Extending Defeasible Reasoning beyond Rational Closure

Julia Cotterrell
cttjul001@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

Defeasible reasoning is a key component of artificial intelligence (AI) systems. We examines the progression from rational to lexicographic closure, alongside tools that support knowledge base generation and user-guided debugging. The Kraus, Lehmann, and Magidor (KLM) framework formalises nonmonotonic logic, providing a structured approach to rational closure, ranking, and resolving conflicts between default rules. While effective, it often fails to capture finer distinctions between defeasible statements.

To address this, we explore alternative inference mechanisms that improve expressivity. Knowledge Base Generators (KBGs) offer structured ranked interpretations, while user interfaces (UI) and debuggers enhance reasoning transparency. Additionally, preferential models extend classical logic by introducing a more flexible hierarchy of reasoning principles.

By integrating these findings, this study contributes to the development of scalable, expressive, and user-friendly reasoning systems for knowledge representation.

KEYWORDS

artificial intelligence, knowledge representation and reasoning, nonmonotonic reasoning, defeasible reasoning, rational closure, lexicographic closure, knowledge base generator, user interface, debugger

1 INTRODUCTION

Artificial intelligence (AI) has significantly evolved and now influences many aspects of modern life. A key component of AI is knowledge representation (KR), which structures information in a format that machines can process, enabling reasoning, decision-making, and problem-solving. KR methods are fundamental to natural language understanding, robotics, and expert systems, which allows AI to infer new facts, update beliefs, and support planning tasks. Among various KR approaches, *propositional logic* remains widely used due to its straightforward syntax and logical foundation.

In propositional logic, knowledge is encoded using basic statements that are either true or false, combined with logical operators such as AND, OR, and NOT [2]. However, one of its limitations is that it follows a *monotonic* reasoning process. This means that once a conclusion is established, it remains unchanged, even when new information suggests otherwise. This is problematic when dealing with exceptions. For instance, while the general statement "birds fly" is often true, exceptions like "penguins do not fly" challenge its rigidity. Addressing this issue, *defeasible reasoning* provides a way to revise conclusions when conflicting or more specific information is introduced [24].

One of the formal foundations for defeasible reasoning is the KLM framework, developed by Kraus, Lehmann, and Magidor [18]. This framework utilises *preferential models* to rank possible valuations, based on typicality, allowing conclusions to be drawn from the most representative cases. Extending this, *rational closure* introduces a ranking system that prioritises more specific defaults over general ones, ensuring that exceptions are appropriately handled [21]. While rational closure provides a structured approach, alternative methods like *lexicographic closure* refine reasoning further by not only considering specificity but also the number of supporting defaults, thus resolving conflicts more effectively [7, 20].

Constructing knowledge bases for such reasoning systems can be complex and time-intensive. To address this, *knowledge base generators* (KBGs) automate the process of defining rules, resolving inconsistencies, and structuring knowledge effectively [23]. Various approaches, such as Hrycej's problem-specific generator [16] and Bisson's KBG framework [3], provide methodologies for systematically creating structured knowledge bases. Additionally, evaluations of these approaches, such as those by Bryant and Krause [4], offer insights into their effectiveness in rule-based reasoning applications.

As defeasible reasoning grows in complexity, effective *user interfaces* (UIs) and debugging tools become essential for managing and interpreting automated reasoning processes. Interactive tools such as the SILK GUI by Grosof et al. [15] enable users to input knowledge, explore reasoning justifications, and visualise inference structures. Debugging frameworks such as that of Coetzer and Britz [10] assist in identifying and correcting inconsistencies within logical ontologies.

This paper explores the progression from rational closure to lexicographic closure, along with the tools that support knowledge base generation and user-guided debugging. By analysing these developments, we aim to contribute to the design of more efficient, scalable, and transparent defeasible reasoning systems.

2 CLASSICAL PROPOSITIONAL LOGIC

Propositional logic serves as the formalisation of knowledge representation and reasoning[17]. In order to do this, we need to first formalise the idea of truth as a concept by assigning one of two arbitrary values to a statement. In this literature review, we will stick to the conventional use of *true*, of *T*, and *false*, of *F*. Then, we would combine the statements using types of connectives to create more complicated statements [17]. This allows propositional logic to assess the truth of statements without the distraction of natural language subtleties. However, its monotonic nature restricts its ability to handle exceptions [22]. This limitation highlights the need for alternative reasoning mechanisms.

2.1 Syntax

In Propositional formulas consists of two main components: atoms and boolean operators.

2.1.1 Atoms. [2] Atoms otherwise called *atomic propositions* are a set of unbounded symbols and are denoted by lowercase Latin alphabet letters. *Atomics Propositions,* $\mathcal{P}$, consist of elements $p, q, r, s, ...$ in such a way that $\mathcal{P} = \{p, q, r, s, ...\}$.

2.1.2 Boolean operators. [2]. Boolean Operators are the connective used to combine atoms to create formulas. These are shown by:

$\wedge$	and
$\vee$	or
$\rightarrow$	if then
$\leftrightarrow$	if, and only if

Figure 1: Set of binary boolean operators

All of the above are considered binary. Negation, $\neg$, is also a boolean operator but is unary meaning that it takes one operand rather than two like the above binary operators [2].

2.1.3 Formulas. Formulas, also referred to as propositional formulas or fml, are created by combining atoms and binary operators to make more complex statements. This is necessary to reach the point of knowledge representation. The grammar used to produce formulas are [2]:

$$
\begin{aligned}
fml &::= p && \text{for some } p \in \mathcal{P} \\
fml &::= \neg fml \\
fml &::= fml\ op\ fml \\
op &::= \vee \mid \wedge \mid \rightarrow \mid \leftrightarrow
\end{aligned}
$$

The set of all formulas that can be derived from the grammar is denoted by $\mathcal{L}$ and elements in $\mathcal{L}$ are denoted by lower case Greek alphabet letters $\{\beta, \alpha, \sigma, ...\}$ [2].

2.2 Semantics

The semantics of any logic or language allow for the understanding of meaning. It is defined by interpretations that assign a value, T or F, to each atom in a formula allowing semantic rules to determine the truth value of a formula [2]. It is necessary now to differentiate between Object- level and Meta level semantics. In Propositional Logic, any part of the language that is used to model knowledge is considered object-level, while anything that operates over the object-level is considered meta-level. Valuations, atoms, and boolean operators would all be considered object level while satisfiability and entailment would be considered meta-level [17].

2.3 Object-Level Semantics

2.3.1 Valuations. Valuations, otherwise referred to as *worlds* or *interpretations*, assign a truth value to atoms and are donated by the lowercase Latin alphabet letters $\{u, v, w\}$. Valuation, u, is defined as a function: $P \rightarrow \{T, F\}$. That means, a valuation assigns either *true*, or *false* to each atom $p \in \mathcal{P}$ in the language. [17]. The set of all valuations is donated by $\mathcal{U}$. For example, given $\mathcal{P} = \{p, q, r\}$, a random valuation of $u \in \mathcal{U}$ could result in $u(p) = F, u(q) = T$ and $u(r) = F$.

In shorthand notation, this is written as $\overline{\text{p}}\text{q}\overline{\text{r}}$, in which a bar above the atom indicates *false* and the absence of the bar indicates *true.*

2.3.2 Atoms. Atoms individually have no intrinsic value, but when woven together with boolean operators to create formulas, they help us understand our knowledge base. When considering example statements such as "penguins exist" and "birds exist", we can assign atoms p and b to represent them, respectively. However, we could have rather assigned "penguins fly" and "birds fly" to the atoms p and b. However, in the first case p is true while in the second it is false. This is to show that, without context, atoms themselves do not have any intrinsic value.

2.3.3 Boolean Operators. Boolean operators, on the other hand, do have intrinsic value as they connect two atoms that already have their own set of truth values. A new truth set of the formula is formed as a result of the boolean operator and the truth values of the two atoms.

Given $\mathcal{P} = \{p, q\}$, the possible valuations $u \in \mathcal{U}$ can be represented in a *truth table*, that is a convenient way of showing semantics of a formula. This is shown below: As we recall, all of the above

p	q	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$	$\neg p$	$\neg q \rightarrow \neg p$
T	T	T	T	T	T	F	T
T	F	T	F	F	F	F	F
F	T	T	F	T	F	T	T
F	F	F	F	T	T	T	T

Figure 2: The truth table for Boolean operators

Boolean operators are binary except for $\neg$, which is unary.

The semantics of boolean operators remain constant when connecting two formulas in the same way they remain constant when connecting two atoms. That is, given the formula $A := (p \rightarrow q) \leftrightarrow (\neg q \rightarrow\ p)$ and the valuation $u := \overline{\text{p}}\text{q}$. The truth table above shows that $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are both T. Resulting in A being T under u.

2.4 Meta-Level Semantics

2.4.1 Satisfiability. An atom is considered satisfiable if, for some interpretation of atom, p, the valuation, u, is T. This is donated as $u \Vdash p$. If it is considered unsatisfiable, it is written as $u \nVdash p$ [2].

This definition can be extended to include the satisfaction of formulas. This occurs when a valuation $u \in \mathcal{U}$ satisfies a formula $A \in \mathcal{L}$ if and only if A has a truth value of T [2].

This can be extended further by considering *knowledge bases* $\mathcal{K}$, the finite set of formulas that restrict the finite $u \in \mathcal{U}$ that satisfy the knowledge base. For instance, if we let b equal "birds" and f equal "fly" and we have the knowledge base $\mathcal{K} = \{b, b \rightarrow f\}$. This means that there are two known facts in the knowledge base, that "birds exist" and "birds fly". Every u that satisfies this knowledge base is considered a *model*, otherwise known as a fact about the world. In this example, the only model would be $\{bf\}$. It is possible for every $u \in \mathcal{U}$ that $u \nVdash \mathcal{K}$, in other words, there are no models of $\mathcal{K}$ [17].

2.4.2 Entailment. Entailment is the logical consequence of a set of statements. That is to say that entailment defines which formulas are a result of a set of predefined formulas. Formally, defined as α is entailed by $\mathcal{K}$ if and only if for every $u \in$ such that $u \models \mathcal{K}$, it is true that $\alpha \models \mathcal{K}$ [17]. This is essential in understanding how to systematically reason with logic.

3 DEFEASIBLE REASONING

Classical propositional logic is monotonic, which means that it is limited in its ability to handle exceptional knowledge. This is evident when dealing with an example:

$$\mathcal{K} = \{p \to b, b \to f, p \to \neg f\}$$

Where p = "penguin", b = "bird" and f = "fly". Using this knowledge base, classical propositional logic rules would conclude that penguins do not exist. This is done by following the logic $p \to b$ and $b \to f$, therefore $p \to f$ but in $\mathcal{K}$ it states $p \to \neg f$. Thus we can conclude through proof of contradiction that penguins do not exist under propositional logic rules. However, we know that they do and are an exception to the rule that $p \to f$. Real-world reasoning often involves exceptions and retractable conclusions. Defeasible reasoning, a form of nonmonotonic logic, deals with this by retracting conclusions when contradictory evidence arises. Numerous formalisations of defeasible reasoning have been defined. However, this literature review is going to focus on the KLM framework [18] getting extended by *rational closure* [9] and *lexicographic closure* [20].

3.1 Preferential Approach

The preferential approach was defined by Shoham [27] in 1987 that states for a formula α to be satisfiable, it should be satisfied by all of the models of α that are preferred the most. Semantically this would be defined by saying "α preferentially entails β", denoted as $\{\alpha\} \mathrel{|\!\approx} \beta$, if and only if all preferential models of α also satisfy β. Gabbay [13] defined a corresponding proof-theoretic system for preferential reasoning. KLM framework expanded on both of these approaches [17].

3.2 The KLM Framework

KLM [18] defines "typicality" by arguing that nonmonotonic logic should be able to say "x is typically y". This is a meta-level consequence relation, denoted by $\mathrel{|\!\approx}$. They defined six postulates that consequence relations must satisfy to be considered a *preferential consequence relation*, these are then modelled using *preferential interpretations* [27]. A seventh postulate was added in [20] to deal with conflicts in defaults developing a new set of consequence relations *rational consequence relations*. These are modelled similarly to *preferential consequence relation*, called *ranked interpretations* that define the semantics of the relations [21].

3.2.1 Ranked Interpretations.

A Ranked interpretation is defined as [5]

$$\mathcal{R} : \mathcal{U} \to \mathcal{N} \cup \infty$$

Which satisfies the property that for every $i \in \mathcal{N}$, if there exists an $u \in \mathcal{U}$ such that $\mathcal{R}(u) = i$ then there exists a $v \in \mathcal{U}$ such that $0 \leq \mathcal{R}(v) < i$. This means that within $\mathcal{R}$ each valuation of atoms is ranked from 0 to ∞, where 0 is considered "typical" and ∞ is considered "impossible".

To illustrate this, let $\mathcal{P} := \{p, b, f, w\}$. Where p = "penguin", b = "bird", f = "fly", and w = "wings". Below is one possible ranked interpretation:

∞	$pbfw$
2	$pbf\overline{w}$ $pb\overline{f}w$ $p\overline{b}fw$ $\overline{p}bfw$
1	$pb\overline{f}\overline{w}$ $p\overline{b}f\overline{w}$ $p\overline{b}\overline{f}w$ $\overline{p}bf\overline{w}$ $\overline{p}b\overline{f}w$ $\overline{p}\overline{b}fw$
0	$p\overline{b}\overline{f}\overline{w}$ $\overline{p}b\overline{f}\overline{w}$ $\overline{p}\overline{b}f\overline{w}$ $\overline{p}\overline{b}\overline{f}w$ $\overline{p}\overline{b}\overline{f}\overline{w}$

Figure 3: A possible ranked interpretation of $\mathcal{P} := \{p, b, f, w\}$

This is only one of many possible ranked interpretations of $\mathcal{P}$. Above it shows $\mathcal{R}(pbfw) = \infty$ and therefore considered impossible. Similarly, $\mathcal{R}(p\overline{b}\overline{f}\overline{w}) = \mathcal{R}(\overline{p}b\overline{f}\overline{w}) = \mathcal{R}(\overline{p}\overline{b}f\overline{w}) = \mathcal{R}(\overline{p}\overline{b}\overline{f}w) = \mathcal{R}(\overline{p}\overline{b}\overline{f}\overline{w}) = 0$ and therefore most likely in the typical world. Further, $u := pb\overline{f}\overline{w}$ would be preferred over $v := pbf\overline{w}$ because $\mathcal{R}(u) < \mathcal{R}(v)$

3.2.2 Defeasible Implication.

Until this point, we have treated $\mathrel{|\!\sim}$ as a meta-level consequence relation over a classical propositional language, but now that an entailment relation is to be defined in preferential semantics, $\mathrel{|\!\sim}$ needs to be considered an object-level connector. This is done by formalising a new language:

$$\mathcal{L}_P := \mathcal{L} \cup \{\alpha \mathrel{|\!\sim} \beta | \alpha, \beta \in \mathcal{L}\}$$

This defines propositional logic with the added connective $\mathrel{|\!\sim}$. $\mathrel{|\!\sim}$ is read as "typically implies" and is the counterpart to the propositional logic connective $\to$. A set of defeasible implications, $\alpha \mathrel{|\!\sim} \beta$ forms a defeasible knowledge base, $\mathcal{K}$, also referred to as a conditional knowledge base [17].

3.2.3 Satisfiability.

To define the notion of entailment this section will revert to the use of *preferential semantics.* First we need to define satisfaction in a preferential interpretation[17]: given a preferential interpretation $\mathcal{P}$ and defeasible implication $\alpha \mathrel{|\!\sim} \beta$, $\mathcal{P}$ satisfies $\alpha \mathrel{|\!\sim} \beta$, written $\mathcal{P} \Vdash \alpha \mathrel{|\!\sim} \beta$, if and only if for every u minimal in $[\![\alpha]\!]^{\mathcal{P}}$, $u \Vdash \beta$. If $\mathcal{P} \Vdash \alpha \mathrel{|\!\sim} \beta$, then $\mathcal{P}$ is said to be a model of $\alpha \mathrel{|\!\sim} \beta$. Meaning that for $\mathcal{P}$ to satisfy $\alpha \mathrel{|\!\sim} \beta$, all the minimal worlds that satisfy α must satisfy β.

This definition can be extended to knowledge bases by saying $\mathcal{K}$ is satisfied by a preferential interpretation $\mathcal{P}$, written $\mathcal{P} \Vdash \mathcal{K}$, if and only if every formula in $\mathcal{K}$ is satisfied by $\mathcal{P}$. In this case, $\mathcal{P}$ is considered a model of $\mathcal{K}$ [17].

Defeasible knowledge bases may contain both classical propositional formulas and defeasible formulas by expressing classical formulas as defeasible implications. Denoted as:

$$\neg\alpha \mathrel{|\!\sim} \bot \text{ for any } \mathcal{P}, \mathcal{P} \Vdash \alpha \text{ if and only if } \mathcal{P} \Vdash \neg\alpha \mathrel{|\!\sim} \bot$$

However, $\bot$ is the propositional constant that represents *false* meaning there are no minimal states where α is false, and therefore α is true in $\mathcal{P}$. This shows that any mention of defeasible implication may also refer to classical statements.[7].

3.2.4 Entailment.

[18] proposed six postulates that consequence relations must satisfy to be considered a *preferential consequence relation*, denoted as $\mathrel{|\!\sim}$. Lehmann [20] added a seventh postulate, Rational Monotony (RM), that refines what can be included in a consequence relation. [7] later reformulates these postulates to characterise *defeasible entailment*, denoted by $\mathrel{|\!\approx}$, at the meta-leval.

$$\textbf{(LLE)}\quad \frac{\mathcal{K} \models \alpha \leftrightarrow \beta, \mathcal{K} \approx \alpha \mid\!\sim \gamma}{\mathcal{K} \approx \beta \mid\!\sim \gamma} \qquad \textbf{(Or)}\quad \frac{\mathcal{K} \approx \alpha \mid\!\sim \gamma, \mathcal{K} \approx \beta \mid\!\sim \gamma}{\mathcal{K} \approx \alpha \vee \beta \mid\!\sim \gamma}$$

$$\textbf{(RW)}\quad \frac{\mathcal{K} \models \alpha \to \beta, \mathcal{K} \approx \gamma \mid\!\sim \alpha}{\mathcal{K} \approx \gamma \mid\!\sim \beta} \qquad \textbf{(CM)}\quad \frac{\mathcal{K} \approx \alpha \mid\!\sim \gamma, \mathcal{K} \approx \alpha \mid\!\sim \beta}{\mathcal{K} \approx \alpha \wedge \beta \mid\!\sim \gamma}$$

$$\textbf{(Ref)}\quad \mathcal{K} \approx \alpha \mid\!\sim \alpha \qquad \textbf{(RM)}\quad \frac{\mathcal{K} \approx \alpha \mid\!\sim \gamma, \mathcal{K} \not\approx \alpha \not\mid\!\sim \beta}{\mathcal{K} \approx \alpha \wedge \beta \mid\!\sim \gamma}$$

$$\textbf{(And}\quad \frac{\mathcal{K} \approx \alpha \mid\!\sim \beta, \mathcal{K} \approx \alpha \mid\!\sim \gamma}{\mathcal{K} \approx \alpha \mid\!\sim \beta \wedge \gamma}$$

Any defeasible entailment relation, $\approx$, that satisfies the above KLM postulates is considered *LM-rational* [7]. Lehmann and Magidor showed that for any defeasible entailment relation to be satisfied it is a necessary condition that it is *LM-rational*. Later on, we will discuss if this is an efficient condition. The KLM framework defines *preferential entailment* as an entailment relation conforming to the first six postulates [18]. Later, Lehmann and Magidor extended this to show *ranked entailment* could be defined using ranked interpretations using all seven postulates [20]. However, some rationales exist suggesting that ranked entailment is identical to preferential entailment [21] and that neither is *LM-rational* [7] because they do not conform to RM.

A significant limitation of both of these entailments is that they are still monotonic, and since the goal is nonmonotonic we need to extend the KLM framework to explore other options.

Before we can extend defeasible reasoning within the KLM framework, we need to define ranked entailment[18]: a knowledge base $\mathcal{K}$ rank entails a defeasible implication $\alpha \mid\!\sim \beta$ if every ranked interpretation satisfying $\mathcal{K}$ also satisfies $\alpha \mid\!\sim \beta$, and vice versa. However, as mentioned above, this form of defeasible entailment is still monotonic [7].

3.2.5 Basic Defeasible Entailment. In [6] To explore whether *LM-rationality* is sufficient Casini et al. expanded on the KLM framework by proposing a class of *basic defeasible entailment relations* that not only satisfies LM-rationality, but also Inclusion, and Classic Preservation.

$$\textbf{(Inclusion)}\quad \mathcal{K} \approx \alpha \mid\!\sim \beta \text{ for every } \alpha \mid\!\sim \beta \in \mathcal{K}$$

$$\textbf{(Classic Preservation)}\quad \mathcal{K} \approx \alpha \mid\!\sim \bot \text{ if and only if } \mathcal{K} \approx_R \alpha \mid\!\sim \bot$$

All elements of $\mathcal{K}$ have to be defeasibly entailed by $\mathcal{K}$ to achieve *Inclusion* and *Classical Preservation* requires classical defeasible implications that are defeasibly entailed by $\mathcal{K}$ to correspond exactly to those that are rank entailed by $\mathcal{K}$. A corollary of this is Classic Consistency, which argues that a knowledge base can be considered consistent if and only if it is consistent regarding rank entailment.

3.2.6 Rational Defeasible Entailment. After analysing *Basic defeasible entailment* [6] proceeded to claim it is too permissive. Noting that the defeasible implications entailed by *Basic defeasible entailment* are not always the same as *Rational Closure* (discussed later). This implies the need for a stronger class of defeasible entailment relations that apply rational closure as a nonmonotonic core. This requires the following property to be defined [6]:

$$\textbf{(RC Extension)}\quad \mathit{If}\, \mathcal{K} \approx_{RC} \alpha \mid\!\sim \beta, \text{then } \mathcal{K} \approx \alpha \mid\!\sim \beta$$

The *Rational Closure Extension* requires a defeasible entailment relation to extend the rational closure of $\mathcal{K}$. The subset of basic defeasible entailment relations that satisfy *RC extension* are considered *Rational Defeasible Entailment*

3.3 Rational Closure[7]

Rational closure (RC) can be defined either semantically, using a *minimal ranked model*, or syntactically, via *base ranks* [17]. The former considers typicality *between* ranked models of a knowledge base K, while the latter examines typicality *within* ranked models of K [6].

3.3.1 Minimal Ranked Model. To define rational closure semantically, we introduce a partial order $\preceq_{\mathcal{K}}$ overall ranked models of a knowledge base $\mathcal{K}$: given $R_1, R_2 \in Mod(\mathcal{K})$, we say that $R_1 \preceq_{\mathcal{K}} R_2$ if and only if for every $u \in \mathcal{V}$, $R_1(u) \leq R_2(u)$. Intuitively, a ranked model that assigns lower ranks to all valuations is considered more typical [6].

It has been shown that this ordering has a unique minimal element, denoted $\mathcal{R}^{RC}_{\mathcal{K}}$, even if the size and structure of the set differ [14]. The rational closure of $\mathcal{K}$ is then defined as follows: a defeasible implication $\alpha \mid\!\sim \beta$ belongs to the rational closure of $\mathcal{K}$ (denoted $\mathcal{K} \approx_{RC} \alpha \mid\!\sim \beta$) if and only if it holds in $\mathcal{R}^{RC}_{\mathcal{K}}$.

Since any LM-rational entailment relation is generated from a ranked interpretation [21], rational closure is LM-rational, as it is only defined by $\mathcal{R}^{RC}_{\mathcal{K}}$.

To illustrate this consider $\mathcal{K} = \{p \to b, b \mid\!\sim f, b \mid\!\sim w, p \mid\!\sim \neg f\}$, which states that penguins are birds, birds typically fly, birds typically have wings, and penguins do not typically fly. The minimal ranked model $\mathcal{R}^{RC}_{\mathcal{K}}$ of $\mathcal{K}$ would be constructed as [7]:

∞	$\bar{b}\bar{f}p\bar{w}$ $\bar{b}fpw$ $\bar{b}fp\bar{w}$ $\bar{b}\bar{f}pw$
2	$bfpw$ $bfp\bar{w}$
1	$b\bar{f}\bar{p}w$ $b\bar{f}\bar{p}\bar{w}$ $b\bar{f}pw$ $b\bar{f}p\bar{w}$
0	$bf\bar{p}w$ $\bar{b}f\bar{p}w$ $\bar{b}f\bar{p}\bar{w}$ $\bar{b}\bar{f}\bar{p}w$ $\bar{b}\bar{f}\bar{p}\bar{w}$ $bf\bar{p}\bar{w}$

Figure 4: The minimal ranked model $\mathcal{R}^{RC}_{\mathcal{K}}$ of $\mathcal{K}$

Given the query $p \mid\!\sim w$ that questions if penguins typically have wings, we check if all minimal p-worlds (circled above) satisfy w. Since $b\bar{f}p\bar{w}$ does not, we conclude $\mathcal{K} \not\approx_{RC} p \mid\!\sim w$, meaning rational closure does not conclude that penguins have wings.

3.3.2 Base Ranks. Syntactically, rational closure can be defined using base ranks. A formula $\alpha \in \mathcal{L}$ is *exceptional* concerning a knowledge base $\mathcal{K}$ if $\mathcal{K} \approx_R \top \mid\!\sim \neg\alpha$, which means that the negation of α is typical in every preferential interpretation that is satisfied in $\mathcal{K}$. This is to say that α is false in all minimal valuations in every ranked model of $\mathcal{K}$ [7]. Equivalently, α is exceptional if its negation follows from the counterpart, referred to as *materialisation*, of $\mathcal{K}$, defined as:

$$\overrightarrow{\mathcal{K}} := \{\alpha \to \beta \mid \alpha \mid\!\sim \beta \in \mathcal{K}\}[17].$$

We then define a sequence of knowledge bases$\mathcal{E}_0^{\mathcal{K}}, \ldots, \mathcal{E}_\infty^{\mathcal{K}}$ as follows [7]:

$$\mathcal{E}_0^{\mathcal{K}} := \mathcal{K}$$
$$\mathcal{E}_i^{\mathcal{K}} := \varepsilon(\mathcal{E}_{i-1}^{\mathcal{K}}) \text{ for } 0 < i < n$$
$$\mathcal{E}_\infty^{\mathcal{K}} := \mathcal{E}_n^{\mathcal{K}}$$

Note that formulas in $\mathcal{E}_\infty^{\mathcal{K}}$ represent the formulas in $\mathcal{K}$ that are logically equivalent to $\perp$ and are considered a Classical Formula. The base rank of a formula α, denoted $br_{\mathcal{K}}(\alpha)$, is the smallest r such that α is not exceptional in $\mathcal{E}_r^{\mathcal{K}}$. The base rank of a defeasible implication is the base rank of its antecedent, i.e., $br_{\mathcal{K}}(\alpha \mid\sim \beta) := br_{\mathcal{K}}(\alpha)$ [17].

To illustrate this, recall the knowledge base $\mathcal{K} = \{p \rightarrow b, b \mid\sim f, b \mid\sim w, p \mid\sim \neg f\}$. This will provide the following base ranks:

$\mathcal{E}_0^{\mathcal{K}} = \{p \rightarrow b, b \mid\sim f, b \mid\sim w, p \mid\sim \neg f\}$ (by definition)

$\mathcal{E}_1^{\mathcal{K}} = \{p \rightarrow b, p \mid\sim \neg f\}$ (since $\overrightarrow{\mathcal{E}_0^{\mathcal{K}}} \models \neg p, p \rightarrow b$ but $\overrightarrow{\mathcal{E}_0^{\mathcal{K}}} \not\models \neg b$)

$\mathcal{E}_2^{\mathcal{K}} = \{p \rightarrow b\}$ (since $\overrightarrow{\mathcal{E}_1^{\mathcal{K}}} \models p \rightarrow b$ but $\overrightarrow{\mathcal{E}_1^{\mathcal{K}}} \not\models \neg p$)

$\mathcal{E}_\infty^{\mathcal{K}} = \mathcal{E}_2^{\mathcal{K}}$ (since $\overrightarrow{\mathcal{E}_2^{\mathcal{K}}} \models p \rightarrow b$ again)

A visualisation of the above ranking is:

R_∞	$p \rightarrow b$
R_1	$p \rightarrow \neg f$
R_0	$b \rightarrow f$ $\quad$ $b \rightarrow w$

Figure 5: The partitions of $\overrightarrow{\mathcal{K}}$ produced by `BaseRank`

Using the definition of base ranks, rational closure is defined as follows [14]: $\mathcal{K} \models_{RC} \alpha \mid\sim \beta$ if and only if $br_{\mathcal{K}}(\alpha) < br_{\mathcal{K}}(\alpha \wedge \neg\beta)$ or $br_{\mathcal{K}}(\alpha) = \infty$. This allows the base rank of a propositional formula to act on the syntax of the defeasible entailment relation directly. [14] observed that rational closure and minimal ranked entailment will give the same set of inferences.

3.3.3 Algorithm for RC Entailment Queries. The rational closure entailment algorithm consists of two stages: `BaseRank`, which partitions the knowledge base into rank-based subsets, and `RationalClosure`, which evaluates entailment [7]. Given a query $\alpha \mid\sim \beta$, the algorithm:

(1) Computes base rank partitions $R_0, \ldots, R_n, R_\infty$ using `BaseRank`.
(2) Checks if $\alpha \mid\sim \beta$ is exceptional in $R_0 \cup \cdots \cup R_\infty$.
(3) Iteratively removes lower-ranked components until $\alpha \mid\sim \beta$ is no longer exceptional or only R_∞ remains.
(4) Checks if $\overrightarrow{R_i \cup \cdots \cup R_\infty} \models \alpha \rightarrow \beta$.

If the final entailment holds, the query is in the rational closure of $\mathcal{K}$, otherwise, it is not [12]. This algorithm ensures that rational closure remains computationally tractable while capturing the most conservative defeasible conclusions.

3.4 Lexicographic Closure[9]

Lexicographic closure (LC) was first introduced in [20] as a refinement of rational closure, formalising the pattern of default reasoning described in [25]. They started by describing a set of intuitive properties that should guide lexicographical closure as they underpin default reasoning. Like rational closure, lexicographic closure can be characterised using ranked models, specifically the *lexicographic ranked model* [7]. However, there are multiple versions of lexicographic closure based on different ranking mechanisms [11]. We examine two such definitions—one from [20] and another from [7]—and explore an alternative syntactic formulation using a lexicographic rank function.

3.4.1 Lexicographic Ranked Model in [20]. The lexicographic ranking approach introduced by Lehmann is based on evaluating the severity of violations in a knowledge base $\mathcal{K}$. The *order* of $\mathcal{K}$ is defined as the highest base rank of any formula in $\mathcal{K}$, excluding ∞ [20]. Shown as:

$$K^K := \max\{\mathrm{br}(\alpha) \mid \alpha \mid\sim \beta \in K, \mathrm{br}(\alpha) \in \mathrm{N}\} \qquad (1)$$

Using this, we assign each subset $D \subseteq \mathcal{K}$ a tuple $n_D = \langle n_0, n_1, \ldots, n_{\mathcal{K}} \rangle$, where:

- n_0 is the number of defeasible implications in D with a base rank of ∞.
- n_i (for $i > 0$) is the number of defeasible implications in D with a base rank of $\mathcal{K} - i$.

These tuples are then used to impose a lexicographic order $\prec_S$ over subsets of $\mathcal{K}$. Given valuations $u, v \in \mathcal{V}$, we define the lexicographic ordering $\preceq_{LC}$ as follows: $u \preceq_{LC} v$ if and only if $V(u) \preceq_S V(v)$, where $V(x)$ is the set of defeasible implications violated by valuation x [17]. This ordering induces a ranked interpretation $\mathcal{R}_{\mathcal{K}}^{LC}$, which defines entailment such that $\mathcal{K} \models_{LC} \alpha \mid\sim \beta$ if and only if $\mathcal{R}_{\mathcal{K}}^{LC} \models \alpha \mid\sim \beta$.

3.4.2 Lexicographic Ranked Model in [7]. An alternative lexicographic ranking was introduced in [7]. Here, we define a function $C^{\mathcal{K}} :\rightarrow \mathbb{N}$ such that $C^{\mathcal{K}}(v)$ counts the number of defeasible implications in $\mathcal{K}$ that are satisfied by v. The lexicographic ordering $\preceq_{LC}^{K}$ over valuations, , is then defined as follows:

- If $\mathcal{R}_{\mathcal{K}}^{RC}(v) = \infty$, then $u \preceq_{LC} v$.
- If $\mathcal{R}_{\mathcal{K}}^{RC}(u) < \mathcal{R}_{\mathcal{K}}^{RC}(v)$, then $u \preceq_{LC} v$.
- If $\mathcal{R}_{\mathcal{K}}^{RC}(u) = \mathcal{R}_{\mathcal{K}}^{RC}(v)$, then $u \preceq_{LC} v$ if and only if $C^{\mathcal{K}}(u) \geq C^{\mathcal{K}}(v)$.

Using this ordering, a ranked interpretation $\mathcal{R}_{\mathcal{K}}^{LC}$ is generated, and lexicographic entailment is defined such that [7]:

$$\mathcal{K} \models_{LC} \alpha \mid\sim \beta \text{ if } \mathcal{R}_{\mathcal{K}}^{LC} \models \alpha \mid\sim \beta.$$

Both ranking mechanisms refine rational closure by ranking valuations within the minimal ranked model $\mathcal{R}_{\mathcal{K}}^{RC}$. However, Lehmann's approach ranks valuations by the severity of their violations, while Casini et al.'s method further refines this by considering the number of satisfied defeasible implications. These approaches are not equivalent, and a detailed comparison can be found in [11].

3.4.3 Lexicographic Rank. Since lexicographic closure is based on a ranked model that preserves $\mathcal{R}_{\mathcal{K}}^{RC}$'s structure, it can also be defined using a rank function. The *lexicographic rank*, denoted $r_{\mathcal{K}}^{LC}$, is given by:

$$r_{\mathcal{K}}^{LC}(\alpha) := \min\{\mathcal{R}_{\mathcal{K}}^{LC}(v) \mid v \in [[\alpha]]\}.$$

Using $r^{LC}_{\mathcal{K}}$, lexicographic closure can be defined syntactically [7]:

$$\mathcal{K} \approx_{LC} \alpha \mid\!\sim \beta \text{ if and only if } r^{LC}_{\mathcal{K}}(\alpha) < r^{LC}_{\mathcal{K}}(\alpha \land \neg\beta) \text{ or } r^{LC}_{\mathcal{K}}(\alpha) = \infty.$$

3.4.4 Algorithm for LC Entailment Queries. The algorithm for computing lexicographic closure is a generalisation of the rational closure algorithm [6]. The `DefeasibleEntailment` algorithm takes a knowledge base $\mathcal{K}$, a rank function r, and a query $\alpha \mid\!\sim \beta$, and returns $\mathcal{K} \approx_r \alpha \mid\!\sim \beta$. When instantiated with $r^{LC}_{\mathcal{K}}$, it computes lexicographic closure [7].

The algorithm consists of:

(1) Computing rank partitions $R_0, \ldots, R_\infty$ using Rank.
(2) Checking whether $\alpha \mid\!\sim \beta$ is exceptional in $R_0 \cup \cdots \cup R_\infty$.
(3) Removing lower-ranked partitions until $\alpha \mid\!\sim \beta$ is no longer exceptional or only R_∞ remains.
(4) Checking if $\overrightarrow{R_i \cup \cdots \cup R_\infty} \models \alpha \rightarrow \beta$.

The algorithm returns true if and only if $\mathcal{K} \approx_{LC} \alpha \mid\!\sim \beta$ [7].

`DefeasibleEntailment` is an extension of `RationalClosure` however it makes stronger presumptive inferences. To show this we will refer back to the example $\mathcal{K} = \{p \rightarrow b, b \mid\!\sim f, b \mid\!\sim w, p \mid\!\sim \neg f\}$ that will give us the lexicographic ranked model $r^{LC}_{\mathcal{K}}$ below [7]: Comparing the two models, rational closure does not conclude $p \mid\!\sim$

∞	$\overline{b}fp\overline{w}$ $\overline{bf}pw$ $\overline{bf}p\overline{w}$ $\overline{b}fpw$
5	$bfp\overline{w}$
4	$bfpw$
3	$b\overline{f}p\overline{w}$ $b\overline{f}\overline{pw}$
2	$b\overline{f}pw$ $b\overline{f}\overline{p}w$
1	$bf\overline{pw}$
0	$bf\overline{p}w$ $\overline{b}f\overline{p}w$ $\overline{b}f\overline{pw}$ $\overline{bf}\overline{p}w$ $\overline{bf}\overline{pw}$

Figure 6: The lexicographic ranked model $\mathcal{R}^{LC}_{\mathcal{K}}$ of $\mathcal{K}$

w, while lexicographic closure does. This difference illustrates that rational closure is more conservative, while lexicographic closure assumes typical properties unless explicitly contradicted.

Rather than identifying a single "correct" defeasible entailment relation, these approaches offer different reasoning paradigms. Rational closure aligns with prototypical reasoning, where general conclusions are drawn conservatively, whereas lexicographic closure embodies a presumptive approach, inferring typical properties more aggressively [17].

4 PROTOTYPE VERSUS PRESUMPTIVE

Prototypical and presumptive entailment differ in their approach to default reasoning [20]. While both are nonmonotonic reasoning strategies, they rely on distinct methodologies for handling exceptions and defaults. Default logic and preferential reasoning provide formal foundations for these approaches [12, 25].

4.0.1 Prototypical entailment. Prototypical entailment constructs a *perfect image* of what an entity should be like. If an instance does not conform to this predefined model, it is classified as atypical and is treated as completely exceptional. This method enforces strict category boundaries, which makes it particularly useful in contexts requiring well-defined classifications, such as taxonomy and ontology development.

4.0.2 Presumptive entailment. In contrast, presumptive entailment dynamically *analyses individual exceptions* and prioritises them based on contextual relevance. Rather than discarding an atypical case outright, it seeks to reason about the exception itself. This flexibility is advantageous in domains where uncertainty is prevalent and gradual adaptation is essential, such as legal interpretation and medical diagnosis.

5 CURRENT WORK

Defeasible reasoning has been integrated into several computational frameworks, each emphasising different aspects of nonmonotonic reasoning. Some of the most notable implementations include:

5.1 Other Implementations of Defeasible Reasoning

5.1.1 Description Logics . Description logics (DLs) [8] provide a formal foundation for structured knowledge representation. Extensions of DLs incorporate defeasible reasoning mechanisms to manage exceptions within ontologies. This is particularly useful in semantic web applications and AI-driven classification systems.

5.1.2 Model-Based Defeasible Reasoning . Model-based approaches construct ranked models to determine entailment dynamically[11]. These methods leverage preferential structures to establish a hierarchy of defaults, allowing for more fine-grained reasoning about exceptions. Model-based defeasible reasoning is frequently applied in AI systems requiring adaptive decision-making.

5.1.3 Cognitive Defeasible Reasoning . Cognitive approaches to defeasible reasoning model human-like patterns of inference. These methods emphasise context-sensitive learning, where an agent refines their reasoning strategies based on experience [1]. Such approaches are promising for human-computer interaction and AI explainability.

Computational trade-offs must be considered when implementing defeasible reasoning in AI [5]. The balance between expressivity and tractability remains an ongoing research challenge. Future work may explore integrating defeasible reasoning with machine learning models to enhance automated decision-making under uncertainty.

5.2 Knowledge Base Generator

Knowledge Base Generators (KBGs) play a crucial role in artificial intelligence by automating the construction of knowledge-based systems. These generators allow for the creation of domain-specific knowledge bases that can be adapted to various problem-solving contexts. By leveraging predefined templates and rule-based configurations, KBGs enhance the efficiency and accuracy of knowledge representation. There have been numerous KBGs developed across different fields. These are a few notable ones:

[19] presents a knowledge-based report generator, which utilises predefined templates to automatically generate reports. Such tools are particularly useful in domains requiring structured data interpretation and reporting, reinforcing the versatility of KBGs across multiple applications.

[16] discusses a knowledge-based programme generator designed for specific problem domains, illustrating how such generators can streamline the design process by tailoring the knowledge representation to application-specific requirements.

[4] provides a comprehensive review of defeasible reasoning implementations, examining how KBGs can assist in handling incomplete or uncertain information. Their work underscores the importance of integrating defeasible reasoning capabilities within KBGs to ensure dynamic updates in response to evolving knowledge.

[23] explores the concept of dynamically generating knowledge-based systems, emphasising the adaptability and efficiency of KBGs in the development of systems. This approach facilitates the automatic structuring of knowledge bases, reducing the effort required to design and implement complex AI systems.

5.3 User Interface and Debugging Tool

The usability of defeasible reasoning systems heavily depends on the availability of intuitive user interfaces (UIs) and debugging tools. These tools enable users to interact with knowledge bases, execute queries, and identify logical inconsistencies, thereby improving the reliability of the reasoning process. There have been many approaches developed to solve this:

[**?**] introduce the *SILK* graphical user interface, designed specifically for defeasible reasoning. This UI provides users with a visual representation of reasoning paths, allowing them to explore different conclusions and understand how reasoning outcomes are derived. The ability to trace the reasoning process enhances both accessibility and interpretability, making defeasible reasoning more accessible.

A Protégé tool, OntoDebug, has been developed by [26] that uses an interactive approach to ontology debugging. It asks the user questions to collect the missing information. However, unintuitive results can occur when ‘multi-level exceptions’. [10] extends on [26] to resolve these inconsistencies by focusing on debugging tools for classical ontologies, emphasising the need for systematic approaches to identifying and resolving errors in defeasible reasoning systems. They argue that integrating debugging functionalities with reasoning tools ensures consistency and correctness in knowledge-based applications.

6 CONCLUSIONS

In this paper, we have explored the limitations of rational closure in defeasible reasoning and proposed methods to extend its applicability. Our analysis highlights that while rational closure provides a solid foundation for reasoning with exceptions, it often fails to account for nuanced distinctions between different defeasible statements.

By incorporating additional inference mechanisms, such as lexicographic closure [7, 20], we demonstrated that more refined approaches can better capture real-world reasoning patterns. Furthermore, the role of preferential models [18] in the extension of defeasible logic was discussed, showing that a more flexible hierarchy of reasoning principles allows for improved expressivity.

Future work includes exploring hybrid approaches that blend different closure operations, as well as investigating computational efficiencies in applying these methods in large-scale reasoning systems. There is also significant potential for further exploration of nonmonotonic knowledge base generators, particularly in how they influence the construction and adaptation of ranked interpretations. Investigating these generators could provide deeper insights into reasoning mechanisms beyond standard rational closure, especially in handling exceptions, contradictions, and dynamic knowledge updates. Ultimately, by extending defeasible reasoning beyond rational closure, we aim to contribute towards more sophisticated and practically viable nonmonotonic reasoning systems.

REFERENCES

[1] Baker, C. K., Denny, C., Freund, P., and Meyer, T. Cognitive defeasible reasoning: the extent to which forms of defeasible reasoning correspond with human reasoning. In *Southern African Conference for Artificial Intelligence Research* (2020), Springer, pp. 199–219.

[2] Ben-Ari, M. *Mathematical logic for computer science.* Springer Science & Business Media, 2012.

[3] Bisson, G. Kbg: A generator of knowledge bases[1]. In *Machine Learning-EWSL-91: European Working Session on Learning, Porto, Portugal, March 6-8, 1991. Proceedings* (1991), vol. 482, Springer Science & Business Media, p. 137.

[4] Bryant, D., and Krause, P. A review of current defeasible reasoning implementations. *The Knowledge Engineering Review 23*, 3 (2008), 227–260.

[5] Casini, G., Harrison, M., Meyer, T., and Swan, R. Arbitrary ranking of defeasible subsumption.

[6] Casini, G., Meyer, T., and Varzinczak, I. Defeasible entailment: From rational closure to lexicographic closure and beyond. In *Proceeding of the 17th International Workshop on Non-Monotonic Reasoning (NMR 2018)* (2018), pp. 109–118.

[7] Casini, G., Meyer, T., and Varzinczak, I. Taking defeasible entailment beyond rational closure. In *Logics in Artificial Intelligence: 16th European Conference, JELIA 2019, Rende, Italy, May 7–11, 2019, Proceedings 16* (2019), Springer, pp. 182–197.

[8] Casini, G., and Straccia, U. Rational closure for defeasible description logics. In *Logics in Artificial Intelligence: 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings 12* (2010), Springer, pp. 77–90.

[9] Casini, G., and Straccia, U. Lexicographic closure for defeasible description logics. In *Proc. of Australasian Ontology Workshop* (2012), vol. 969, pp. 28–39.

[10] Coetzer, S., and Britz, K. Debugging classical ontologies using defeasible reasoning tools. In *Formal Ontology in Information Systems (FOIS 2022)* (2022), vol. 344, IOS Press, pp. 97–106.

[11] Cohen, J., Combrinck, C., and Meyer, T. Model-based defeasible reasoning, 2022.

[12] Freund, M. Preferential reasoning in the perspective of poole default logic. *Artificial Intelligence 98*, 1-2 (1998), 209–235.

[13] Gabbay, D. M. *Theoretical foundations for non-monotonic reasoning in expert systems.* Springer, 1985.

[14] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence 226* (2015), 1–33.

[15] Grosof, B., Burstein, M., Dean, M., Andersen, C., Benyo, B., Ferguson, W., Inclezan, D., and Shapiro, R. A silk graphical ui for defeasible reasoning, with a biology causal process example. In *9th International Semantic Web Conference ISWC 2010* (2010), Citeseer, p. 113.

[16] Hrycej, T. A knowledge-based problem-specific program generator. *ACM SIGPLAN Notices 22*, 2 (1987), 53–61.

[17] Kaliski, A. An overview of klm-style defeasible entailment. *NA* (2020).

[18] Kraus, S., Lehmann, D., and Magidor, M. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence 44*, 1-2 (1990), 167–207.

[19] Kukich, K. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics* (1983), pp. 145–150.

[20] Lehmann, D. Another perspective on default reasoning. *Annals of mathematics and artificial intelligence 15* (1995), 61–82.

[21] Lehmann, D., and Magidor, M. What does a conditional knowledge base entail? *Artificial intelligence 55*, 1 (1992), 1–60.

[22] Levesque, H. J. Knowledge representation and reasoning. *Annual review of computer science 1*, 1 (1986), 255–287.

[23] Moisan, S. Generating knowledge-based system generators. *Insights into Advancements in Intelligent Information Technologies: Discoveries: Discoveries* (2012), 1.

[24] Pollock, J. L. Defeasible reasoning. *Cognitive science 11*, 4 (1987), 481–518.

[25] Reiter, R. A logic for default reasoning. *Artificial intelligence 13*, 1-2 (1980), 81–132.

[26] Schekotihin, K., Rodler, P., and Schmid, W. Ontodebug: Interactive ontology

debugging plug-in for protégé. In *Foundations of Information and Knowledge Systems: 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14–18, 2018, Proceedings 10* (2018), Springer, pp. 340–359.

[27] Shoham, Y. A semantical approach to nonmonotonic logics. In *Readings in Non-Monotonic Reasoning*, M. L. Ginsberg, Ed. Morgan Kaufmann, 1987, pp. 227–249.