathsf{Clive})$*. The refined interpretation* $\mathrm{ref}(\tilde{\mathcal{R}})$ *satisfies both, which seems desirable in light of the fact that* $\tilde{\mathcal{R}}$ *implies nothing atypical about Fred or Clive.*

Refinement allows us to extend minimal-model entailment as follows:

Definition 36. *A knowledge base* $\mathcal{K} \subseteq \mathcal{L}$ refined entails *a formula* $\alpha \in \mathcal{L}$*, denoted* $\mathcal{K} \mid\!\approx_{RM} \alpha$*, iff either* $\mathcal{K}$ *is inconsistent or* $\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}}) \Vdash \alpha$.

Unsurprisingly, refinement preserves all of the defeasible properties and meta-properties of minimal-model entailment that we care about. In particular, it is defeasible on both the object-level and the meta-level, and satisfies all of the enriched rationality postulates:

Lemma 41. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base, and consider the set $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mathrel{|\!\approx_{RM}} \alpha\}$. Then $\mathcal{S}$ is complete for $\tilde{\mathbb{R}}_{\rightsquigarrow}$.*

As it is based on minimal-model entailment, refined entailment also shares many of the strengths and weaknesses of rational closure. In fact, refinement is a *monotonic* operation, in the sense that any conclusion that can be drawn using minimal-model entailment can also be drawn using refined entailment:

Lemma 42. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base. Then for any formula $\alpha \in \mathcal{L}$, $\mathcal{K} \mathrel{|\!\approx_{M}} \alpha$ implies that $\mathcal{K} \mathrel{|\!\approx_{RM}} \alpha$.*

Refined entailment is therefore the most promising of the three entailment relations we have looked at. It is truly defeasible, it satisfies desirable meta-properties and it sanctions all of the conclusions that we would expect from a "rational closure"-type entailment relation for Defeasible Datalog. This is also true of minimal-model entailment, of course, but refined entailment goes further and solves a number of problems that minimal-model entailment has with ground facts. As a final word on Defeasible Datalog, we will show refined entailment in action and prove that it does indeed solve Example 43:

Example 46. *Consider the knowledge base $\mathcal{K}$ from Example 43:*

$$\mathsf{bird}(\mathsf{Fred})$$
$$\mathsf{fly}(X) \leftsquigarrow \mathsf{bird}(X)$$

We claim that $\mathcal{K} \mathrel{|\!\approx_{RM}} \mathsf{fly}(\mathsf{Fred}) \leftsquigarrow \top$. From Definition 35, it suffices to show that there is some $\mathcal{E} \in \tilde{\mathbb{H}}$ with a maximal set of typical constants $T_{\mathcal{E}} = \{\mathsf{Fred}, \mathsf{E}_1\}$ such that $\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = 0$, and that every such $\mathcal{E}$ satisfies $\mathsf{fly}(\mathsf{Fred})$. For the existence claim, consider $\mathcal{E} = \langle \tilde{\mathcal{H}}_{\mathcal{E}}, \{\mathsf{Fred}, \mathsf{E}_1\}\rangle$, where:

$$\tilde{\mathcal{H}}_{\mathcal{E}} = \{\mathsf{bird}(\mathsf{Fred}), \mathsf{fly}(\mathsf{Fred})\}$$

Now let $\tilde{\mathcal{R}}$ be the ranked interpretation that maps $\mathcal{E}$ to 0, and everything else to ∞. Then $\tilde{\mathcal{R}} \Vdash \mathcal{K}$, and thus $\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = 0$ as required. Next, we show that every such $\mathcal{E}$ satisfies $\mathsf{fly}(\mathsf{Fred})$.

Let $\mathcal{E} \in \tilde{\mathbb{H}}$ be such that $\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = 0$ and $T_{\mathcal{E}} = \{\mathsf{Fred}, \mathsf{E}_1\}$. Then by the first formula in $\mathcal{K}$, $\mathcal{E} \Vdash \mathsf{bird}(\mathsf{Fred})$. But since $\mathsf{Fred} \in T_{\mathcal{E}}$, and since $\mathcal{E}$ has minimal rank in $\tilde{\mathcal{R}}_{\mathcal{K}}$, we conclude that $\mathcal{E} \mathrel{|\!\sim} \mathsf{fly}(X) \leftarrow \mathsf{bird}(X)$ and thus that $\mathcal{E} \Vdash \mathsf{fly}(\mathsf{Fred})$ as required.

Chapter 7

Summary

Our research objective, as stated in Section 1.1, was to investigate analogues of KLM-style inference for Datalog. Realising this goal led to the construction of *Defeasible Datalog*, which was defined and discussed extensively in Chapter 6. In this chapter, we review to what extent our research objective can be considered complete, and suggest a number of possibilities for future research.

7.1 Conclusions

Before we look at Defeasible Datalog specifically, let's first review the major moving parts in any KLM-style description of defeasible reasoning:

1. A *syntax* for expressing classical and defeasible statements. In propositional KLM, this consists of the twiddle operator "$\alpha \mathrel{|\!\sim} \beta$" (see Section 3.1). In defeasible ALC, this consists of the defeasible subsumption operator "$C \sqsubseteq D$" (see Section 4.2).

2. A *semantics* for interpreting classical and defeasible statements. In both propositional KLM and defeasible ALC this is given by various kinds of ranked interpretations (see Definitions 7 and 16).

3. A *soundness and completeness result* that characterises the semantics in terms of a number of *rationality postulates*. For propositional KLM, this is given by Lemmas 9 and 11, and for defeasible ALC it is given by Theorem 6.

4. A *defeasible entailment relation* that describes how inference should work for knowledge bases containing defeasible information. A standard approach is rational closure, which for propositional KLM is given by Definition 10, and for defeasible ALC is given by Definitions 20 (subsumption statements only) and 24 (assertional statements as well).

The syntax of Defeasible Datalog (see the start to Chapter 6) follows that of Morris et al. [28], and extends the syntax of vanilla Datalog in a few ways. The first is that arbitrary boolean combinations of atoms - which we call *compounds* - are allowed as rule heads and bodies. This gives us a lot of flexibility for analysis, but will likely need to be restricted for computational efficiency. The second change is that Defeasible Datalog allows for *defeasible rules* with the twiddle arrow " $\leftsquigarrow$ ", much like the propositional and ALC cases. Nothing seems particularly controversial here.

We have discussed two possible interpretations of Defeasible Datalog formulas. The first is the *preferential semantics* of Morris et al [28], which interprets formulas using rankings of Herbrand interpretations (see Section 6.5 and Definition 30). This was found to be lacking, as it fails to model even fairly simple knowledge bases like the Tweety test (see Lemma 36). Our main contribution in this thesis is the *enriched preferential semantics* of Section 6.6, which solves this problem and admits a characterisation in terms of a number of *rationality postulates* (see Definition 26 and Theorem 8).

One difference between this work on defeasibility and that of the early papers by Kraus et al. [22, 25], is that we explicitly separate the language for Defeasible Datalog formulas and the language for the *defeasible properties* that constitute the rationality postulates (see Section 6.2). While this adds some complexity, we feel like it is worth it for the clarity it offers - many a student (myself included) has been led astray by the conflation of the two!

A soundness and completeness result for enriched preferential semantics is provided by Theorems 8 and 9. We note that in order to capture the semantics precisely we have had to add a number of additional postulates, beyond those proposed by Lehmann et al. [25]. This seems to us to be necessary once one moves beyond the confines of unary languages (the *concepts* of ALC are essentially unary predicates, despite allowing for the use of *roles*). Whether the postulates we have characterised are *useful* or not remains to be seen, but at the very least we have shown that a KLM-style approach can work for Datalog.

Finally, we have provided several options for entailment over Defeasible Datalog knowledge bases (see Definitions 33, 34 and 36). The first two of these, namely *rank entailment* and *minimal-model entailment*, are direct analogues of rank entailment and rational closure in propositional KLM. Though they appear to work well for open formulas (see Example 42), these entailment relations don't handle ground facts very well, a problem that is shared with defeasible ALC in the presence of assertional statements (see Example 26). To deal with this we have introduced *refined entailment*, based on the idea of *refinement* that was introduced in Section 6.8. This seems to solve the ground fact problems, as we show in Example 46.

In conclusion, we have sketched out every moving part of a KLM-style approach to defeasible reasoning for Datalog. Though there are no doubt bugs to be ironed out, this seems like a promising start, and we feel that our original research goal

has essentially been achieved.

7.2 Future Work

The most important line of research that needs to be done is to see how far we can push the *language* of Defeasible Datalog. In this thesis we have focussed on a relatively simple extension of Datalog in which compounds of arbitrary arity are permitted. In order to fully generalise defeasible ALC, and to bring Defeasible Datalog's expressiveness in line with modern flavours of Datalog like Datalog$^{\pm}$ [7], we will need to see what happens when restricted forms of existential quantification are added. Existential quantification is known to cause issues in defeasible ALC (see the discussion in Section 4.3), so we expect that this will be a challenging research problem.

Another important line of research is *computational efficiency*. As we stated at the start of Chapter 6, we have completely ignored the question of efficiency in this thesis, focussing instead on putting all of the formal machinery of defeasible reasoning in place. Now that this is done, it would be interesting to add constraints to the language of Defeasible Datalog such that efficient algorithms for entailment can be found. Minimal-model entailment is a direct analogue of rational closure for propositional KLM, which is reducible to classical entailment checks (see the end of Section 3.5). Refinement is also somewhat analogous to rational closure for defeasible ALC, in the sense that *individuals* are made as typical as possible at each rank-layer. This gives us hope that entailment checking for Defeasible Datalog is similarly tractable.

There are some technical problems that should be addressed here as well. One is that our characterisation of enriched ranked interpretations in terms of rationality postulates (see Theorem 9) requires accepting the (WKIRR) and (WKNEG) properties. These are restricted versions of (IRR) and (NEG) respectively, and it would be interesting to find a semantics that can characterise these latter properties. The preferential semantics of Section 6.5 *satisfies* these properties, but aside from the known limitations of the semantics we have been unable to prove a completeness theorem for it.

A final line of research we mention relates to a possible conceptual bug with what we have done. In enriched preferential semantics, a compound is considered *typical* if and only if every constant in the compound is a typical constant. This simplifies the analysis, but one could argue that typicality for individuals is different from typicality for *tuples* of individuals. Adding a more fine-grained ranking scheme to the enriched Herbrand interpretations of Section 6.6 would produce a semantics similar to that of Brafman [4] and Friedman et al. [17] for first-order conditional logic, and it would be interesting to see if such as approach is workable for Defeasible Datalog as well.

Appendix A

Proofs for Section 6.1

This appendix contains a few technical results about the relationship of Herbrand semantics to standard first-order semantics over the Defeasible Datalog fragment of first-order logic. See Section 6.1 for the definition of *classical consequence*, and Section 2.3 for the definition of *classical entailment*.

Lemma 24. *Let $\mathcal{S} \subseteq \mathcal{L}_f \cup \mathcal{L}_\rightarrow$ be an arbitrary set of formulas. Then the formula α is a classical consequence of $\mathcal{S}$ iff $Tr(\alpha)$ is classically entailed by $Tr(\mathcal{S})$, where Tr is the translation operator of Chapter 5.*

Proof. The "if" direction follows directly from the fact that a Herbrand interpretation is a special case of a general first-order structure. Suppose then that α is a classical consequence of $\mathcal{S}$, and consider some first-order structure $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ such that $\mathcal{I} \Vdash \mathrm{Tr}(\mathcal{S})$. Now let Σ' be an extension of the fixed language that contains a set of constant symbols $\textsc{const}'$ at least as large as $\Delta^\mathcal{I}$, and let $\varphi : \textsc{const}' \rightarrow \Delta^\mathcal{I}$ be any surjection.

We define a Herbrand interpretation $\mathcal{H}$ over Σ' by the following criterion:

$$\mathsf{p}(c_1, \ldots, c_n) \in \mathcal{H} \text{ iff } (\varphi(c_1), \ldots, \varphi(c_n)) \in \mathsf{p}^\mathcal{I}$$

Clearly for any fact $A \in \mathcal{L}_f$ we have that $\mathcal{H} \Vdash A$ iff $\mathcal{I} \Vdash \mathrm{Tr}(A)$, and by extension the same is true of any rule in $\mathcal{L}_\rightarrow$. But then by construction we have that $\mathcal{H} \Vdash \mathcal{S}$, and hence by assumption that $\mathcal{H} \Vdash \alpha$. But this implies that $\mathcal{I} \Vdash \mathrm{Tr}(\alpha)$ as required. □

Corollary 2. *Suppose α is a classical consequence of $\mathcal{S} \subseteq \mathcal{L}_f \cup \mathcal{L}_\rightarrow$. Then there is some finite subset $\mathcal{S}' \subseteq \mathcal{S}$ such that α is a classical consequence of $\mathcal{S}'$.*

Proof. By Lemma 24, α is a classical consequence of $\mathcal{S}$ iff $\mathrm{Tr}(A)$ is classically entailed by $\mathrm{Tr}(\mathcal{S})$. But by the compactness theorem for first-order logic this is true iff $\mathrm{Tr}(\alpha)$ is classically entailed by $\mathrm{Tr}(\mathcal{S}')$ for some finite subset $\mathcal{S}' \subseteq$

$\mathcal{S}$. Another application of Lemma 24 and we conclude that α is a classical consequence of $\mathcal{S}'$. □

Appendix B

Proofs for Section 6.3

The first result we prove is Lemma 25, which establishes that all translation bases are essentially equivalent. This relies on the somewhat unsurprising fact that completeness is invariant under relabelling of meta-atoms for Defeasible Datalog:

Lemma 43. *Let $\sigma : \mathcal{A} \to \mathcal{A}$ be a permutation of meta-atoms, and $\mathcal{S} \subseteq \mathcal{L}$ a set of Defeasible Datalog formulas. Then for any defeasible property $P = I_1, \ldots, I_n \implies I_{n+1} \in \mathbb{P}_{\rightsquigarrow}$, $\mathcal{S}$ is P-complete iff $\mathcal{S}$ is $\sigma(P)$-complete.*

Proof. It suffices to prove one direction of the lemma, as permutations have inverses. Suppose that $\mathcal{S}$ is $\sigma(P)$-complete, and let $\Psi = \langle \lambda_\Psi, \tau_\Psi, \varphi_\Psi, \pi_\Psi, \sigma_\Psi \rangle$ be some substitution system such that $\mathcal{S} \Vdash_\Psi I_1, \ldots, I_n$. Then consider the substitution system $\Psi' = \langle \lambda_\Psi, \tau, \varphi_\Psi, \pi_\Psi, \sigma_\Psi \rangle$, where $\tau : \mathcal{A} \to \mathcal{L}_c$ is defined as follows:

$$\tau(A) = (\tau_\Psi \circ \sigma^{-1})(A)$$

Thus for any $I \in \mathbb{I}_{\rightsquigarrow}$, $\mathcal{S} \Vdash_{\Psi'} \sigma(I)$ iff $\mathcal{S} \Vdash_\Psi I$, as $(\tau \circ \sigma)(A) = \tau_\Psi(A)$ for all $A \in \mathcal{A}$. In particular, $\mathcal{S} \Vdash_{\Psi'} \sigma(I_1), \ldots, \sigma(I_n)$ and hence by $\sigma(P)$-completeness we have that $\mathcal{S} \Vdash_{\Psi'} \sigma(I_{n+1})$. But this implies that $\mathcal{S} \Vdash_\Psi I_{n+1}$, and hence $\mathcal{S}$ is P-complete as required. □

As a side note, for most of this appendix chapter we will exclusively deal with sets $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ of *defeasible rules* when arguing about completeness, rather than general sets of formulas $\mathcal{S} \subseteq \mathcal{L}$. This is because the translation of a propositional inclusion formula $I \in \mathbb{I}_{\mid\sim}$ is always a Defeasible Datalog inclusion schema of the form $B \leftsquigarrow A$ or $\models A$, and satisfaction for these kinds of schemas only depends on defeasible rules. In general, of course, this is not the case.

Lemma 25. *Let B, B' be two translation bases. Then for every propositional defeasible property $P \in \mathbb{P}_{\mid\sim}$, $Tr_B(P)$ is a consequence of $Tr_{B'}(P)$ and vice-versa.*

Proof. Let $\sigma : \mathcal{A} \to \mathcal{A}$ be any permutation of meta-atoms such that $B = \sigma \circ B'$ (at least one exists as B and B' are injective). Then $\mathrm{Tr}_B = \mathrm{Tr}_{\sigma \circ B'} = \sigma \circ \mathrm{Tr}_{B'}$, and hence for any set $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ we have that $\mathcal{S}$ is $\mathrm{Tr}_B(P)$-complete iff it is $\sigma(\mathrm{Tr}_{B'}(P))$-complete. But by Lemma 43 this is true iff $\mathcal{S}$ is $\mathrm{Tr}_{B'}(P)$-complete. □

In Chapter 6, we dealt with translation operators on the meta-level. In other words, translation operators take propositional defeasible properties and produce Defeasible Datalog properties. To prove Theorem 7, we will need to introduce a similar mechanism on the *object* level, allowing us to translate between sets of propositional defeasible *formulas* and sets of Defeasible Datalog *formulas*.

In what follows, we assume that both the set $\mathcal{P}$ of propositional atoms and the set $\mathcal{L}_a$ of Defeasible Datalog atoms are countably infinite. This is not a particularly stringent requirement, as we can simply expand one or the other with dummy atoms if necessary.

Let $b : \mathcal{P} \to \mathcal{L}_a$ be any bijection between the two sets, with inverse denoted $b^{-1} : \mathcal{L}_a \to \mathcal{P}$. Then we can lift these maps to bijections $b_* : \mathcal{L}^{\mathcal{P}} \to \mathcal{L}_c$ and $b_*^{-1} : \mathcal{L}_c \to \mathcal{L}^{\mathcal{P}}$ in the obvious way:

1. If p is atomic, then $b_*(\mathsf{p}) = b(\mathsf{p})$.
2. $b_*(\neg \mathsf{p}) = \neg b_*(\mathsf{p})$.
3. $b_*(\mathsf{p} \wedge \mathsf{q}) = b_*(\mathsf{p}) \wedge b_*(\mathsf{q})$.

These lifted maps behave well with the notion of a *classical tautology*, something that we will make use of later:

Lemma 44. *For any propositional formula $\alpha \in \mathcal{L}^{\mathcal{P}}$, α is a classical tautology iff $b_*(\alpha)$ is a classical tautology.*

Proof. First, suppose that α is a classical tautology, and let $\mathcal{H} \subseteq \mathbb{B}$ be any Herbrand interpretation. Then consider the valuation $u_{\mathcal{H}} \in \mathcal{U}^{\mathcal{P}}$, where for each $p \in \mathcal{P}$ we define $u_{\mathcal{H}} \Vdash p$ iff $\mathcal{H} \Vdash b_*(p)$. By structural induction we have that $u_{\mathcal{H}} \Vdash \alpha$ iff $\mathcal{H} \Vdash b_*(\alpha)$, and since α is assumed to be a classical tautology we conclude that $\mathcal{H} \Vdash b_*(\alpha)$. But $\mathcal{H}$ was arbitrary, so $b_*(\alpha)$ must be a classical tautology as required.

Next, suppose that $b_*(\alpha)$ is a classical tautology, and let $u \in \mathcal{U}^{\mathcal{P}}$ be any valuation. Then consider the Herbrand interpretation $\mathcal{H}_u = \{A \in \mathbb{B} : u \Vdash b_*^{-1}(A)\}$. By structural induction, we have that for any fact $A \in \mathcal{L}_f$, $\mathcal{H}_u \Vdash A$ iff $u \Vdash b_*^{-1}(A)$. Thus if $b_*(\alpha)$ is a fact, we are done, as u was arbitrary. However, it may be the case that $b_*(\alpha)$ contains variable symbols.

In this case, let $B \in \mathcal{L}_f$ be any ground substitution of $b_*(\alpha)$, and note that $\beta = b_*^{-1}(B)$ is identical to α up to a permutation of $\mathcal{P}$. Thus β is a classical tautology iff α is, which must be true by the previous paragraph as $B = b_*(\beta)$ is a fact. □

Given a propositional defeasible formula $\alpha \mathrel{|\!\sim} \beta$, we define its translation under b_* to be the following Defeasible Datalog rule:

$$\mathrm{Tr}_{b_*}(\alpha \mathrel{|\!\sim} \beta) = b_*(\beta) \leftsquigarrow b_*(\alpha)$$

Similarly, the translation of the Defeasible Datalog rule $B \leftsquigarrow A$ under b_*^{-1} is defined to be the following propositional defeasible formula:

$$\mathrm{Tr}_{b_*^{-1}}(B \leftsquigarrow A) = b_*^{-1}(A) \mathrel{|\!\sim} b_*^{-1}(B)$$

Furthermore, these object-level translation operators $\mathrm{Tr}_{b_*} : \mathcal{L}^{|\sim} \to \mathcal{L}_{\rightsquigarrow}$ and $\mathrm{Tr}_{b_*^{-1}} : \mathcal{L}_{\rightsquigarrow} \to \mathcal{L}^{|\sim}$ are both bijections. This follows from the fact that b_* and b_*^{-1} are bijections, and similar reasoning shows that they are actually *inverses*:

$$\Big(\mathrm{Tr}_{b_*^{-1}} \circ \mathrm{Tr}_{b_*}\Big)(\alpha \mathrel{|\!\sim} \beta) = \alpha \mathrel{|\!\sim} \beta$$
$$\Big(\mathrm{Tr}_{b_*} \circ \mathrm{Tr}_{b_*^{-1}}\Big)(B \leftsquigarrow A) = B \leftsquigarrow A$$

Our next goal is to connect these object-level translation operators to the meta-level translation operators defined in Section 6.2. Recall that propositional defeasible properties are interpreted in terms of *substitutions* $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$, and Defeasible Datalog properties are interpreted in terms of *substitution systems* $\Psi = \langle \lambda_\Psi, \tau_\Psi, \varphi_\Psi, \pi_\Psi, \sigma_\Psi \rangle$. The various kinds of translations and mappings we can achieve with these objects is visualised in the following diagram:

$$\begin{array}{ccc} \tilde{\mathcal{P}} & \underset{B_*^{-1}}{\overset{B_*}{\rightleftarrows}} & B_*(\tilde{\mathcal{P}}) \subseteq \mathcal{T} \\ \Big\downarrow{\scriptstyle \varphi} & & \Big\downarrow{\scriptstyle \Psi} \\ \mathcal{L}^{\mathcal{P}} & \underset{b_*^{-1}}{\overset{b_*}{\rightleftarrows}} & \mathcal{L}_c \end{array}$$

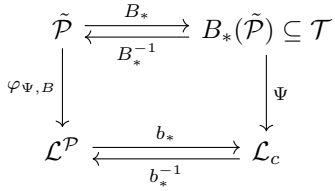
Note that this is *not* a commutative diagram, as different paths through the diagram will generally result in different results. Nevertheless, we can see that the combination of an object-level translation operator and a meta-level translation operator allows us to move freely between propositional substitutions and Defeasible Datalog substitution systems:

Definition 37. *Let B be a translation base, and Ψ a substitution system. Then we define the substitution $\varphi_{\Psi,B} : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ as follows:*

$$\varphi_{\Psi,B} = b_*^{-1} \circ \Psi \circ B_*$$

As a direct consequence, we have the following result:

Lemma 45. *Let B be a translation base, and Ψ a substitution system. Then the following diagram commutes:*



Proof. Follows directly from the definition of $\varphi_{\Psi,B}$. □

Similarly, we can go in the other direction and construct a substitution system for any given choice of substitution:

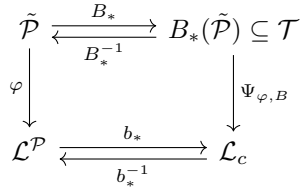
Definition 38. *Let B be a translation base, and $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ a substitution. Then we define the substitution system $\Psi_{\varphi,B} = \langle \lambda_{\varphi,B}, \tau_{\varphi,B}, id, id, id \rangle$ as follows:*

$$\lambda_{\varphi,B}(v) = \begin{cases} \text{VAR} & \text{if } v = \vec{x} \\ \emptyset & \text{otherwise} \end{cases}$$

$$\tau_{\varphi,B}(A) = \begin{cases} \Big(b_* \circ \varphi\Big)(\tilde{p}) & \text{if } A = B(\tilde{p}) \text{ for some } \tilde{p} \in \tilde{\mathcal{P}} \\ \bot & \text{otherwise} \end{cases}$$

Note that the construction of $\tau_{\varphi,B}$ is well-defined, as B is injective by definition. And as before, we have the following commutativity result:

Lemma 46. *Let B be a translation base, and φ a substitution. Then the following diagram commutes:*



Proof. It suffices to show that $\varphi = b_*^{-1} \circ \Psi_{\varphi,B} \circ B_*$. Indeed, consider some $\tilde{p} \in \tilde{\mathcal{P}}$. Then $B(\tilde{p}) = A$ for some meta-atom $A \in \mathcal{A}$, and hence by definition $\tau_{\varphi,B}(A) = (b_* \circ \varphi)(\tilde{p})$. But then $B_*(\tilde{p}) = A(\vec{x})$, and since $\Psi_{\varphi,B}$ is compatible with $A(\vec{x})$ we have that $(\Psi_{\varphi,B} \circ B_*)(\tilde{p}) = \tau_{\varphi,B}(A) = (b_* \circ \varphi)(\tilde{p})$. We conclude that $(b_*^{-1} \circ \Psi_{\varphi,B} \circ B_*)(\tilde{p}) = b_*^{-1}\big((b_* \circ \varphi)(\tilde{p})\big) = \varphi(\tilde{p})$ as required. □

We now have all of the definitions and concepts we need to prove Theorem 7. We will build up the proof in stages, starting from equivalences with respect to inclusion formula satisfaction:

Lemma 47. *Let $I \in \mathbb{I}_{\mid\sim}$ be a propositional inclusion formula, and $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ a set of Defeasible Datalog rules. Then for any substitution system Ψ, $\mathcal{S} \Vdash_\Psi Tr_B(I)$ iff $Tr_b^{-1}(\mathcal{S}) \Vdash_{\varphi_{\Psi,B}} I$.*

Proof. For any such Ψ, we consider each possibility for I separately:

1. Suppose that $I =$"$\models \alpha$". Then $\mathcal{S} \Vdash_\Psi \mathrm{Tr}_B(I)$ iff $(\Psi \circ B_*)(\alpha)$ is a classical tautology, which by Lemma 44 is true iff $(b_*^{-1} \circ \Psi \circ B_*)(\alpha) = \varphi_{\Psi,B}(\alpha)$ is a classical tautology. But by definition this is true iff $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_{\varphi_{\Psi,B}} I$.

2. Suppose that $I =$"$\alpha \mathrel{|\!\sim} \beta$". Then $\mathcal{S} \Vdash_\Psi \mathrm{Tr}_B(I)$ iff $(\Psi \circ B_*)(\beta) \mathrel{\reflectbox{$\rightsquigarrow$}} (\Psi \circ B_*)(\alpha) \in \mathcal{S}$, which is itself true iff $\mathrm{Tr}_b^{-1}(\mathcal{S})$ contains $(b_*^{-1} \circ \Psi \circ B_*)(\alpha) \mathrel{|\!\sim} (b_*^{-1} \circ \Psi \circ B_*)(\beta) = \varphi_{\Psi,B}(\alpha) \mathrel{|\!\sim} \varphi_{\Psi,B}(\beta)$, and hence iff $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_{\varphi_{\Psi,B}} \alpha \mathrel{|\!\sim} \beta = I$.

3. The reasoning for $I =$"$\alpha \not\mathrel{|\!\sim} \beta$" is identical to the previous case.

□

The result holds in the other direction as well:

Lemma 48. *Let $I \in \mathbb{I}_{|\sim}$ be a propositional inclusion formula, and $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ a set of Defeasible Datalog rules. Then for any substitution $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$, $\mathcal{S} \Vdash_{\Psi_{\varphi,B}} \mathrm{Tr}_B(I)$ iff $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_\varphi I$.*

Proof. We consider each possibility for I separately:

1. Suppose that $I =$"$\models \alpha$". Then $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_\varphi I$ iff $\varphi(\alpha)$ is a classical tautology, which by Lemma 44 is true iff $(b_* \circ \varphi)(\alpha)$ is a classical tautology. But by Lemma 46 this is true iff $(\Psi_{\varphi,B} \circ B_*)(\alpha)$ is a classical tautology, and hence by definition iff $\mathcal{S} \Vdash_{\Psi_{\varphi,B}} \mathrm{Tr}_B(I)$.

2. Suppose that $I =$"$\alpha \mathrel{|\!\sim} \beta$". Then $\mathcal{S} \Vdash_{\Psi_{\varphi,B}} \mathrm{Tr}_B(I)$ iff $(\Psi_{\varphi,B} \circ B_*)(\beta) \mathrel{\reflectbox{$\rightsquigarrow$}} (\Psi_{\varphi,B} \circ B_*)(\beta) \in \mathcal{S}$, and hence iff $\mathrm{Tr}_b^{-1}(\mathcal{S})$ contains $(b_*^{-1} \circ \Psi_{\varphi,B} \circ B_*)(\alpha) \mathrel{|\!\sim} (b_*^{-1} \circ \Psi_{\varphi,B} \circ B_*)(\beta) = \varphi(\alpha) \mathrel{|\!\sim} \varphi(\beta)$. But by Lemma 46 this is true iff $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_\varphi \alpha \mathrel{|\!\sim} \beta = I$.

3. The reasoning for $I =$"$\alpha \not\mathrel{|\!\sim} \beta$" is identical to the previous case.

□

Next, we prove equivalences for the case of individual defeasible properties:

Lemma 49. *Let $P = I_1, \ldots, I_n \implies I_{n+1} \in \mathbb{P}_{|\sim}$ be a propositional defeasible property, and $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ a set of Defeasible Datalog rules. Then $\mathcal{S}$ is $\mathrm{Tr}_B(P)$-complete iff $\mathrm{Tr}_b^{-1}(\mathcal{S})$ is P-complete.*

Proof. First suppose that $\mathcal{S}$ is $\mathrm{Tr}_B(P)$-complete, and let $\varphi : \tilde{\mathcal{P}} \to \mathcal{L}^{\mathcal{P}}$ be any substitution such that $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_\varphi I_1, \ldots, I_n$. By Lemma 48, this implies that $\mathcal{S} \Vdash_{\Psi_{\varphi,B}} \mathrm{Tr}_B(I_1), \ldots, \mathrm{Tr}_B(I_n)$, and hence by hypothesis that $\mathcal{S} \Vdash_{\Psi_{\varphi,B}}$

$\mathrm{Tr}_B(I_{n+1})$. But then by Lemma 48 we have that $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_\varphi I_{n+1}$, and since φ was arbitrary we conclude that $\mathrm{Tr}_b^{-1}(\mathcal{S})$ is P-complete.

Next suppose that $\mathrm{Tr}_b^{-1}(\mathcal{S})$ is P-complete, and let Ψ be any substitution system such that $\mathcal{S} \Vdash_\Psi \mathrm{Tr}_B(I_1), \ldots, \mathrm{Tr}_B(I_n)$. By Lemma 47, this implies that $\mathrm{Tr}_b^{-1}(\mathcal{S}) \Vdash_{\varphi_{\Psi,B}} I_1, \ldots, I_n$, and hence by hypothesis that $\mathcal{S} \Vdash_{\varphi_{\Psi,B}} I_{n+1}$. But then by Lemma 47 we have that $\mathcal{S} \Vdash_\Psi \mathrm{Tr}_B(I_{n+1})$, and since Ψ was arbitrary we conclude that $\mathcal{S}$ is $\mathrm{Tr}_B(P)$-complete. □

Finally, we move on to the proof of the theorem itself:

Theorem 7. *Suppose that $D \subseteq \mathbb{P}_{\mathrel{|\sim}}$ is a set of propositional defeasible properties, and that $P \in \mathbb{P}_{\mathrel{|\sim}}$ is a consequence of D. Then for any translation base B, $Tr_B(P)$ is a consequence of $Tr_B(D)$.*

Proof. Let $\mathcal{S} \subseteq \mathcal{L}_{\rightsquigarrow}$ be any $\mathrm{Tr}_B(D)$-complete set of Defeasible Datalog rules. Then by Lemma 49, $\mathrm{Tr}_b^{-1}(\mathcal{S})$ is D-complete, and hence P-complete by hypothesis. But then by another application of Lemma 49 we have that $\mathcal{S}$ is $\mathrm{Tr}_B(P)$-complete. Since $\mathcal{S}$ was arbitrary, we conclude that $\mathrm{Tr}_B(P)$ is a consequence of $\mathrm{Tr}_B(D)$. □

To prove the corollary, we describe an explicit transformation base that takes the propositional rationality postulates to the Defeasible Datalog rationality postulates:

Corollary 3. *Let B be a translation base. Then if $P \in \mathbb{P}_{\mathrel{|\sim}}$ is a consequence of the propositional rationality postulates, $Tr_B(P)$ is a consequence of the Defeasible Datalog rationality postulates.*

Proof. We refer the reader to Chapter 3, Section 3.2 for definitions of the propositional rationality postulates, and Chapter 6, Section 6.2 for definitions of the Defeasible Datalog rationality postulates.

By Lemma 25, we can without loss of generality take $B : \tilde{\mathcal{P}} \to \mathcal{A}$ to be any translation base for which $\alpha \mapsto A$, $\beta \mapsto B$ and $\gamma \mapsto C$. Then one can check that $\mathrm{Tr}_B : \mathbb{P}_{\mathrel{|\sim}} \to \mathbb{P}_{\rightsquigarrow}$ maps the propositional versions of (Rm), (Rw), (And), (Or) and (Rm) onto their Defeasible Datalog counterparts. The only problematic case is therefore (Lle).

Tr_B maps (Lle) instead to the Defeasible Datalog property $C(\vec{x}) \mathrel{\reflectbox{$\rightsquigarrow$}} A(\vec{x}),\ \models A(\vec{x}) \leftrightarrow B(\vec{x}) \implies C(\vec{x}) \rightsquigarrow B(\vec{x})$. While this isn't equal to the Defeasible Datalog version of (Lle), we note that the two are equivalent in the sense that each is a consequence of the other. The required claim follows directly from Theorem 7 under the translation base B. □

This corollary can be used to prove a number of nice consequences of the Defeasible Datalog rationality postulates directly:

Lemma 26. *$\mathbb{R}_{\curvearrowright}$ has as consequence the following defeasible properties:*

1. $\bot \mathrel{\curvearrowright} A(\vec{x}) \implies \neg A(\vec{x}) \mathrel{\curvearrowright} A(\vec{x})$
2. $\neg A(\vec{x}) \mathrel{\curvearrowright} A(\vec{x}) \vee B(\vec{x}),\ \neg B(\vec{x}) \mathrel{\curvearrowright} B(\vec{x}) \vee C(\vec{x}) \implies \neg A(\vec{x}) \mathrel{\curvearrowright} A(\vec{x}) \vee C(\vec{x})$
3. $\bot \mathrel{\curvearrowright} A(\vec{x}),\ \neg A(\vec{x}) \mathrel{\curvearrowleft} A(\vec{x}) \vee B(\vec{x}) \implies \neg B(\vec{x}) \mathrel{\curvearrowright} A(\vec{x}) \vee B(\vec{x})$

Proof. By Lemma 9 and a result of Lehmann *et al.* [25, p. 8], a propositional defeasible property is a consequence of $\mathbb{R}_{\mid\sim}$ iff it holds in every modular interpretation. Thus by Corollary 3, it suffices to show that the following properties hold in every modular interpretation:

1. $\alpha \not\mid\sim \bot \implies \alpha \not\mid\sim \neg\alpha$
2. $\alpha \vee \beta \not\mid\sim \neg\alpha,\ \beta \vee \gamma \not\mid\sim \neg\beta \implies \alpha \vee \gamma \not\mid\sim \neg\alpha$
3. $\alpha \not\mid\sim \bot,\ \alpha \vee \beta \mid\sim \neg\alpha \implies \alpha \vee \beta \not\mid\sim \neg\beta$

For the first, let $\mathcal{M}$ be a modular interpretation such that $\mathcal{M} \Vdash \alpha \not\mid\sim \bot$. Then $\min_{\mathcal{M}} \widehat{\alpha}$ must be non-empty, as otherwise $\mathcal{M}$ would vacuously satisfy $\alpha \mid\sim \bot$. But this implies that $\mathcal{M} \Vdash \alpha \not\mid\sim \neg\alpha$.

The second follows directly from the fact that for any modular interpretation $\mathcal{M}$, $\mathcal{M} \Vdash \alpha \vee \beta \not\mid\sim \neg\alpha$ iff $\mathcal{M}(u) \leq \mathcal{M}(v)$ for some $u \in \min_{\mathcal{M}} \widehat{\alpha}$ and $v \in \min_{\mathcal{M}} \widehat{\beta}$.

For the third, let $\mathcal{M}$ be a modular interpretation such that $\mathcal{M} \Vdash \alpha \not\mid\sim \bot$ and $\mathcal{M} \Vdash \alpha \vee \beta \mid\sim \neg\alpha$. Then from the result in the previous case and the fact that $\min_{\mathcal{M}} \widehat{\alpha} \neq \emptyset$, this implies that $\mathcal{M}(v) < \mathcal{M}(u)$ for some $v \in \min_{\mathcal{M}} \widehat{\beta}$ and $u \in \min_{\mathcal{M}} \widehat{\alpha}$. But this in turn implies that $\mathcal{M} \Vdash \alpha \vee \beta \not\mid\sim \neg\beta$. □

Appendix C

Proofs for Section 6.4

We remind the reader that Section 6.4 contains all relevant definitions for this section, such as that of a substitution system $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$, instances of meta-arguments $\Psi_{\vec{x},\vec{y}} : \textsc{var} \to \textsc{var}$, and the notion of satisfaction with respect to a substitution system $\mathcal{S} \Vdash_\Psi B(\vec{x}) \leftsquigarrow A(\vec{x})$.

We also note one possible source of confusion, which is that we will often use the same symbol "A" to refer both to a meta-atom $A \in \mathcal{A}$, as well as a Defeasible Datalog compound $A \in \mathcal{L}_c$. This makes it convenient to define maps $\tau : \mathcal{A} \to \mathcal{L}_c$ when constructing substitution systems, as we can simply write $\tau(A) = A$, with the understanding that the left-hand side is a meta-atom and the right-hand side a compound (as should be clear from the type signature of τ).

Lemma 27. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a* (Per)*-complete set of formulas, and suppose that $B \leftsquigarrow A \in \mathcal{S}$. Then for any permutation $\sigma : \textsc{var} \to \textsc{var}$, we have that $\sigma(B) \leftsquigarrow \sigma(A) \in \mathcal{S}$.*

Proof. Let $\textsc{var}_{A,B}$ be the set of variable symbols in the compounds A and B. Then consider the substitution system $\Psi = \langle \lambda, \tau, \mathrm{id}, \mathrm{id}, \sigma \rangle$, where:

1. $\lambda(A) = A,\ \lambda(B) = B$
2. $\tau(\vec{x}) = \textsc{var}_{A,B}$

Then by construction we have that $\Psi_{\sigma\vec{x}}(A) = \sigma(A)$ and $\Psi_{\sigma\vec{x}}(B) = \sigma(B)$, and that $\mathcal{S} \Vdash_\Psi B(\vec{x}) \leftsquigarrow A(\vec{x})$. By (Per)-completeness this implies that $\mathcal{S} \Vdash_\Psi B(\sigma\vec{x}) \leftsquigarrow A(\sigma\vec{x})$, and hence that $\Psi_{\sigma\vec{x}}(B) \leftsquigarrow \Psi_{\sigma\vec{x}}(A) = \sigma(B) \leftsquigarrow \sigma(A) \in \mathcal{S}$. □

Lemma 28. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a* (Irr)*-complete set of formulas, and suppose that $B \leftsquigarrow A \in \mathcal{S}$, with $\textsc{var}_A$ and $\textsc{var}_B$ the free variables in A and B respectively. Then for any substitution $\varphi : \textsc{var} \to \textsc{var} \cup \textsc{const}$ that is constant on $\textsc{var}_A$, we have that $\varphi(B) \leftsquigarrow A \in \mathcal{S}$.*

Proof. Consider the substitution system $\Psi = \langle \lambda, \tau, \varphi, \mathrm{id}, \mathrm{id} \rangle$, where:

1. $\lambda(A) = A,\ \lambda(B) = B$
2. $\tau(\vec{x}) = \mathrm{VAR}_A,\ \tau(\vec{y}) = \mathrm{VAR}_B \setminus \mathrm{VAR}_A$

Then by construction we have that $\Psi_{\vec{x}}(A) = A$ and $\Psi_{\vec{x},\varphi\vec{y}}(B) = \varphi(B)$, and that $\mathcal{S} \Vdash_\Psi B(\vec{x},\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$. By (Irr)-completeness this implies that $\mathcal{S} \Vdash_\Psi B(\vec{x},\varphi\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$, and hence that $\Psi_{\vec{x},\varphi\vec{y}}(B) \mathrel{\sim\!\!\!\!\!\sim} \Psi_{\vec{x}}(A) = \varphi(B) \mathrel{\sim\!\!\!\!\!\sim} A \in \mathcal{S}$. □

Lemma 29. (WkIrr) *is a consequence of* (Irr).

Proof. Suppose $\mathcal{S} \subseteq \mathcal{L}$ is (Irr)-complete, and that $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$ is a substitution system such that $\mathcal{S} \Vdash_\Psi B(\vec{x},\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$. Now consider the modified substitution system $\Psi' = \langle \lambda, \tau, \varphi', \pi, \sigma \rangle$, where $\varphi' = \pi$. Then we still have that $\mathcal{S} \Vdash_{\Psi'} B(\vec{x},\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$, and hence by (Irr)-completeness that $\mathcal{S} \Vdash_{\Psi'} B(\vec{x},\varphi\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$. By construction, this is equivalent to $\mathcal{S} \Vdash_\Psi B(\vec{x},\pi\vec{y}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$, and since Ψ was arbitrary we conclude that $\mathcal{S}$ is (WkIrr)-complete. □

Lemma 30. (Per) *is a consequence of* (WkUI)*, which is in turn a consequence of* (UI).

Proof. Consider any $\mathcal{S} \subseteq \mathcal{L}$ and $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$ such that $\mathcal{S} \Vdash_\Psi B(\vec{x}) \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$. Then (UI)-completeness, (WkUI)-completeness and (Per)-completeness would imply that $\mathcal{S} \Vdash_\Psi A(\varphi\vec{x}) \mathrel{\sim\!\!\!\!\!\sim} B(\varphi\vec{x})$, $\mathcal{S} \Vdash_\Psi A(\pi\vec{x}) \mathrel{\sim\!\!\!\!\!\sim} B(\pi\vec{x})$ and $\mathcal{S} \Vdash_\Psi A(\sigma\vec{x}) \mathrel{\sim\!\!\!\!\!\sim} B(\sigma\vec{x})$ respectively. Thus the claim follows immediately from the fact that every permutation substitution σ is a special case of a variable substitution π, and every variable substitution π is a special case of a general substitution φ. □

Lemma 31. (WkNeg) *is a consequence of* (Neg).

Proof. Suppose $\mathcal{S} \subseteq \mathcal{L}$ is (Neg)-complete, and $\Psi = \langle \lambda, \tau, \varphi, \pi, \sigma \rangle$ is any substitution system such that $\mathcal{S} \Vdash_\Psi \forall\varphi.\bot \mathrel{\sim\!\!\!\!\!\sim} \neg A(\varphi\vec{x})$. By definition this means that $\mathcal{S} \Vdash_{\Psi'} \bot \mathrel{\sim\!\!\!\!\!\sim} \neg A(\varphi\vec{x})$ for any substitution system $\Psi' = \langle \lambda, \tau, \varphi', \pi, \sigma \rangle$ that differs from Ψ only in the choice of general substitution φ'. In particular, taking φ' to be the identity substitution we therefore have that $\mathcal{S} \Vdash_{\Psi'} \bot \mathrel{\sim\!\!\!\!\!\sim} A(\vec{x})$, and hence by (Neg)-completeness that $\mathcal{S} \Vdash_{\Psi'} A(\vec{x})$. But this inclusion schema contains no substitution symbols, and thus $\mathcal{S} \Vdash_\Psi A(\vec{x})$ as well, implying that $\mathcal{S}$ is (WkNeg)-complete. □

Lemma 32. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a set of Defeasible Datalog formulas, and let $\mathcal{S}_C$ denote its classical subset. Then $\mathcal{S}$ is complete for* (Mp)*,* (Eq) *and* (EqF) *iff $\mathcal{S}_C$ is deductively closed.*

Proof. First, assume that $\mathcal{S}$ is deductively closed. Then the fact that it is complete for (Mp), (Eq) and (EqF) follows from these respective observations, where φ is any substitution:

1. $\varphi(B)$ is always a classical consequence of $\varphi(A)$ and $B \leftarrow A$
2. $\neg A \vee B$ is always a classical consequence of $B \leftarrow A$
3. $B \leftarrow A$ is always a classical consequence of $\neg A \vee B$

In the other direction, suppose that $\mathcal{S}$ is complete for (Mp), (Eq) and (EqF), and that α is a classical consequence of $\mathcal{S}_C$. Then by Corollary 2 there is some finite subset $\mathcal{S}'_C = \{\beta_1, \ldots, \beta_n\} \subseteq \mathcal{S}_C$ such that α is a classical consequence of $\mathcal{S}'_C$. Furthermore, by (Mp)-completeness we can assume wlog that no two rules in $\mathcal{S}'_C$ share a variable symbol. Now consider the transformation $(\,\cdot\,)^*$, acting on a fact or rule:

1. $(A)^* = A$
2. $(B \leftarrow A)^* = \neg A \vee B$

Then by Lemma 24 and the deduction theorem for first-order logic, the following rule is a classical tautology:

$$\neg(\beta_2^* \wedge \cdots \wedge \beta_n^*) \vee \alpha^* \leftarrow \beta_1^*$$

But by completeness for (Eq) and (EqF), $\varphi(\beta_1^*) \in \mathcal{S}$ for every substitution φ, and thus by (Mp)-completeness every substitution of the left-hand side is similarly in $\mathcal{S}$. The left hand side is classically equivalent to the following rule:

$$\neg(\beta_3^* \wedge \cdots \wedge \beta_n^*) \vee \alpha^* \leftarrow \beta_2^*$$

Thus by iterating this process we eventually conclude that $\varphi(\alpha^*) \in \mathcal{S}$ for every substitution φ. But then by completeness for (Eq) and (EqF) this implies that $\alpha \in \mathcal{S}$, and hence $\mathcal{S}_C$ is deductively closed as required. □

Appendix D

Proofs for Section 6.5

In this appendix we prove that the preferential semantics of Section 6.5 satisfies a number of desirable defeasible properties, and provide a counterexample illustrating its representational problems. First, we look at the rationality postulates from Section 6.2:

Lemma 33. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ be a ranked interpretation. Then $\mathcal{S}_\mathcal{R}$ is $\mathbb{R}_{\sim}$-complete.*

Proof. We consider each case separately:

(Refl) Follows from the fact that $\mathcal{H} \Vdash A \leftarrow A$ for all $\mathcal{H} \in \mathbb{H}$.

(Lle) Follows from the fact that if $A \leftarrow B$ and $B \leftarrow A$ are both classical tautologies, then $\widehat{A} = \widehat{B}$.

(Rw) Follows from the fact that if $C \leftarrow B$ is a classical tautology, $\mathcal{H} \in \mathbb{H}$ and $\mathcal{H} \Vdash B \leftarrow A$, then $\mathcal{H} \Vdash C \leftarrow A$.

(And) Follows from the fact that for any $\mathcal{H} \in \mathbb{H}$, $\mathcal{H} \Vdash B \leftarrow A$ and $\mathcal{H} \Vdash C \leftarrow A$ implies that $\mathcal{H} \Vdash B \wedge C \leftarrow A$.

(Or) Suppose that $\mathcal{R} \Vdash C \leftsquigarrow A$ and $\mathcal{R} \Vdash C \leftsquigarrow B$, and consider some $\mathcal{H} \in \min_\mathcal{R} \widehat{A \vee B}$. Then for any ground subsitution φ, $\mathcal{H} \Vdash \varphi(A)$ implies that $\mathcal{H} \in \min_\mathcal{R} \widehat{A}$ and hence $\mathcal{H} \Vdash C \leftarrow A$, and $\mathcal{H} \Vdash \varphi(B)$ implies that $\mathcal{H} \in \min_\mathcal{R} \widehat{B}$ and hence $\mathcal{H} \Vdash C \leftarrow B$. Since at least one of these has to be the case, $\mathcal{H} \Vdash C \leftarrow A \vee B$ and hence $\mathcal{R} \Vdash C \leftsquigarrow A \vee B$.

(Rm) Suppose that $\mathcal{R} \Vdash C \leftsquigarrow A$ and $\mathcal{R} \nVdash \neg B \leftsquigarrow A$. Then there is some $\mathcal{H} \in \min_\mathcal{R} \widehat{A}$ such that $\mathcal{H} \in \widehat{B}$. But this implies that $\min_\mathcal{R} \widehat{A \wedge B} \subseteq \min_\mathcal{R} \widehat{A}$ and hence $\mathcal{R} \Vdash C \leftsquigarrow A \wedge B$.

□

Next, we consider some of the properties discussed in Section 6.4:

Lemma 34. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ be a ranked interpretation. Then $\mathcal{S}_\mathcal{R}$ is complete for* (Per)*,* (Irr)*,* (Neg)*,* (Sup)*,* (SupF)*,* (Mp)*,* (Eq) *and* (EqF)*.*

Proof. We consider each case separately:

(Per) Follows from the fact that for any $\mathcal{H} \in \mathbb{H}$ and permutation $\sigma : \text{VAR} \to \text{VAR}$, $\mathcal{H} \Vdash B \leftarrow A$ iff $\mathcal{H} \Vdash \sigma(B) \leftarrow \sigma(A)$.

(Irr) Follows from the fact that for any $\mathcal{H} \in \widehat{A}$, $\mathcal{H} \Vdash B \leftarrow A$ iff $\mathcal{H} \Vdash \varphi(B) \leftarrow A$ for every substitution φ that is constant on the variable symbols of A.

(Neg) Follows from the fact that if $\mathcal{R} \Vdash \bot \mathrel{\sim\!\!\!\!\!\!\sim} \neg A$, then $\mathcal{H} \Vdash A$ for every $\mathcal{H} \in \mathbb{H}^\mathcal{R}$ and hence $\mathcal{R} \Vdash A$.

(Sup) Follows from the fact that $\mathcal{R} \Vdash B \leftarrow A$ iff $\mathcal{H} \Vdash B \leftarrow A$ for every $\mathcal{H} \in \mathbb{H}^\mathcal{R}$.

(SupF) Follows from the fact that $\mathcal{R} \Vdash A$ iff $\mathcal{H} \Vdash \bot \leftarrow \neg A$ for every $\mathcal{H} \in \mathbb{H}^\mathcal{R}$.

(Mp) Follows from the fact that if $B \leftarrow A$ is a classical tautology, then $\varphi(B)$ is a classical consequence of $\varphi(A)$ for any substitution φ.

(Eq) Follows from the fact that $B \leftarrow A$ and $\neg A \vee B$ are classically equivalent.

(EqF) Follows from the fact that $\neg A \vee B$ and $B \leftarrow A$ are classically equivalent.

□

The last property we look at seems to be common to all kinds of preferential-style semantics:

Lemma 35. *Let $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ be a ranked interpretation. Then $\mathcal{S}_\mathcal{R}$ is complete for the following defeasible property:*

$$\text{(Pref)} \quad \bot \mathrel{\sim\!\!\!\!\!\!\sim} A(\vec{x}) \implies \neg A(\vec{x}) \mathrel{\sim\!\!\!\!\!\!\sim} A(\vec{x}) \vee A(\pi\vec{x})$$

Proof. Suppose $\mathcal{R} \nVdash \bot \mathrel{\sim\!\!\!\!\!\!\sim} A$, and consider some variable substitution $\pi : \text{VAR} \to \text{VAR}$ and $\mathcal{H} \in \min_\mathcal{R} \widehat{A \vee \pi(A)}$. Then $\mathcal{H}$ necessarily satisfies some instance of A, because any instance of $\pi(A)$ is also an instance of A, and hence $\mathcal{H} \in \min_\mathcal{R} \widehat{A}$. Thus $\mathcal{H} \nVdash \neg A \leftarrow A \vee \pi(A)$, and hence $\mathcal{R} \nVdash \neg A \mathrel{\sim\!\!\!\!\!\!\sim} A \vee \pi(A)$ as required. □

Finally, we show that preferential semantics is not expressive enough as it stands to model the Tweety test knowledge base of Section 6.5:

Lemma 36. *There is no non-trivial ranked interpretation $\mathcal{R}$ over any first-order language satisfying $\mathcal{K}_{\mathsf{tweety}}$.*

Proof. Suppose that $\mathcal{R} : \mathbb{H} \to \mathbb{N}^\infty$ is a non-trivial ranked interpretation satisfying $\mathcal{K}_{\mathsf{tweety}}$. Then $\mathbb{H}^\mathcal{R}$ is non-empty, and every $\mathcal{H} \in \mathbb{H}^\mathcal{R}$ satisfies $\mathsf{bird}(\mathsf{Tweety}) \wedge \neg\mathsf{fly}(\mathsf{Tweety})$. But this implies that no $\mathcal{H} \in \mathbb{H}^\mathcal{R}$ satisfies $\mathsf{fly}(X) \leftarrow \mathsf{bird}(X)$, which directly from the definitions proves that $\mathcal{R} \not\Vdash \mathsf{fly}(X) \leftsquigarrow \mathsf{bird}(X)$, a contradiction. □

Appendix E

Proofs for Section 6.6

This appendix contains the proofs of Theorems 8 (soundness) and 9 (completeness) for enriched ranked interpretations with respect to the enriched rationality postulates $\tilde{\mathbb{R}}_{\rightsquigarrow}$ of Section 6.6. Soundness is the easier direction to establish:

Theorem 8. *Let $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^{\infty}$ be an enriched ranked interpretation. Then $\mathcal{S}_{\tilde{\mathcal{R}}}$ is $\mathbb{R}_{\rightsquigarrow}$-complete, and also complete for* (Pref)*,* (Sup)*,* (SupF)*,* (Per)*,* (WkIrr)*,* (WkNeg)*,* (Mp)*,* (Eq) *and* (EqF)*.*

Proof. We consider each case separately:

(Refl) Follows from the fact that $\mathcal{E} \mid\!\sim A \leftarrow A$ for all $\mathcal{E} \in \tilde{\mathbb{H}}$.

(Lle) Follows from the fact that if $A \leftarrow B$ and $B \leftarrow A$ are both classical tautologies, then $\tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A) = \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(B)$.

(Rw) Follows from the fact that if $C \leftarrow B$ is a classical tautology, $\mathcal{E} \in \tilde{\mathbb{H}}$ and $\mathcal{E} \mid\!\sim B \leftarrow A$, then $\mathcal{E} \mid\!\sim C \leftarrow A$.

(And) Follows from the fact that for any $\mathcal{E} \in \tilde{\mathbb{H}}$, $\mathcal{E} \mid\!\sim B \leftarrow A$ and $\mathcal{E} \mid\!\sim C \leftarrow A$ implies that $\mathcal{E} \mid\!\sim B \wedge C \leftarrow A$.

(Or) Suppose that $\tilde{\mathcal{R}} \Vdash C \leftsquigarrow A$ and $\tilde{\mathcal{R}} \Vdash C \leftsquigarrow B$, and consider some $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A \vee B)$. Then for any $\varphi : \textsc{var} \to T_{\mathcal{E}}$, $\mathcal{E} \Vdash \varphi(A)$ implies that $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ and hence $\mathcal{E} \mid\!\sim C \leftarrow A$, and $\mathcal{E} \Vdash \varphi(B)$ implies that $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(B)$ and hence $\mathcal{E} \mid\!\sim C \leftarrow B$. Since at least one of these has to be the case, $\mathcal{E} \mid\!\sim C \leftarrow A \vee B$ and hence $\tilde{\mathcal{R}} \Vdash C \leftsquigarrow A \vee B$.

(Rm) Suppose that $\tilde{\mathcal{R}} \Vdash C \leftsquigarrow A$ and $\tilde{\mathcal{R}} \nVdash \neg B \leftsquigarrow A$. Then there is some $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ such that $\mathcal{E} \in \tilde{\mathbb{H}}^{\mathcal{R}}(B)$. But this implies that $\min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A \wedge B) \subseteq \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$, and thus $\tilde{\mathcal{R}} \Vdash C \leftsquigarrow A \wedge B$.

(PREF) Suppose $\tilde{\mathcal{R}} \not\Vdash \bot \leftsquigarrow A$, and consider some variable substitution $\pi : \text{VAR} \to \text{VAR}$ and $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A \vee \pi(A))$. Then $\mathcal{E}$ necessarily satisfies $\varphi(A)$ for some $\varphi : \text{VAR} \to T_{\mathcal{E}}$, because any instance of $\pi(A)$ is also an instance of A, and hence $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$. Thus $\mathcal{E} \not\sim \neg A \leftarrow A \vee \pi(A)$, and hence $\tilde{\mathcal{R}} \not\Vdash A \leftsquigarrow A \vee \pi(A)$ as required.

(SUP) Follows from the fact that $\mathcal{E} \Vdash B \leftarrow A$ implies $\mathcal{E} \mid\sim B \leftarrow A$.

(SUPF) Follows from the fact that $\mathcal{E} \Vdash A$ implies $\mathcal{E} \mid\sim \bot \leftarrow \neg A$.

(PER) Follows from the fact that for any $\mathcal{E} \in \tilde{\mathbb{H}}$ and permutation $\sigma : \text{VAR} \to \text{VAR}$, $\mathcal{E} \mid\sim B \leftarrow A$ iff $\mathcal{E} \mid\sim \sigma(B) \leftarrow \sigma(A)$.

(WKIRR) Follows from the fact that for any $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$, $\mathcal{E} \mid\sim B \leftarrow A$ iff $\mathcal{E} \mid\sim \pi(B) \leftarrow A$ for every variable substitution $\pi : \text{VAR} \to \text{VAR}$ that is constant on the variable sybols of A.

(WKNEG) Follows from the fact that $\tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(\neg\varphi(A)) = \emptyset$ iff $\tilde{\mathcal{R}} \Vdash \bot \leftsquigarrow \neg\varphi(A)$, and thus $\tilde{\mathcal{R}} \Vdash \bot \leftsquigarrow \neg\varphi(A)$ for every substitution φ iff $\mathcal{E} \Vdash A$ for every $\mathcal{E} \in \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}$.

(MP) Follows from the fact that if $B \leftarrow A$ is a classical tautology, then $\varphi(B)$ is a classical consequence of $\varphi(A)$ for any substitution φ.

(EQ) Follows from the fact that $B \leftarrow A$ and $\neg A \vee B$ are classically equivalent.

(EQF) Follows from the fact that $\neg A \vee B$ and $B \leftarrow A$ are classically equivalent.

□

Completeness, on the other hand, requires a bit of legwork. In the next few sections we will introduce some new technical lemmas and concepts, and then use them to prove the completeness theorem.

E.1 Technical Lemmas

The completeness theorem is concerned with sets $\mathcal{S} \subseteq \mathcal{L}$ of Defeasible Datalog formulas that are complete for the enriched rationality postulates $\tilde{\mathbb{R}}_{\leftsquigarrow}$, which we'll refer to as *enriched sets*. The first lemma we'll need shows that if something holds classically in these sets, then it also holds defeasibly under all possible background conditions. This is a generalisation of the (SUP) and (SUPF) properties:

Lemma 50. *The following two defeasible properties are consequences of* $\tilde{\mathbb{R}}_{\leftsquigarrow}$*:*

1. $B(\vec{x}) \implies B(\vec{x}) \leftsquigarrow A(\vec{x})$
2. $C(\vec{x}) \leftarrow B(\vec{x}) \implies \neg B(\vec{x}) \vee C(\vec{x}) \leftsquigarrow A(\vec{x})$

Proof. For the first, suppose that $\mathcal{S} \subseteq \mathcal{L}$ is a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set, and that Ψ is a substitution system such that $\mathcal{S} \Vdash_\Psi B(\vec{x})$. Then by Lemma 32 and the fact that $A \leftarrow B$ is a classical consequence of B, we have that $\mathcal{S} \Vdash_\Psi B(\vec{x}) \leftarrow A(\vec{x})$. But then by (Sup)-completeness we conclude that $\mathcal{S} \Vdash_\Psi B(\vec{x}) \rightsquigarrow A(\vec{x})$ as required.

For the second, suppose that $\mathcal{S} \subseteq \mathcal{L}$ is a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set, and that Ψ is a substitution system such that $\mathcal{S} \Vdash_\Psi C(\vec{x}) \leftarrow B(\vec{x})$. Then by Lemma 32 and the fact that $\neg B \vee C \leftarrow A$ is a classical consequence of $C \leftarrow B$, we have that $\mathcal{S} \Vdash_\Psi \neg B(\vec{x}) \vee C(\vec{x}) \leftarrow A(\vec{x})$. But then by (Sup)-completeness we again conclude that $\mathcal{S} \Vdash_\Psi \neg B(\vec{x}) \vee C(\vec{x}) \rightsquigarrow A(\vec{x})$ as required. □

The second lemma we'll need has to do with a relation we can define on compounds $A \in \mathcal{L}_c$ with respect to an enriched set, which roughly captures what it means for a compound to be *"at least as typical"* as another compound. This relation is based on the "$\alpha R \beta$" relation introduced by Lehmann et al. [25, p. 46] in the proof of their completeness result for propositional logic:

Definition 39. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then for any compounds $A, B \in \mathcal{L}_c$, we say A is* at least as typical as *B iff $\neg A \not\rightsquigarrow A \vee B \in \mathcal{S}$. We denote this by $A \preceq_\mathcal{S} B$, and write $A \equiv_\mathcal{S} B$ to mean $A \preceq_\mathcal{S} B$ and $B \preceq_\mathcal{S} A$.*

We can view $A \preceq_\mathcal{S} B$ as defining a binary relation on the set of compounds $\mathcal{L}_c$. In general, $\preceq_\mathcal{S}$ is not a partial order, as it fails to be reflexive! This follows from completeness for (Lle) and (Rw), as if $\mathcal{S}$ contains $\bot \rightsquigarrow A$ then it also contains $\neg A \rightsquigarrow A$ and hence $A \not\preceq_\mathcal{S} A$. We introduce the following name for compounds that violate reflexity:

Definition 40. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then a compound $A \in \mathcal{L}_c$ is* consistent *with respect to $\mathcal{S}$ if $\bot \rightsquigarrow A \notin \mathcal{S}$, and* inconsistent *with respect to $\mathcal{S}$ otherwise.*

Consistency turns out to be the only obstacle to $\preceq_\mathcal{S}$ being a partial order. In fact, over the set of consistent compounds, $\preceq_\mathcal{S}$ is a *total* order:

Lemma 51. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then $\preceq_\mathcal{S}$ is reflexive, transitive and total over the set of consistent compounds.*

Proof. Follows directly from Lemma 26. □

The next lemma we'll need about enriched sets is that the $\preceq_\mathcal{S}$ relation interacts with defeasible consequence in the following way:

Lemma 52. *The following defeasible property is a consequence of $\tilde{\mathbb{R}}_{\rightsquigarrow}$:*

1. $A(\vec{x}) \preceq_\mathcal{S} B(\vec{x}),\ C(\vec{x}) \rightsquigarrow B(\vec{x}) \implies \neg B(\pi\vec{x}) \vee C(\pi\vec{x}) \rightsquigarrow A(\vec{x})$

Proof. Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set, and suppose that Ψ is a substitution system such that $\mathcal{S} \Vdash_\Psi A(\vec{x}) \preceq_\mathcal{S} B(\vec{x})$ and $\mathcal{S} \Vdash_\Psi C(\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} B(\vec{x})$. We consider two cases.

First, suppose that $\mathcal{S} \Vdash_\Psi \bot \mathrel{\sim\!\!\!\!\!\leftarrow} B(\vec{x})$. Then by Theorem 7 and the fact that $\beta \mathrel{|\!\sim} \bot \implies \alpha \mathrel{|\!\sim} \neg\beta$ is a consequence of the propositional rationality postulates, this implies that $\mathcal{S} \Vdash_\Psi \neg B(\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\vec{x})$. But then an application of (Rw)-completeness gives $\mathcal{S} \Vdash_\Psi \neg B(\pi\vec{x}) \vee C(\pi\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\vec{x})$ as required.

Next, suppose that $\mathcal{S} \Vdash_\Psi \bot \mathrel{\not\sim\!\!\!\!\!\leftarrow} B(\vec{x})$. By a symmetric application of (Pref)-completeness, this implies that $\mathcal{S} \Vdash_\Psi B(\vec{x}) \equiv_\mathcal{S} B(\sigma\vec{x})$, and hence by Lemma 51 that $\mathcal{S} \Vdash_\Psi A(\vec{x}) \preceq_\mathcal{S} B(\sigma\vec{x})$. By (Per)-completeness, we have that $\mathcal{S} \Vdash_\Psi C(\sigma\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} B(\sigma\vec{x})$, and since $\alpha \preceq_\mathcal{S} \beta,\ \beta \mathrel{|\!\sim} \gamma \implies \alpha \mathrel{|\!\sim} \beta \to \gamma$ is a consequence of the propositional rationality postulates we conclude from Theorem 7 that $\mathcal{S} \Vdash_\Psi \neg B(\sigma\vec{x}) \vee C(\sigma\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\vec{x})$. But since the permutation σ in Ψ was arbitrary in our assumptions, an application of (WkIrr)-completeness gives $\mathcal{S} \Vdash_\Psi \neg B(\pi\vec{x}) \vee C(\pi\vec{x}) \mathrel{\sim\!\!\!\!\!\leftarrow} A(\vec{x})$ as required. □

Note that in the interest of clarity we have abused notation slightly, and used "$A(\vec{x}) \preceq_\mathcal{S} B(\vec{x})$" to denote the inclusion schema "$\neg A(\vec{x}) \mathrel{\not\sim\!\!\!\!\!\leftarrow} A(\vec{x}) \vee B(\vec{x})$". Finally, we show that the defeasible part of an enriched set is closed under a certain kind of classical reasoning:

Lemma 53. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set, and $A \in \mathcal{L}_c$ some compound. Now consider the following set of classical formulas:*

$$\Gamma = (\mathcal{S} \setminus \mathcal{L}_{\rightsquigarrow}) \cup \left\{\pi(C) \leftarrow \pi(B) : A \equiv_\mathcal{S} B,\ C \mathrel{\sim\!\!\!\!\!\leftarrow} B \in \mathcal{S} \text{ and } \pi : \textsc{var} \to \textsc{var}\right\}$$

Then $D \mathrel{\sim\!\!\!\!\!\leftarrow} A \in \mathcal{S}$ iff $D \leftarrow A$ is a classical consequence of Γ.

Proof. The "only if" direction follows directly from the definitions, so suppose that $D \leftarrow A$ is a classical consequence of Γ. Then by Corollary 2 there is some finite subset $\Gamma' \subseteq \Gamma$ such that $D \leftarrow A$ is a classical consequence of Γ'. By Lemma 32, we can furthermore replace facts $A \in \Gamma'$ by their rule equivalents $\bot \leftarrow \neg A$, and assume without loss of generality that Γ' contains only rules.

Assuming that $\Gamma' = \{C_i \leftarrow B_i : 1 \leq i \leq n\}$, by Lemma 24 and the deduction theorem for first-order logic, we have that the following formula is a classical tautology:

$$D \leftarrow A \wedge (\neg B_1 \vee C_1) \wedge \cdots \wedge (\neg B_n \vee C_n)$$

By Lemmas 50 and 52, we have that $\neg B_i \vee C_i \mathrel{\sim\!\!\!\!\!\leftarrow} A \in \mathcal{S}$ for every $1 \leq i \leq n$, and thus by completeness for (Refl) and (And) that $\Lambda \mathrel{\sim\!\!\!\!\!\leftarrow} A \in \mathcal{S}$, where Λ denotes the right-hand side of the above formula. But then (Rw)-completeness implies that $D \mathrel{\sim\!\!\!\!\!\leftarrow} A \in \mathcal{S}$ as required. □

E.2 Normal Enriched Herbrand Interpretations

Let's review what it means for an enriched ranked interpretation $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^\infty$ to satisfy a defeasible rule $B \mathrel{\sim\!\!\leftarrow} A$. According to the discussion after Definition 32, this is true whenever the minimal-rank enriched Herbrand interpretations $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ all satisfy $\mathcal{E} \mid\!\sim B \leftarrow A$. Note, however, that these minimal interpretations also satisfy all of the *other* defeasible consequences of A with respect to $\tilde{\mathcal{R}}$. Consider any $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$, for instance. Then $\mathcal{E} \mid\!\sim C \leftarrow A$ whenever $\tilde{\mathcal{R}} \Vdash C \mathrel{\sim\!\!\leftarrow} A$.

More generally, these minimal interpretations are also constrained by the defeasible properties of compounds *less typical* than A:

Lemma 54. *Let $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega^\infty$ be an enriched ranked interpretation, and consider the set $\mathcal{S}_{\tilde{\mathcal{R}}} = \{\alpha \in \mathcal{L} : \tilde{\mathcal{R}} \Vdash \alpha\}$. Then if $A \preceq_{\mathcal{S}} B$ and $\tilde{\mathcal{R}} \Vdash C \mathrel{\sim\!\!\leftarrow} B$, $\mathcal{E} \mid\!\sim \pi(C) \leftarrow \pi(B)$ for every $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ and substitution $\pi : \textsc{var} \to \textsc{var}$.*

Proof. Follows directly from Theorem 8 and Lemma 52. □

Our strategy for proving the completeness theorem will be to characterise the minimal interpretations $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ *purely* in terms of properties of the enriched set $\mathcal{S}_{\tilde{\mathcal{R}}}$. We will then be able to "reverse-engineer" this procedure, and figure out what these minimal $\mathcal{E}$ should look like for an arbitrary enriched set $\mathcal{S}$ and compound A. We refer to these "reverse-engineered" enriched Herbrand interpretations as *normal*:

Definition 41. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then an enriched Herbrand interpretation $\mathcal{E} \in \tilde{\mathbb{H}}$ is* $\mathcal{S}$-normal *for a compound $A \in \mathcal{L}_c$ if the following three conditions hold:*

1. *$\mathcal{E} \Vdash \alpha$ for every $\alpha \in \mathcal{S} \setminus \mathcal{L}_{\rightsquigarrow}$.*
2. *$\mathcal{E} \Vdash \varphi(A)$ for some substitution $\varphi : \textsc{var} \to T_{\mathcal{E}}$.*
3. *$\mathcal{E} \mid\!\sim C \leftarrow B$ whenever $A \equiv_{\mathcal{S}} B$ and $C \mathrel{\sim\!\!\leftarrow} B \in \mathcal{S}$.*

We denote the set of $\mathcal{S}$-normal interpretations for a compound $A \in \mathcal{L}_c$ by $\text{norm}_{\mathcal{S}}(A)$. Just as minimal interpretations characterise the defeasible rules satisfied by an enriched ranked interpretation, normal interpretations characterise the defeasible rules in an enriched set:

Lemma 55. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then $B \mathrel{\sim\!\!\leftarrow} A \in \mathcal{S}$ iff $\mathcal{E} \mid\!\sim B \leftarrow A$ for every $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$.*

Proof. The "only if" direction follows directly from the definitions, so suppose that $\mathcal{E} \mid\!\sim B \leftarrow A$ for every $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$. By Lemma 53, $B \mathrel{\sim\!\!\leftarrow} A \in \mathcal{S}$ iff $B \leftarrow A$ is a classical consequence of Γ, where Γ is the set of formulas defined

in the proof of Lemma 53. Assume for the sake of contradiction, then, that $B \sim\!\!\!\!\leftarrow A \notin \mathcal{S}$.

This implies that there is some Herbrand interpretation $\mathcal{H} \subseteq \tilde{\mathbb{B}}$, such that $\mathcal{H} \Vdash \Gamma$ and $\mathcal{H} \nVdash B \leftarrow A$, which in turn implies that there is some ground substitution φ such that $\mathcal{H} \Vdash \varphi(A) \wedge \neg\varphi(B)$. Letting $\textsc{var}_A$ and $\textsc{var}_B$ denote the variable symbols in A, B respectively, we define $T = \varphi(\textsc{var}_A \cup \textsc{var}_B)$. Then consider the enriched Herbrand interpretation $\mathcal{E} = \langle \mathcal{H}, T \rangle$. We claim that $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$:

1. $\mathcal{E} \Vdash \mathcal{S} \setminus \mathcal{L}_{\sim\!\!\!\!\rightarrow}$ because $\mathcal{S} \setminus \mathcal{L}_{\sim\!\!\!\!\rightarrow} \subseteq \Gamma$ and $\mathcal{H} \Vdash \Gamma$.
2. $\mathcal{E} \Vdash \varphi(A)$ by construction.
3. $\mathcal{E} \Vdash\!\!\sim C \leftarrow B$ whenever $A \equiv_{\mathcal{S}} B$ and $C \sim\!\!\!\!\leftarrow B \in \mathcal{S}$ because $\mathcal{H} \Vdash \Gamma$.

But by construction, $\mathcal{E} \nVdash\!\!\sim B \leftarrow A$, contradicting our assumptions, and thus we conclude that $B \sim\!\!\!\!\leftarrow A \in \mathcal{S}$ as required. □

Similarly, just as the interpretations of finite rank characterise the classical formulas satisfied by an enriched ranked interpretation, normal interpretations characterise the classical formulas in an enriched set:

Lemma 56. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\sim\!\!\!\!\rightarrow}$-complete set. Then for any classical formula $\alpha \in \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$, $\alpha \in \mathcal{S}$ iff $\mathcal{E} \Vdash \alpha$ for every $A \in \mathcal{L}_c$ and $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$.*

Proof. The "only if" direction follows directly from the definitions, so suppose that for every $A \in \mathcal{L}_c$ and $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$ we have that $\mathcal{E} \Vdash \alpha$. By Lemma 32 we can assume without loss of generality that α is some rule $C \leftarrow B \in \mathcal{L}_{\rightarrow}$. But this implies that $\mathcal{E} \Vdash\!\!\sim \neg\varphi(B) \vee \varphi(C) \leftarrow A$ for every substitution φ, $A \in \mathcal{L}_c$ and $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$, which by Lemma 55 implies that $\neg\varphi(B) \vee \varphi(C) \sim\!\!\!\!\leftarrow A \in \mathcal{S}$ for every substitution φ and $A \in \mathcal{L}_c$. In particular, $\neg\varphi(B) \vee \varphi(C) \sim\!\!\!\!\leftarrow \neg(\neg\varphi(B) \vee \varphi(C)) \in \mathcal{S}$ and thus by completeness for (Refl), (And) and (Rw) we have that $\bot \sim\!\!\!\!\leftarrow \neg(\neg\varphi(B) \vee \varphi(C)) \in \mathcal{S}$ for every substitution φ. Finally, an application of (WkNeg)-completeness gives that $\neg B \vee C \in \mathcal{S}$ and hence by Lemma 32 that $C \leftarrow B \in \mathcal{S}$ as required. □

Finally, one justification for the "at least as typical as" relation $\preceq_{\mathcal{S}}$ is that $A \preceq_{\mathcal{S}} B$ is satisfied by an enriched ranked interpretation precisely when the rank of the minimal A-interpretations is less than or equal to the rank of the minimal B interpretations (where we take the rank of the empty set to be ∞ for consistency). And indeed, this property is shared by normal interpretations:

Lemma 57. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\sim\!\!\!\!\rightarrow}$-complete set, and suppose that $A \preceq_{\mathcal{S}} B$. Then for any $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$, if there exists some substitition $\varphi : \textsc{var} \to T_{\mathcal{E}}$ such that $\mathcal{E} \Vdash \varphi(B)$, this implies that $\mathcal{E} \in \text{norm}_{\mathcal{S}}(B)$ and $A \equiv_{\mathcal{S}} B$.*

Proof. Suppose $\mathcal{E} \Vdash \varphi(B)$ for some $\mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)$ and $\varphi : \textsc{var} \to T_{\mathcal{E}}$. We first show that $\mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(B)$:

1. $\mathcal{E} \Vdash \alpha$ for every $\alpha \in \mathcal{S} \setminus \mathcal{L}_{\rightsquigarrow}$ since $\mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)$.
2. $\mathcal{E} \Vdash \varphi(B)$ by assumption.
3. Suppose that $B \equiv_{\mathcal{S}} C$ and $D \mathrel{\reflectbox{$\rightsquigarrow$}} C \in \mathcal{S}$. Then by Lemma 51 we have $A \equiv_{\mathcal{S}} C$, and hence $\mathcal{E} \mathrel{|\!\sim} D \leftarrow C$ as $\mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)$.

The last step is to show that $A \equiv_{\mathcal{S}} B$. Assume for the sake of contradiction that $B \not\preceq_{\mathcal{S}} A$, i.e. that $\neg B \mathrel{\reflectbox{$\rightsquigarrow$}} A \vee B \in \mathcal{S}$. Then by assumption, $A \preceq_{\mathcal{S}} B$, which implies that $A \equiv_{\mathcal{S}} A \vee B$ and hence $\mathcal{E} \mathrel{|\!\sim} \neg B \leftarrow A \vee B$ as $\mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)$. But this is a contradiction, since $\mathcal{E} \Vdash \varphi(B)$ by assumption. □

E.3 Proof of Completeness

Our goal in this section is to prove the completeness theorem, which states than an enriched set can be precisely characterised by an enriched ranked interpretation. We will proceed by ranking the set's normal interpretations in such a way as to achieve this:

Definition 42. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then we define $\Omega_{\mathcal{S}} = \{\langle A, \mathcal{E}\rangle : A \in \mathcal{L}_c,\ \mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)\}$. Furthermore, we define the relation $\leq_{\mathcal{S}}$ on $\Omega_{\mathcal{S}}$ by the following criterion:*

$$\langle A, \mathcal{E}\rangle \leq_{\mathcal{S}} \langle B, \mathcal{E}'\rangle \text{ iff } A \preceq_{\mathcal{S}} B$$

An immediate consequence of the definition is that $\leq_{\mathcal{S}}$ is a total order:

Lemma 58. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set. Then $\leq_{\mathcal{S}}$ is reflexive, transitive and total.*

Proof. By Lemma 55, a compound $A \in \mathcal{L}_c$ is consistent iff $\mathrm{norm}_{\mathcal{S}}(A) \neq \emptyset$. But then Lemma 51 implies directly that $\leq_{\mathcal{S}}$ is a total order. □

This is in fact everything we need to finish our proof. The total order $\Omega_{\mathcal{S}}$ will play the role of the domain in the enriched ranked interpretation we construct, and the various lemmas and definitions we have introduced in the last two sections suffice to prove that it characterises $\mathcal{S}$ precisely:

Theorem 9. *Let $\mathcal{S} \subseteq \mathcal{L}$ be a $\tilde{\mathbb{R}}_{\rightsquigarrow}$-complete set of formulas. Then there exists some enriched ranked interpretation $\tilde{\mathcal{R}}$ such that $\mathcal{S} = \mathcal{S}_{\tilde{\mathcal{R}}}$.*

Proof. Consider the following enriched ranked interpretation $\tilde{\mathcal{R}} : \tilde{\mathbb{H}} \to \Omega_{\mathcal{S}}^{\infty}$, where by convention we take $\min \emptyset = \infty$:

$$\tilde{\mathcal{R}}(\mathcal{E}) = \min_{\leq_{\mathcal{S}}} \big\{\langle A, \mathcal{E}\rangle : A \in \mathcal{L}_c \text{ and } \mathcal{E} \in \mathrm{norm}_{\mathcal{S}}(A)\big\}$$

We claim that $\mathcal{S} = \mathcal{S}_{\tilde{\mathcal{R}}}$, as required. First, consider some classical formula $\alpha \in \mathcal{L}_f \cup \mathcal{L}_{\rightarrow}$. Then by Lemma 56, $\alpha \in \mathcal{S}$ iff $\mathcal{E} \Vdash \alpha$ for every $A \in \mathcal{L}_c$ and $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$. But by definition of $\Omega_{\mathcal{S}}$, $\tilde{\mathcal{R}}(\mathcal{E})$ is finite iff $\mathcal{E}$ is $\mathcal{S}$-normal for some $A \in \mathcal{L}_c$, and hence this is true iff $\tilde{\mathcal{R}} \Vdash \alpha$.

Next, consider some defeasible rule $B \mathrel{\sim\!\!\!\!\leftarrow} A \in \mathcal{L}_{\sim\!\!\rightarrow}$. By Lemma 55, $B \mathrel{\sim\!\!\!\!\leftarrow} A \in \mathcal{S}$ iff $\mathcal{E} \mathrel{|\!\sim} B \leftarrow A$ for every $\mathcal{E} \in \text{norm}_{\mathcal{S}}(A)$. Similarly, $\mathcal{E} \Vdash B \mathrel{\sim\!\!\!\!\leftarrow} A$ iff $\mathcal{E} \mathrel{|\!\sim} B \leftarrow A$ for every $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$. The claim therefore follows immediately if we can show that $\text{norm}_{\mathcal{S}}(A) = \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$ for every $A \in \mathcal{L}_c$.

Suppose then that $\mathcal{E}, \mathcal{E}' \in \text{norm}_{\mathcal{S}}(A)$. By Lemma 58, $\leq_{\mathcal{S}}$ is total, and thus $\tilde{\mathcal{R}}(\mathcal{E}) \leq \tilde{\mathcal{R}}(\mathcal{E}')$ or $\tilde{\mathcal{R}}(\mathcal{E}') \leq \tilde{\mathcal{R}}(\mathcal{E})$. Without loss of generality, we assume the former. But then either $\tilde{\mathcal{R}}(\mathcal{E}) = \tilde{\mathcal{R}}(\mathcal{E}')$, or by definition there must be some $B \in \mathcal{L}_c$ such that $B \preceq_{\mathcal{S}} A$ and $\mathcal{E}' \in \text{norm}_{\mathcal{S}}(B)$. But then by Lemma 57, $A \equiv_{\mathcal{S}} B$, and again we conclude that $\tilde{\mathcal{R}}(\mathcal{E}) = \tilde{\mathcal{R}}(\mathcal{E}')$. Since $\mathcal{E}, \mathcal{E}'$ were arbitrary, we conclude that everything in $\text{norm}_{\mathcal{S}}(A)$ has the same $\tilde{\mathcal{R}}$-rank, and thus $\text{norm}_{\mathcal{S}}(A) = \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$. □

Appendix F

Proofs for Section 6.7

In this appendix we collect some basic properties of *rank entailment* and *minimal-model entailment*, which are introduced in Definitions 33 and 34 respectively. To begin with, we consider rank entailment, and show that it is a consequence operator in the sense of Tarski [33]. In particular, this implies that rank entailment is *monotonic* on the meta-level, despite dealing with the consequences of defeasible formulas on the object-level:

Lemma 37. *Rank entailment satisfies the following three properties:*

$$\begin{aligned}
&(\textsc{Incl}) && \alpha \in \mathcal{K} \text{ implies } \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \\
&(\textsc{Cumu}) && \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \text{ and } \mathcal{K} \cup \{\alpha\} \mathrel{|\!\approx_{\mathcal{R}}} \beta \text{ implies } \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \beta \\
&(\textsc{Mono}) && \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha \text{ implies } \mathcal{K} \cup \{\beta\} \mathrel{|\!\approx_{\mathcal{R}}} \alpha
\end{aligned}$$

Proof. We split the proof into cases:

(Incl) Suppose $\alpha \in \mathcal{K}$. Then every $\tilde{\mathcal{R}}$ satisfying $\mathcal{K}$ must clearly satisfy α, and thus by definition $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha$.

(Cumu) Suppose $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha$ and $\mathcal{K} \cup \{\alpha\} \mathrel{|\!\approx_{\mathcal{R}}} \beta$. Then by definition, if $\tilde{\mathcal{R}} \Vdash \mathcal{K}$ then $\tilde{\mathcal{R}} \Vdash \alpha$. In particular, $\tilde{\mathcal{R}} \Vdash \mathcal{K}$ iff $\tilde{\mathcal{R}} \Vdash \mathcal{K} \cup \{\alpha\}$, and thus it immediately follows that $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \beta$.

(Mono) Note that $\tilde{\mathcal{R}}$ is a model of $\mathcal{K} \cup \{\beta\}$ only if $\tilde{\mathcal{R}}$ is a model of $\mathcal{K}$, and thus if $\mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha$ then $\mathcal{K} \cup \{\beta\} \mathrel{|\!\approx_{\mathcal{R}}} \alpha$ as well.

□

Rank entailment also respects many rationality postulates on a meta-level:

Lemma 38. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base, and consider the set $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mathrel{|\!\approx_{\mathcal{R}}} \alpha\}$. Then $\mathcal{S}$ is complete for every enriched postulate $\tilde{\mathbb{R}}_{\rightsquigarrow}$ except* (Rm).

Proof. This follows from the observation that all of the properties in $\mathbb{R}_{\sim}$ except (Rm) are *positive Horn clauses* and hence closed under intersection. In other words, if $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{L}$ are both complete for these properties, then $\mathcal{S} \cap \mathcal{S}'$ is also complete for these properties. □

Next, we consider properties of minimal-model entailment. First of all, it might be surprising that a unique minimal model even *exists* for Defeasible Datalog knowledge bases:

Lemma 39. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a consistent knowledge base, and define the* minimal model *of $\mathcal{K}$ to be the following enriched ranked interpretation $\tilde{\mathcal{R}}_{\mathcal{K}} : \tilde{\mathbb{H}} \to \mathbb{N}^{\infty}$:*

$$\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) = \min\left\{\tilde{\mathcal{R}}(\mathcal{E}) : \tilde{\mathcal{R}} \Vdash \mathcal{K}\right\}$$

Then this minimal model always satisfies $\mathcal{K}$.

Proof. Clearly if $\alpha \in \mathcal{K}$ is a classical formula, then $\tilde{\mathcal{R}}_{\mathcal{K}} \Vdash \alpha$. So suppose that $\mathcal{K}$ contains a defeasible rule $B \leftsquigarrow A$, and consider any $\mathcal{E} \in \min_{\tilde{\mathcal{R}}_{\mathcal{K}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_{\mathcal{K}}}(A)$. Then by definition there must exist some $\tilde{\mathcal{R}}$ satisfying $\mathcal{K}$ such that $\mathcal{E} \in \min_{\tilde{\mathcal{R}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}}(A)$, and thus $\mathcal{E} \Vdash B \leftarrow A$. But $\mathcal{E}$ is arbitrary, so we conclude that $\tilde{\mathcal{R}}_{\mathcal{K}} \Vdash B \leftsquigarrow A$ as required. □

Minimal-model entailment improves upon rank entailment as it respects *all* of the rationality postulates on the meta-level:

Lemma 40. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base, and consider the set $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mathrel{|\!\approx}_M \alpha\}$. Then $\mathcal{S}$ is complete for $\tilde{\mathbb{R}}_{\sim}$.*

Proof. This follows directly from Theorem 8, which states that enriched ranked interpretations satisfy rational sets of formulas, and the fact that $\mathcal{K} \mathrel{|\!\approx}_M \alpha$ iff $\tilde{\mathcal{R}}_{\mathcal{K}} \Vdash \alpha$. □

Appendix G

Proofs for Section 6.8

In this appendix we prove that the *refinement* operator of Section 6.8 preserves the enriched rationality postulates, and is well-defined in the sense that it monotonically increases the number of defeasible rules satisfied by an enriched ranked interpretation:

Lemma 41. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base, and consider the set $\mathcal{S} = \{\alpha \in \mathcal{L} : \mathcal{K} \mid\!\approx_{RM} \alpha\}$. Then $\mathcal{S}$ is complete for $\tilde{\mathbb{R}}_{\rightsquigarrow}$.*

Proof. This follows directly from Theorem 9, the fact that $\mathcal{K} \mid\!\approx_{RM} \alpha$ iff $\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}}) \Vdash \alpha$, and the fact that $\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}})$ can be relabelled into a true enriched ranked interpretation by logical finiteness. □

Lemma 42. *Let $\mathcal{K} \subseteq \mathcal{L}$ be a knowledge base. Then for any formula $\alpha \in \mathcal{L}$, $\mathcal{K} \mid\!\approx_{M} \alpha$ implies that $\mathcal{K} \mid\!\approx_{RM} \alpha$.*

Proof. By definition, refinement only changes the relative rankings within each rank-layer in $\tilde{\mathcal{R}}$. To be more precise, if $\tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}) < \tilde{\mathcal{R}}_{\mathcal{K}}(\mathcal{E}')$, then $\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}})(\mathcal{E}) < \mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}})(\mathcal{E}')$. This implies that $\min_{\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}})} \tilde{\mathbb{H}}^{\mathrm{ref}(\tilde{\mathcal{R}}_{\mathcal{K}})}(A) \subseteq \min_{\tilde{\mathcal{R}}_{\mathcal{K}}} \tilde{\mathbb{H}}^{\tilde{\mathcal{R}}_{\mathcal{K}}}(A)$ for every $A \in \mathcal{L}_c$, and the claim follows. □

Bibliography

[1] Franz Baader et al. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[2] R. Booth and J.B. Paris. "A Note on the Rational Closure of Knowledge Bases with Both Positive and Negative Knowledge". In: *Journal of Logic, Language and Information* 7.2 (1998), pp. 165–190.

[3] Richard Booth et al. "On rational entailment for Propositional Typicality Logic". In: *Artificial Intelligence* 277 (Sept. 2019).

[4] Ronen I Brafman. "A first-order conditional logic with qualitative statistical semantics". In: *Journal of Logic and Computation* 7.6 (1997), pp. 777–803.

[5] Katarina Britz et al. "Theoretical Foundations of Defeasible Description Logics". In: *arXiv preprint arXiv:1904.07559* (Apr. 2019).

[6] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[7] Andrea Calì et al. "Datalog+/-: A family of logical knowledge representation and query languages for new applications". In: *2010 25th Annual IEEE Symposium on Logic in Computer Science*. IEEE. 2010, pp. 228–242.

[8] Giovanni Casini, Thomas Meyer, and Guy Paterson-Jones. "KLM-Style Defeasibility for Restricted First-Order Logic". In: *Proceedings of 19th International Workshop on Non-Monotonic Reasoning* (2021).

[9] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. "Taking defeasible entailment beyond rational closure". In: *European Conference on Logics in Artificial Intelligence*. Springer. 2019, pp. 182–197.

[10] Giovanni Casini et al. "Nonmonotonic reasoning in Description Logics: Rational Closure for the ABox". In: vol. 1014. Jan. 2013, pp. 600–615.

[11] Giovanni Casini et al. "Relevant Closure: A New Form of Defeasible Reasoning for Description Logics". In: *JELIA 2014*. 2014, pp. 92–106.

[12] Stefano Ceri, Georg Gottlob, Letizia Tanca, et al. "What you always wanted to know about Datalog(and never dared to ask)". In: *IEEE transactions on knowledge and data engineering* 1.1 (1989), pp. 146–166.

[13] Brian A Davey and Hilary A Priestley. *Introduction to lattices and order*. Cambridge university press, 2002.

[14] James P. Delgrande. "On first-order conditional logics". In: *Artificial Intelligence* 105.1 (1998), pp. 105–137.

[15] Francesco M Donini and Fabio Massacci. "EXPTIME tableaux for ALC". In: *Artificial Intelligence* 124.1 (2000), pp. 87–138.

[16] N. Friedman and J.Y. Halpern. "Plausibility Measures and Default Reasoning". In: *Journal of the ACM* 48.4 (2001), pp. 648–685.

[17] Nir Friedman, Joseph Y Halpern, and Daphne Koller. "First-order conditional logic revisited". In: *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*. 1996, pp. 1305–1312.

[18] Michael Genesereth and Eric Kao. "The Herbrand Manifesto". In: *International Symposium on Rules and Rule Markup Languages for the Semantic Web*. Springer. 2015, pp. 3–12.

[19] L. Giordano et al. "Semantic characterization of rational closure: From propositional logic to description logics". In: *Art. Int.* 226 (2015), pp. 1–33.

[20] James R Groff, Paul N Weinberg, and Andrew J Oppel. *SQL: the complete reference*. Vol. 2. McGraw-Hill/Osborne, 2002.

[21] Michael Harrison and Thomas Meyer. "DDLV: A System for rational preferential reasoning for datalog". In: *South African Computer Journal* 32 (Dec. 2020). DOI: 10.18489/sacj.v32i2.850.

[22] S. Kraus, D. Lehmann, and M. Magidor. "Nonmonotonic reasoning, preferential models and cumulative logics". In: *Artificial Intelligence* 44 (1990), pp. 167–207.

[23] D. Lehmann. "Another perspective on default reasoning". In: *Annals of Math. and Art. Int.* 15.1 (1995), pp. 61–82.

[24] D. Lehmann and M. Magidor. "Preferential Logics: the Predicate Calculus Case". In: *Proceedings of TARK*. 1990, pp. 57–72.

[25] D. Lehmann and M. Magidor. "What does a conditional knowledge base entail?" In: *Art. Int.* 55 (1992), pp. 1–60.

[26] D. Makinson. *Bridges from Classical to Nonmonotonic Logic*. Vol. 5. Texts in Computing. King's College Publications, 2005.

[27] James Donald Monk. *Mathematical logic*. Vol. 37. Springer Science & Business Media, 2012.

[28] Matthew Morris, Tala Ross, and Thomas Meyer. "Algorithmic definitions for KLM-style defeasible disjunctive Datalog". In: *South African Computer Journal* 32 (Dec. 2020). DOI: 10.18489/sacj.v32i2.846.

[29] Guy Paterson-Jones, Giovanni Casini, and Thomas Meyer. "A Boolean Extension of KLM-Style Conditional Reasoning". In: *Southern African Conference for Artificial Intelligence Research*. Springer. 2021, pp. 236–252.

[30] Maximilian Pensel and Anni-Yasmin Turhan. "Reasoning in the defeasible description logic, computing standard inferences under rational and relevant semantics". In: *International Journal of Approximate Reasoning* 103 (2018), pp. 28–70.

[31] John L Pollock. "Defeasible reasoning". In: *Cognitive science* 11.4 (1987), pp. 481–518.

[32] Robert Stalnaker. "What is a nonmonotonic consequence relation?" In: *Fundamenta Informaticae* 21.1, 2 (1994), pp. 7–21.

[33] A. Tarski. *Introduction to logic*. Oxford University Press, 1941.

[34] A. Tarski. *On some fundamental concepts of metamathematics. [1930] Logic, Semantics, Metamathematics. Papers from 1923 to 1938, translated by J.H. Woodger. Pages 30–36*. Clarendon Press, 1956.

[35] Maarten H Van Emden and Robert A Kowalski. "The semantics of predicate logic as a programming language". In: *Journal of the ACM (JACM)* 23.4 (1976), pp. 733–742.

[36] Robert S Wolf. *A tour through mathematical logic*. Vol. 30. American Mathematical Soc., 2005.