Extending Defeasible Reasoning beyond Rational Closure

Nevaniah Gounden
gndnev002@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

Defeasible reasoning provides a logical structure that reflects the dynamic nature of real-world decision-making, enabling conclusions to adjust as new information and circumstances arise. Several formal frameworks have been developed to model defeasible reasoning, with rational closure standing as one of the most prominent being researched. Lexicographic closure has also received significant attention, offering a more nuanced approach. Both frameworks characterize forms of nonmonotonic entailment, allowing conclusions to be drawn from default assumptions while remaining open to revision in the presence of more specific or higher-priority information. This literature review aims to deepen the understanding of current rational closure and lexicographic closure algorithms, asses their computational characteristics, and explore methods for improving their efficiency. The findings aim to provide insights into the performance, strengths, and limitations of these algorithms.

CCS CONCEPTS

• Theory of computation → Automated reasoning; **Logic and verification**; **• Computing methodologies → Knowledge representation and reasoning**; **Nonmonotonic, default reasoning and belief revision**.

KEYWORDS

artificial intelligence, knowledge representation and reasoning, nonmonotonic reasoning, defeasible reasoning, rational closure, lexicographic closure, computational efficiency, algorithmic complexity

1 INTRODUCTION

Artificial intelligence (AI) has grown rapidly and now plays a central role in daily life. Knowledge representation (KR) encodes information in structured, machine-readable formats, enabling intelligent systems to reason, make decisions, and solve complex problems [14]. Effective KR supports inference, learning, planning, and communication, making it essential in fields such as natural language processing, autonomous systems, and decision support [2, 5, 15]. While many KR schemes exist, this paper focuses on *propositional logic*, chosen for its simplicity and widespread use in AI applications.

Propositional logic represents knowledge using atomic formulas that are either true or false, combined with logical connectives such as conjunction, disjunction, and negation [3]. However, its monotonic nature limits its capacity to handle exceptions: once a conclusion is drawn, it cannot be withdrawn. For example, while the general rule “cars run on petrol” may hold, it fails to account for specific exceptions like “electric cars do not use petrol.” This motivates the use of *defeasible reasoning*, which supports revisable conclusions in light of more specific or contradictory information [26].

The KLM framework, developed by Kraus, Lehmann, and Magidor [21], provides a formal basis for defeasible reasoning through *preferential models*, which rank scenarios based on typicality. Conclusions are drawn from the most typical models, with exceptions handled naturally. Building on this, *rational closure* ranks defaults by specificity, ensuring that more specific defaults override general ones [24]. While rational closure is cautious and provides a systematic ranking of defaults, stronger reasoning strategies such as lexicographic closure consider not only the rank but also the number of satisfied defaults, thereby resolving conflicts more decisively[9, 23].

Constructing and evaluating complex conditional knowledge bases for such reasoning tasks can be prone to errors and time-consuming. *Knowledge base generators* (KBGs) aim to automate this process by supporting consistent rule creation, conflict resolution, and structured knowledge output [25]. Practical systems such as Hrycej’s problem-specific program generator [19] and Bisson’s KBG framework [4], along with evaluations by Bryant and Krause [7], offer reusable techniques for automating rule-based reasoning workflows — including rational and lexicographic closure computation.

As defeasible reasoning systems increase in complexity, the role of intuitive *user interfaces* (UIs) and debugging tools becomes increasingly important. Graphical tools such as the SILK GUI, developed by Grosof et al. [18], allow users to input knowledge, explore rule justifications, and debug defeasible reasoning by visualizing justification trees and rule defeats. Debugging frameworks, such as the one proposed by Coetzer and Britz [12], support conflict identification and correction in classical ontologies. Earlier systems like DEIMOS [1] demonstrate how human-readable outputs and step-by-step derivations can foster greater trust and understandability in automated reasoning.

This review examines the development from rational closure to lexicographic closure, along with the supporting infrastructure of automated knowledge base generation and user-facing debugging tools. The goal is to inform the design of efficient algorithms and the construction of scalable, explainable defeasible reasoning systems.

2 PROPOSITIONAL LOGIC

Propositional logic is a fundamental framework due to its well-defined structure and broad applicability in intelligent systems [3]. Although it lacks mechanisms for handling exceptions, its clear syntax and semantics provide a basis for defeasible reasoning models such as the KLM framework and rational closure [6]. Understanding these factors pertaining to propositional logic is essential for developing reasoning systems that balance formal precision with real-world flexibility.

2.1 Syntax

Propositional logic represents statements as *atoms* (e.g., P: "it is raining"), combined using *Boolean operators* $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$, representing negation, disjunction, conjunction, implication, and equivalence. Formulas are built by combining atoms and operators under formal syntax rules [3]. For example, $(P \wedge Q) \rightarrow (\neg R \vee P)$ expresses a structured logical relationship.

2.2 Object-Level Semantics

2.2.1 Valuations. A valuation $u : \mathcal{P} \rightarrow \{T, F\}$ assigns truth values to atoms [20]. Given $\mathcal{P} = \{p, q, r\}$, an example valuation is $u(p) = T, u(q) = F, u(r) = F$, written as $\mathrm{p}\overline{\mathrm{qr}}$.

2.2.2 Atoms. Atoms derive meaning from context [20](e.g., p: "cars use petrol," q: "electric cars do not use petrol").

2.2.3 Boolean Operators. Boolean operators determine the truth of complex formulas via truth tables [3], ensuring consistency in logical evaluations.

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

Table 1: Truth table for binary Boolean operators

For the unary operator $\neg$, we define $\neg p$ as false if p is true and vice versa.

2.3 Meta-Level Semantics

2.3.1 Satisfiability. A valuation u satisfies an atom p, written $u \Vdash p$, if $u(p) = T$ [3]. This extends to formulas: $u \Vdash A$ if A evaluates to T under u. A knowledge base $\mathcal{K}$ is satisfiable if at least one valuation satisfies all its formulas. Otherwise, it is unsatisfiable.

2.3.2 Entailment. A formula A is *entailed* by $\mathcal{K}$, written $\mathcal{K} \models A$, if all models of $\mathcal{K}$ are also models of A, meaning $Mod(\mathcal{K}) \subseteq Mod(A)$ [3].

2.4 Object vs Meta Levels

The *object level* involves formulas and logical derivations, such as deriving $p \wedge r$ from $p \wedge q$ and $q \rightarrow r$. The *meta level* examines logical properties, including satisfaction and entailment, with symbols $\models$ and $\Vdash$ representing relationships between formulas and valuations [3]. This distinction clarifies the roles of syntax, truth values, and reasoning in propositional logic.

3 DEFEASIBLE REASONING

We will now use classical propositional logic to solve a real-world problem that was mentioned previously. We consider a knowledge base $\mathcal{K} = \{e \rightarrow c, c \rightarrow p, e \rightarrow \neg p\}$ that contains the knowledge that an electric car is a car, cars use petrol, and an electric car does not use petrol, respectively. Since $e \rightarrow c$ and $c \rightarrow p$, we conclude, using the basic propositional logical rules, that $e \rightarrow p$. However, the knowledge base says $e \rightarrow \neg p$. This leads us to the conclusion that e is not true, meaning electric cars do not exist. However, we know that this is not the case from real-world knowledge. As demonstrated, propositional logic does not handle exceptions; therefore, we introduce **defeasible reasoning** that deals with these real-world scenarios.

3.1 The KLM Framework

The KLM framework, developed by Kraus, Lehmann, and Magidor [21], is a key foundation for defeasible reasoning. It introduces the concept of *typicality* through a meta-level consequence relation, $\mid\sim$, which expresses statements like "typically, if α then β" as $\alpha \mid\sim \beta$. The framework is built on six core postulates that balance precision with intuitive common sense reasoning, ensuring cautious conclusions without overly strong assumptions [20].

KLM defines *preferential consequence relations*, modeled using *preferential interpretations*, where the interpretations are classified by typicality [28]. Inferences are drawn from the most typical cases. However, to handle conflicts between defaults, Lehmann and Magidor [24] introduced a seventh postulate, leading to *rational consequence relations*. These refine preferential entailment using *ranked interpretations*, establishing a more structured hierarchy of typicality.

3.1.1 Ranked Interpretations. Ranked interpretations are central to rational consequence relations. They assign each valuation of a knowledge base a non-negative integer rank or ∞, where lower ranks represent more typical valuations and higher ranks indicate less typical scenarios [9]. This ranking helps defeasible reasoning systems determine which scenarios best represent the defaults. For instance, given atoms p, q, r, valuations that fully align with the most general defaults may be ranked 0, while those that partially comply are assigned higher ranks, and impossible valuations are given ∞. This provides a structured, mathematically precise way to handle exceptions.

For example, given the atoms $\mathcal{P} := \{p, q, r\}$, one possible ranked interpretation is given below:

∞	$\mathrm{pq}\overline{\mathrm{r}}$
2	pqr $\mathrm{p}\overline{\mathrm{q}}\mathrm{r}$ $\overline{\mathrm{p}}\mathrm{qr}$
1	$\overline{\mathrm{pq}}\mathrm{r}$ $\mathrm{p}\overline{\mathrm{qr}}$
0	$\mathrm{pq}\overline{\mathrm{r}}$ $\overline{\mathrm{pqr}}$

Figure 1: A possible ranked interpretation of $\mathcal{P}$

This ranked interpretation would imply that the valuation $u := \overline{\mathrm{pq}}\mathrm{r}$ is more likely than the valuation $v := \mathrm{pqr}$, since $\mathcal{R}(u) < \mathcal{R}(v)$. We can also infer that the valuation $\mathrm{pq}\overline{\mathrm{r}}$ is impossible.

3.1.2 Defeasible Implication. In KLM-based frameworks, the concept of a *defeasible implication* is introduced, where $\alpha \mid\sim \beta$ signifies that in the most typical scenarios where α holds, β is also expected to hold [20]. These implications form the basis of a *defeasible knowledge base*, which consists of conditional rules that guide inference. The satisfiability of a defeasible implication is determined by verifying that all minimal α-worlds — those with the lowest rank — also satisfy β. This structure allows reasoning systems to make assumptions that hold under normal circumstances while allowing exceptions to emerge naturally when more specific or conflicting information is introduced.

3.1.3 Satisfiability. In the context of ranked interpretations, satisfiability of a defeasible implication depends on its truth in the most typical or *minimal* worlds — those with the lowest assigned rank [20]. Formally, given a ranked interpretation $\mathcal{R}$ and a formula $\alpha \in \mathcal{L}$, a valuation u is considered minimal in $[\![\alpha]\!]^{\mathcal{R}}$ if no other valuation $v \in [\![\alpha]\!]^{\mathcal{R}}$ has a lower rank than u. For example, using the ranked interpretation provided earlier and setting $\alpha := p$, the minimal α-world is identified as the valuation $\mathrm{p}\overline{\mathrm{qr}}$, having the lowest rank of 1 compared to others.

A ranked interpretation $\mathcal{R}$ is said to satisfy a defeasible implication $\alpha \mid\sim \beta$ if, for every minimal world where α holds, β also holds. This condition ensures that defaults are upheld in the most typical circumstances. Extending this concept, a ranked interpretation satisfies an entire defeasible knowledge base $\mathcal{K}$ if it satisfies every defeasible implication and classical formula within it [20]. To incorporate classical formulas into this framework, each formula α is represented as a defeasible implication of the form $\neg\alpha \mid\sim \bot$, where $\bot$ is a contradiction. This representation guarantees that classical formulas are considered true in all finitely ranked worlds, since no minimal world can satisfy a contradiction [8]. For instance, consider the classical formula r (meaning "the engine runs"). This can be expressed as $\neg r \mid\sim \bot$, meaning that in any minimal world where $\neg r$ is true, a contradiction would have to hold, which is impossible. Therefore, all minimal worlds must satisfy r, enforcing classical truth within the defeasible framework. This representation ensures consistency between classical reasoning and typicality-based defeasible reasoning.

3.1.4 Entailment. Kraus, Lehmann, and Magidor [21] initially proposed six rationality postulates that any reasonable meta-level consequence relation, denoted by $\mid\sim$, should satisfy. These postulates aim to capture how default reasoning should behave in the presence of general rules and exceptions. Later, Lehmann and Magidor [24] introduced a seventh postulate, Rational Monotony (RM), to further refine the reasoning process and allow for more cautious inference patterns. Casini et al. [9] reformulated these conditions to define *defeasible entailment*, denoted by $\approx$, which characterizes how conclusions can be carefully derived from a defeasible knowledge base at the meta-level.

The seven rationality postulates are defined as follows:

(LLE) $\dfrac{\mathcal{K} \models\!\!\approx \alpha \leftrightarrow \beta, \mathcal{K} \models\!\!\approx \alpha \mid\sim \gamma}{\mathcal{K} \models\!\!\approx \beta \mid\sim \gamma}$ **(Or)** $\dfrac{\mathcal{K} \models\!\!\approx \alpha \mid\sim \gamma, \mathcal{K} \models\!\!\approx \beta \mid\sim \gamma}{\mathcal{K} \models\!\!\approx \alpha \vee \beta \mid\sim \gamma}$

(RW) $\dfrac{\mathcal{K} \models\!\!\approx \alpha \rightarrow \beta, \mathcal{K} \models\!\!\approx \gamma \mid\sim \alpha}{\mathcal{K} \models\!\!\approx \gamma \mid\sim \beta}$ **(CM)** $\dfrac{\mathcal{K} \models\!\!\approx \alpha \mid\sim \gamma, \mathcal{K} \models\!\!\approx \alpha \mid\sim \beta}{\mathcal{K} \models\!\!\approx \alpha \wedge \beta \mid\sim \gamma}$

(Ref) $\mathcal{K} \models\!\!\approx \alpha \mid\sim \alpha$ **(RM)** $\dfrac{\mathcal{K} \models\!\!\approx \alpha \mid\sim \gamma, \mathcal{K} \not\models\!\!\approx \alpha \not\mid\sim \beta}{\mathcal{K} \models\!\!\approx \alpha \wedge \beta \mid\sim \gamma}$

(And $\dfrac{\mathcal{K} \models\!\!\approx \alpha \mid\sim \beta, \mathcal{K} \models\!\!\approx \alpha \mid\sim \gamma}{\mathcal{K} \models\!\!\approx \alpha \mid\sim \beta \wedge \gamma}$

A consequence relation satisfying the standard postulates, including Rational Monotony (RM), is considered *LM-rational.* KLM preferential entailment meets all but RM [21], relying on *preferential interpretations* where inferences follow from the most typical worlds. Lehmann and Magidor [24] introduced *ranked entailment*, defined via ranked interpretations, as a refinement that satisfies RM. Thus, while preferential entailment is not LM-rational, ranked entailment is [9].

One key limitation of classical entailment is its monotonicity: once a conclusion is derived, it cannot be retracted in light of new information. While this simplifies inference, it does not reflect real-world reasoning, where exceptions frequently arise. Ranked entailment, although nonmonotonic, provides a structured foundation for more sophisticated forms of defeasible reasoning, such as rational closure and lexicographic closure [8].

To illustrate Rational Monotony with an example, we consider a knowledge base with the defaults "cars typically use petrol" ($c \mid\sim p$) and "electric cars are cars that typically do not use petrol" ($e \mid\sim \neg p$, with $e \rightarrow c$). Initially, if we know an object is a car (c), we will conclude that it uses petrol. However, if additional information is introduced that this car is electric (e), and we have no reason to believe that electric cars use petrol, the Rational Monotony postulate allows us to refine our inference. Instead of sticking with the general rule, we revise our conclusion and accept that this specific car does not use petrol.

3.2 Extending the KLM Framework

3.2.1 Ranked Entailment. We need to define ranked entailment in order to extend defeasible entailment within the KLM framework [9]: A defeasible implication $\alpha \mid\sim \beta$ is considered to be rank-entailment of a knowledge base $\mathcal{K}$, written as $\mathcal{K} \models\!\!\approx_R \alpha \mid\sim \beta$, if it holds in every *ranked interpretation* that serves as a model of $\mathcal{K}$ by satisfying all its formulas. Unlike preferential entailment, ranked entailment satisfies Rational Monotony and therefore meets the criteria for LM-rationality [21].

3.2.2 Basic Defeasible Entailment. Casini et al. [8] expanded on the KLM framework by introducing a broader category of defeasible entailment relations, referred to as *basic defeasible entailment relations.* These relations are not only LM-rational but also adhere to two additional key conditions:

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

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

The *Inclusion* property ensures that every defeasible implication present in the knowledge base $\mathcal{K}$ is also derivable from it. *Classic Preservation*, on the other hand, guarantees that contradictions - statements (of the form $\alpha \mid\sim \bot$) that are entailed in the defeasible sense correspond precisely to those rank entailments derived from $\mathcal{K}$. This highlights how rank entailment functions as the structured semantic foundation on which these more expressive forms of defeasible reasoning are built.

3.2.3 Rational Defeasible Entailment. Basic defeasible entailment, while useful, is often considered too permissive because it does not always yield intuitively reasonable inferences [21, 24]. It allows conclusions that may not align with how rational agents typically reason when dealing with conflicting defaults. To address these limitations, rational defeasible entailment was introduced as a refinement that ensures more controlled reasoning by incorporating stricter constraints on inference [9, 17]. Rational defeasible entailment builds upon basic defeasible entailment but enforces additional properties that prevent overly permissive conclusions, ensuring that only well-justified inferences are drawn [26]. This refinement is particularly important in cases where multiple defaults interact in complex ways, requiring a systematic method for resolving conflicts while maintaining consistency [10, 23].

A key requirement for any reasonable defeasible entailment relation is that it should, at a minimum, include all inferences drawn by a well-defined baseline system [16]. This ensures that defeasible reasoning does not become arbitrary or unpredictable. Towards this goal, rational closure serves as this baseline, providing a structured approach to defeasible entailment that maintains logical coherence while ranking default rules according to their specificity [8]. By introducing an explicit ranking mechanism, rational closure refines defeasible entailment by systematically resolving conflicts between defaults, ensuring that conclusions are both reasonable and justifiable [11].

3.3 Rational Closure

Rational closure (RC) can be defined semantically through a minimal ranked model or syntactically via base ranks [8]. The minimal ranked model defines RC as the unique ranked interpretation that satisfies the knowledge base while minimizing exceptionality across formulas. On the other hand, the syntactic construction of rational closure assigns ranks to formulas based on exceptionality, capturing their relative typicality within the knowledge base. These models collectively capture rational closure's approach to defeasible reasoning by balancing internal consistency and comparative plausibility across models.

3.3.1 Minimal Ranked Model. The concept of a *minimal ranked model* plays a central role in defining rational closure. A ranked interpretation $\mathcal{R}$ assigns each valuation (or world) a rank in $\mathbb{N}\cup\{\infty\}$, where lower ranks correspond to more typical scenarios, and ∞ indicates impossible worlds. A ranked interpretation is considered a *model* of a knowledge base $\mathcal{K}$ if it satisfies all the defeasible implications and classical formulas in $\mathcal{K}$. Among all such ranked models, the *minimal ranked model* is the one that assigns the lowest possible ranks to valuations while still respecting the constraints of $\mathcal{K}$ [8, 17].

The minimal ranked model of a knowledge base $\mathcal{K}$, denoted by $\mathcal{R}^{RC}_{\mathcal{K}}$, is defined as the ranked model that is pointwise minimal among all ranked models of $\mathcal{K}$. Formally:

$$\forall \mathcal{R}' \text{ ranked model of } \mathcal{K}, \quad \mathcal{R}^{RC}_{\mathcal{K}}(u) \leq \mathcal{R}'(u) \text{ for every valuation } u.$$

This minimal model provides the most conservative representation of defaults; it only increases the rank of a valuation when forced to by inconsistencies or conflicts with more typical defaults. For example, if two defaults $c \mid\!\sim p$ (cars typically use petrol) and $e \mid\!\sim \neg p$ (electric cars typically do not use petrol) conflict, valuations where both e and p hold are assigned higher ranks to indicate that they are less typical. Rational closure uses this minimal ranked model to determine which defeasible conclusions hold [8, 17, 20].

Given the query $e \mid\!\sim p$ (whether electric cars typically use petrol), we check all minimal e-worlds in $\mathcal{R}^{RC}_{\mathcal{K}}$. The minimal e-worlds are those at rank 1, such as $ec\neg p$, which do not satisfy p. Therefore, rational closure does not entail $e \mid\!\sim p$, indicating that electric cars do not typically use petrol.

∞	ecp
2	ecp (redundant world for illustration if needed)
1	$ec\neg p$, $c\neg p\neg e$
0	$cp\neg e$, $\neg c\neg p\neg e$, $\neg cp\neg e$

Figure 2: A minimal ranked model for the knowledge base $\mathcal{K}$ with c = car, p = uses petrol, and e = electric.

3.3.2 Base Ranks. The concept of *base ranks* provides a structured way to determine how exceptional a formula is relative to a knowledge base $\mathcal{K}$. Recall that rank entailment is denoted by $\models\!\!\approx_R$. A formula $\alpha \in \mathcal{L}$ is said to be *exceptional* in $\mathcal{K}$ if the most typical valuations contradict it, formally defined as:

$$\mathcal{K} \models\!\!\approx_R \top \mid\!\sim \neg\alpha$$

In other words, α is exceptional if it is false in the most typical worlds across all ranked models of $\mathcal{K}$ [9]. Another way to detect exceptional formulas is through the *materialisation* of a knowledge base, denoted by $\overrightarrow{\mathcal{K}}$, which translates each defeasible implication $\alpha \mid\!\sim \beta$ into a classical formula $\alpha \rightarrow \beta$:

$$\overrightarrow{\mathcal{K}} := \{\alpha \rightarrow \beta \mid \alpha \mid\!\sim \beta \in \mathcal{K}\}$$

A formula α is exceptional if $\overrightarrow{\mathcal{K}} \models \neg\alpha$ [20].

We define a sequence of knowledge bases to iteratively capture exceptional defaults:

$$\mathcal{E}^{\mathcal{K}}_0 := \mathcal{K}$$

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

$$\mathcal{E}^{\mathcal{K}}_\infty := \mathcal{E}^{\mathcal{K}}_n \text{ where } n \text{ is the smallest index for which } \mathcal{E}^{\mathcal{K}}_n = \mathcal{E}^{\mathcal{K}}_{n+1}$$

Here, $\varepsilon(\mathcal{E}^{\mathcal{K}}_{i-1})$ refers to the set of defaults whose antecedents are exceptional with respect to $\mathcal{E}^{\mathcal{K}}_{i-1}$ [9]. This process continues until no further changes occur, guaranteeing termination since $\mathcal{K}$ is finite.

The *base rank* of a formula α, denoted $br_{\mathcal{K}}(\alpha)$, is the least integer r such that α is no longer exceptional in $\mathcal{E}^{\mathcal{K}}_r$:

$$br_{\mathcal{K}}(\alpha) := \min\{r \mid \overrightarrow{\mathcal{E}^{\mathcal{K}}_r} \not\models \neg\alpha\}$$

For defeasible implications, the base rank is assigned to the antecedent:

$$br_{\mathcal{K}}(\alpha \mid\!\sim \beta) := br_{\mathcal{K}}(\alpha)$$

Rational closure can then be defined using base ranks. A default $\alpha \mid\!\sim \beta$ is entailed in rational closure if and only if:

$$\mathcal{K} \models\!\!\approx_{RC} \alpha \mid\!\sim \beta \iff br_{\mathcal{K}}(\alpha) < br_{\mathcal{K}}(\alpha \wedge \neg\beta) \text{ or } br_{\mathcal{K}}(\alpha) = \infty$$

Additionally, there is a direct relationship between base ranks and the minimal ranked model $\mathcal{R}^{RC}_{\mathcal{K}}$:

$$br_{\mathcal{K}}(\alpha) = \min\{i \mid \exists v \in Mod(\alpha) \text{ with } \mathcal{R}^{RC}_{\mathcal{K}}(v) = i\}$$

This shows that the base rank of α corresponds to the lowest rank of valuations satisfying α in the minimal ranked model [17].

Example: Consider the following knowledge base:

$$\mathcal{K} = \{c \mathrel{|\!\sim} p, e \mathrel{|\!\sim} \neg p, e \rightarrow c\}$$

Here, c represents *car*, p represents *uses petrol*, and e represents *electric car*. We start by computing the sequence of exceptional knowledge bases:

$$\mathcal{E}_0^{\mathcal{K}} = \{c \mathrel{|\!\sim} p, e \mathrel{|\!\sim} \neg p, e \rightarrow c\}$$

Check if e is exceptional by seeing if $\overrightarrow{\mathcal{E}_0^{\mathcal{K}}} \models \neg e$. Initially, this holds because e contradicts the stronger typicality that cars use petrol, but e does not. Thus:

$\mathcal{E}_1^{\mathcal{K}} = \{e \mathrel{|\!\sim} \neg p\}$ (only keeping defaults with exceptional antecedents)

On checking again, e no longer appears exceptional with respect to $\mathcal{E}_1^{\mathcal{K}}$, so the base rank of e is 1. Therefore:

$$br_{\mathcal{K}}(e \mathrel{|\!\sim} \neg p) = br_{\mathcal{K}}(e) = 1$$

This corresponds to the rank of the minimal e-worlds in $\mathcal{R}_{\mathcal{K}}^{RC}$, demonstrating how the system determines the typical conditions under which electric cars do not use petrol.

3.3.3 The Rational Closure Algorithm. With the concept of base ranks defined, we can now describe the process of computing the rational closure of a knowledge base $\mathcal{K}$. The rational closure algorithm consists of two main steps: `BaseRank` and `RationalClosure`. In the `BaseRank` step, the materialisation of the knowledge base, $\overrightarrow{\mathcal{K}}$, is taken, and its rules are grouped according to their base rank. This produces sets of rules R_i, where each set contains all statements whose antecedents have a base rank of i:

$$R_i := \{\alpha \rightarrow \beta \mid \alpha \mathrel{|\!\sim} \beta \in \mathcal{K}, br_{\mathcal{K}}(\alpha) = i\}$$

[9].

Once these partitions are established, the `RationalClosure` step uses them to determine whether a query $\alpha \mathrel{|\!\sim} \beta$ belongs to the rational closure of $\mathcal{K}$. This is done by checking if the query is exceptional in all partitions ($R_0 \cup \cdots \cup R_\infty$). If it is, the algorithm successively removes partitions, starting from the lowest rank R_0, then R_1, and so on, until it finds a partition where the query is no longer exceptional or only R_∞ remains. At that point, the algorithm checks whether the materialised version of the query, $\alpha \rightarrow \beta$, logically follows from the remaining partitions. If it does, the algorithm returns true; if not, it returns false [9]. Freund proved that this process is sound and complete: the algorithm returns true if and only if $\mathcal{K} \mathrel{|\!\approx}_{RC} \alpha \mathrel{|\!\sim} \beta$ [16].

Let us revisit our car–petrol knowledge base to illustrate this process. The `BaseRank` step produces the following partitions: Now

R_∞	$e \rightarrow c$
R_1	$e \mathrel{|\!\sim} \neg p$
R_0	$c \mathrel{|\!\sim} p$

Figure 3: Partitions of $\overrightarrow{\mathcal{K}}$ produced by `BaseRank`

consider the query $e \mathrel{|\!\sim} p$ (whether electric cars typically use petrol). We first check if this query is exceptional in $R_0 \cup R_1 \cup R_\infty$. Since electric cars are an exception to the default that cars use petrol, the query is indeed exceptional. The algorithm removes R_0 and re-checks with $R_1 \cup R_\infty$. In this reduced set, the query is no longer exceptional, so the algorithm checks whether $R_1 \cup R_\infty$ entails $e \rightarrow p$. Since it does not, the algorithm returns false, confirming that rational closure does not conclude that electric cars typically use petrol — aligning with our semantic expectations.

This example highlights the cautious, or *prototypical*, nature of rational closure. It assigns properties that hold in the most typical cases and avoids assumptions about atypical scenarios. For example, it does not attribute typical car properties, like using petrol, to electric cars when these conflict with more specific knowledge. This contrasts with *lexicographic closure*, which is more *presumptive* and tends to assign defaults unless explicitly contradicted [20]. This difference will become clearer when we later evaluate the same query using lexicographic closure.

3.4 Lexicographic Closure

Lexicographic closure, introduced by Lehmann [23], extends default reasoning [27] with a lexicographically ordered set of defaults. Like rational closure, it can be defined using a *lexicographic ranked model* [9]. Two types of lexicographic closure exist: one based on a simple linear ranking [23], and another with more complex ordering [9]. Both types can be defined through a lexicographic rank function, similar to the base rank function in rational closure [9], which prioritizes higher-ranked defaults in reasoning.

3.4.1 Lexicographic Ranked Model in [23]. The order of a defeasible knowledge base K is defined as the maximum finite base rank of any formula in K, excluding those with rank ∞ [20]. For example, suppose "cars use petrol" has a higher base rank than "electric vehicles don't use petrol"; the order of K would be the higher of these finite ranks. For a knowledge base of order k, each subset $D \subseteq K$ is assigned a tuple $n_D = \langle n_0, n_1, \ldots, n_k \rangle$, where n_0 counts defaults with base rank ∞, and n_i counts those with base rank $k - i$ [20]. Subsets are ordered lexicographically based on these tuples, with $D_1 \prec_S D_2$ if n_{D_1} is lexicographically smaller than n_{D_2}. Valuations are then ordered using $\preceq_{LC}$ based on the lex order of the tuples of violated defaults. Using this, the lexicographic ranked model R_K^{LC} is constructed. A default $\alpha \mathrel{|\!\sim} \beta$ is entailed in lexicographic closure, written $K \models_{LC} \alpha \mathrel{|\!\sim} \beta$, if it holds in R_K^{LC}. In our example, this would mean that "electric vehicles don't use petrol" is entailed because more typical defaults are prioritized lexicographically.

3.4.2 Lexicographic Ranked Model in [9]. Given a knowledge base $\mathcal{K}$, we define a function $C^{\mathcal{K}} : \mathcal{U} \rightarrow \mathbb{N}$, where $C^{\mathcal{K}}(v) = |\{\alpha \mathrel{|\!\sim} \beta \in \mathcal{K} \mid v \models \alpha \rightarrow \beta\}|$ [9]. This maps each valuation to the number of defeasible implications whose materialisations it satisfies. We then impose an order $\preceq_{LC}$ on valuations $u, v \in \mathcal{U}$, where $u \preceq_{LC} v$ if and only if:

- $\mathcal{R}_{\mathcal{K}}^{RC}(v) = \infty$, or
- $\mathcal{R}_{\mathcal{K}}^{RC}(u) < \mathcal{R}_{\mathcal{K}}^{RC}(v)$, or
- $\mathcal{R}_{\mathcal{K}}^{RC}(u) = \mathcal{R}_{\mathcal{K}}^{RC}(v)$ and $C^{\mathcal{K}}(u) \geq C^{\mathcal{K}}(v)$

This ordering produces the ranked interpretation $\mathcal{R}_{\mathcal{K}}^{LC}$, which defines the entailment relation $\mathrel{|\!\approx}_{LC}$. A defeasible implication $\alpha \mathrel{|\!\sim} \beta$ belongs to the lexicographic closure of $\mathcal{K}$, denoted $\mathcal{K} \mathrel{|\!\approx}_{LC} \alpha \mathrel{|\!\sim} \beta$, if and only if $\mathcal{R}_{\mathcal{K}}^{LC} \models \alpha \mathrel{|\!\sim} \beta$ [9].

Both Lehmann's [23] and Casini et al.'s [9] approaches use valuation ranking to extend rational closure, but differ in refinement criteria. Lehmann's model ranks valuations by the severity of violated defaults, evaluating how *serious* a set of violations is via tuples assigned to subsets of $\mathcal{K}$; violations of higher-ranked defaults are considered more severe, and valuations with lower tuples are preferred [20].

Casini et al. refine this by first ranking valuations according to $\mathcal{R}^{RC}_{\mathcal{K}}$ and then, within each rank, ordering them by the number of satisfied defeasible implications. Valuations that satisfy more defaults are placed lower in the order and considered more typical [9]. While both methods extend rational closure, they are not equivalent; a detailed comparison is provided in [13].

Example: Cars and Petrol. Consider the following knowledge base:

(1) $c \rightarrow p$ (Cars use petrol)
(2) $e \rightarrow c$ (Electric cars are cars)
(3) $e \nrightarrow p$ (Electric cars typically do not use petrol)

We evaluate possible worlds:

- w_1: c, p, e (Electric cars use petrol, violating $e \nrightarrow p$)
- w_2: $c, p, e, \neg p$ (Electric cars do not use petrol, contradicting $c \rightarrow p$)
- w_3: $c, p, \neg e$ (No electric cars exist)

Lexicographic closure ranks valuations by minimizing violations of higher-ranked defaults and, within equal ranks, favoring those that satisfy more defeasible statements. This prioritizes w_2 over w_1, preserving the rule that electric cars typically do not use petrol.

Lehmann's lexicographic closure [23] focuses on reducing the severity of violations, ranking valuations according to the most serious defaults they violate. In contrast, Casini et al.'s approach [9] refines this ranking by using $\mathcal{R}^{RC}_{\mathcal{K}}$ and favoring valuations that satisfy more defaults rather than focusing solely on violations.

In essence, Lehmann's model emphasizes minimizing the severity of violations, while Casini et al.'s model emphasizes maximizing satisfied defaults within each rank. These differences yield distinct interpretations of typicality and conflict resolution in nonmonotonic reasoning [13].

3.4.3 Lexicographic Rank. Casini et al. [9] demonstrated that lexicographic closure can be characterized using a ranked model that refines the rational closure model $\mathcal{R}^{RC}_{\mathcal{K}}$. Building on this, they define the rank of a formula as the minimum rank of all valuations in the lexicographic model that satisfy it. Letting $\mathcal{R}^{LC}_{\mathcal{K}}(v)$ denote the rank of valuation v, we define the lexicographic rank of a formula α as $r^{LC}_{\mathcal{K}}(\alpha) := \min\{\mathcal{R}^{LC}_{\mathcal{K}}(v) \mid v \models \alpha\}$. Using this rank function, lexicographic closure can be defined as follows: a defeasible implication $\alpha \mathrel{|\!\sim} \beta$ is entailed in the lexicographic closure of $\mathcal{K}$, written $\mathcal{K} \mathrel{|\!\approx}_{LC} \alpha \mathrel{|\!\sim} \beta$, if and only if $r^{LC}_{\mathcal{K}}(\alpha) < r^{LC}_{\mathcal{K}}(\alpha \wedge \neg\beta)$ or $r^{LC}_{\mathcal{K}}(\alpha) = \infty$ [9].

3.4.4 Algorithm for LC Entailment Queries. The lexicographic closure entailment algorithm, based on Casini et al. [8, 9], determines whether a defeasible implication $\alpha \mathrel{|\!\sim} \beta$ belongs to the lexicographic closure of a knowledge base $\mathcal{K}$. It uses the minimal ranked model from rational closure ($\mathcal{R}^{RC}_{\mathcal{K}}$), and refines it by comparing the number of satisfied materialised defaults per valuation, producing a lexicographic ordering over valuations.

Algorithm 1 Lexicographic Closure Entailment Check

Input: A defeasible knowledge base $\mathcal{K}$, query $\alpha \mathrel{|\!\sim} \beta$
Output: true if $\mathcal{K} \mathrel{|\!\approx}_{LC} \alpha \mathrel{|\!\sim} \beta$, false otherwise

Compute rational closure ranks and $\mathcal{R}^{RC}_{\mathcal{K}}(v)$ for each valuation v

Compute $C^{\mathcal{K}}(v) = |\{\alpha \mathrel{|\!\sim} \beta \in \mathcal{K} \mid v \models \alpha \rightarrow \beta\}|$ for each valuation v Order valuations u, v using $\preceq_{LC}$:

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

Construct the lexicographic ranked model $\mathcal{R}^{LC}_{\mathcal{K}}$ Compute $r^{LC}_{\mathcal{K}}(\alpha) = \min\{\mathcal{R}^{LC}_{\mathcal{K}}(v) \mid v \models \alpha\}$ and $r^{LC}_{\mathcal{K}}(\alpha \wedge \neg\beta)$ **if** $r^{LC}_{\mathcal{K}}(\alpha) < r^{LC}_{\mathcal{K}}(\alpha \wedge \neg\beta)$ *or* $r^{LC}_{\mathcal{K}}(\alpha) = \infty$ **then**
  return *true*
else
  return *false*

3.4.5 Example: Cars and Petrol. Consider the following defeasible knowledge base:

(1) $c \mathrel{|\!\sim} p$ (Cars typically use petrol)
(2) $e \mathrel{|\!\sim} c$ (Electric cars are typically cars)
(3) $e \mathrel{|\!\sim} \neg p$ (Electric cars typically do not use petrol)

We evaluate all truth assignments (valuations) to the propositional variables c, e, and p, and rank them based on how well they satisfy the defaults. The minimal ranked model from rational closure $\mathcal{R}^{RC}_{\mathcal{K}}$ is computed first, then refined into the lexicographic ranked model $R^{LC}_{\mathcal{K}}$ based on how many defaults each valuation satisfies.

Rank	Valuation	Notes
0	$c, e, \neg p$	Satisfies all defaults
1	$c, p, \neg e$	Vacuously satisfies $e \mathrel{|\!\sim} \neg p$; satisfies $c \mathrel{|\!\sim} p$
2	c, e, p	Violates $e \mathrel{|\!\sim} \neg p$
∞	(none)	No valuation contradicts all defaults

Table 2: The lexicographic ranked model $R^{LC}_{\mathcal{K}}$ for the car-petrol knowledge base

To determine whether $\mathcal{K} \mathrel{|\!\approx}_{LC} e \mathrel{|\!\sim} \neg p$, we compute:

- $r^{LC}_{\mathcal{K}}(e) = 0$, from valuation $c, e, \neg p$
- $r^{LC}_{\mathcal{K}}(e \wedge p) = 2$, from valuation c, e, p

Since $r^{LC}_{\mathcal{K}}(e) < r^{LC}_{\mathcal{K}}(e \wedge p)$, the entailment $\mathcal{K} \mathrel{|\!\approx}_{LC} e \mathrel{|\!\sim} \neg p$ holds. This confirms that, under lexicographic closure, electric cars are typically not considered to use petrol, as the system prioritizes the most specific and strongly supported defaults.

3.4.6 Interpretation. This procedure ensures that higher-ranked valuations (those with more severe violations or fewer satisfied defaults) are less preferred. The lexicographic ordering allows us to reason about typicality and draw conclusions in a way that prioritizes both the severity of rule violations and the number of satisfied defaults, providing a refined extension of rational closure.

4 CURRENT WORK

4.1 Improving Algorithm Efficiency

4.1.1 Rational closure algorithm. While the rational closure algorithm described by Casini et al. [9] and Freund [16] is sound and complete, its iterative process—constructing exceptional subsets $\mathcal{K}_0, \mathcal{K}_1, \ldots$ until a query is no longer exceptional—can be computationally demanding for large knowledge bases. Efficiency can be improved by precomputing *exceptional formulas* and caching their base ranks to avoid redundant entailment checks [17]. Instead of linearly scanning through all rank layers, a binary search strategy can reduce the number of entailment checks from $O(n)$ to $O(\log n)$, where n is the maximum base rank. Moreover, Giordano et al. [17] demonstrate that rational closure conditions can be evaluated using minimal ranked models, enabling direct rank lookup without repeated exceptionality testing. Finally, representing defeasible implications via materialisation and exploiting compact formula forms, such as Horn clauses [3], can further reduce classical entailment overhead. Collectively, these optimizations—binary search, model-based rank caching, and efficient formula representations—can lower reasoning complexity from $O(mn)$ to $O(m \log n)$, where m is the number of defeasible rules in $\mathcal{K}$.

4.1.2 Lexicographic algorithm. While the lexicographic closure algorithm described by Casini et al. [8, 9] is both sound and complete, it can become computationally expensive due to the need to refine and compare valuations across multiple ranks. Efficiency can be improved by precomputing the minimal ranked model and caching lexicographic ranks, turning repeated rank comparisons into constant-time lookups [17]. Furthermore, instead of scanning all rank layers linearly, binary search techniques over ranks can reduce entailment checks from $O(n)$ to $O(\log n)$, where n is the maximum base rank. Finally, applying materialisation reductions and representing classical formulas in compressed Horn form [3, 5] can further reduce classical entailment overhead. Together, these optimizations—rank caching, binary search, and Horn-formula compression—make lexicographic closure reasoning more scalable for large knowledge bases.

5 KNOWLEDGE BASE GENERATOR

Knowledge-based generators (KBGs) are systems designed to automatically produce specialized knowledge bases or applications from higher-level specifications. They aim to reduce manual effort in developing complex reasoning systems by automating repetitive structures and logic patterns [25]. This is particularly important in projects like ours, where the complexity of constructing conditional knowledge bases and computing rational or lexicographic closure can become overwhelming. KBGs allow for rapid iteration and consistent generation of rule structures, supporting tasks such as automated rule inference, structured output generation, and conditional entailment verification. In the context of this project, a knowledge-based generator could dynamically create test knowledge bases, reducing human error and speeding up the process of validating and optimizing closure algorithms.

5.0.1 Existing Algorithms for Knowledge-Based Systems. Several approaches and algorithms for KBGs have been proposed in the literature. Early contributions such as Hrycej's problem-specific program generator [19] demonstrated the feasibility of creating rule-driven software from abstract specifications. Kukich [22] introduced a report generator that translated structured input into readable summaries, laying the groundwork for knowledge-to-text systems. Bisson's KBG system [4] further refined this by enabling the automatic construction of knowledge bases with consistent rule application. More recently, Bryant and Krause [7] provided a comprehensive review of defeasible reasoning implementations, highlighting the need for efficient, reusable algorithmic frameworks.

These systems share common algorithmic principles: rule materialization, conflict detection, prioritization structures, and output verification. Such elements align closely with the algorithms employed in this project, including base rank partitioning and rational closure computation. The potential integration of KBGs into defeasible reasoning workflows could not only automate test case creation but also ensure that extensions like lexicographic closure are applied in a systematic and scalable manner. As we work to optimize reasoning algorithms, insights from these existing generators offer practical methods for automation and refinement.

6 USER INTERFACES AND DEBUGGING TOOLS

As the complexity of defeasible reasoning systems grows, the need for intuitive user interfaces (UIs) and robust debugging tools becomes increasingly important. These tools are essential for making complex knowledge bases understandable and for helping users trace and correct inference errors. Grosof et al. [18] introduced a graphical user interface (GUI) for the SILK system that supports knowledge entry, query answering, and justification browsing for defeasible reasoning. The interface helps users explore prioritized rule interactions by visualizing justification trees, identifying defeated rules, and tracing inference chains — making the reasoning process more transparent and easier to debug.

Similarly, Coetzer and Britz [12] explored debugging in classical ontologies using defeasible reasoning frameworks to identify inconsistencies and suggest corrections, highlighting the crossover potential between ontology engineering and defeasible logic.

For this project, these insights are crucial, as we aim to not only implement a defeasible reasoning system but also provide a supportive environment for users to interact with and refine knowledge bases. The integration of a UI allows users to view ranked models, trace entailments, and visualise exceptions, while debugging features will assist in identifying conflicting rules and understanding why certain entailments fail. Prior work, such as DEIMOS [1], demonstrates the value of implementing reasoning systems alongside clear, human-readable outputs that explain derivations and highlight reasoning steps. This ensures that both technical and non-technical users can understand and trust the reasoning results, supporting the project's goal of developing an accessible and explainable defeasible reasoning tool.

7 CONCLUSION

This literature review has examined the progression from propositional logic to advanced forms of defeasible reasoning, outlining key theoretical developments and their relevance to intelligent systems. While propositional logic offers a foundational framework for representing and reasoning about knowledge [3], its inability to handle typicality and exceptions limits its expressiveness in dynamic, real-world contexts. The KLM framework addresses this shortcoming by introducing rationality postulates that support tentative, retractable inferences, allowing systems to reason about defaults and their exceptions [21].

Building on this framework, Casini et al. [9] introduced two primary classes of defeasible entailment: basic defeasible entailment and rational defeasible entailment—the latter instantiated by rational closure. Rational closure is recognized for its cautious and prototypical inference behavior, permitting more specific defaults to override general ones while minimizing unjustified conclusions [10, 24]. In contrast, lexicographic closure adopts a more assertive strategy, refining ranked models by considering both the specificity of defaults and the number of satisfied implications. This leads to a more decisive form of conflict resolution [9, 13, 23]. Together, these approaches span a spectrum from conservative to presumptive reasoning, offering flexible tools suited to different knowledge domains.

Despite these advances, further exploration of rational defeasible entailment beyond rational and lexicographic closure remains an open area of research [9]. Investigating new frameworks that more precisely balance caution and assertiveness could provide valuable insights for adaptive and context-sensitive reasoning systems.

As reasoning systems become more complex, supportive tools that facilitate their construction, execution, and interpretation are increasingly vital. Knowledge-based generators (KBGs) serve this role by automating the creation of structured rule sets, improving consistency, and reducing manual effort [25]. Systems developed by Hrycej [19] and Bisson [4] demonstrate key techniques such as rule materialization and conflict detection, which are critical for scalable testing and evaluation of closure algorithms. The integration of KBGs into defeasible reasoning workflows not only enhances development speed but also ensures systematic experimentation with default-based reasoning.

Equally important are user interfaces and debugging tools that promote transparency and user engagement. Graphical environments like the SILK system [18] enable users to visualize rule interactions, explore justification graphs, and inspect rule defeats. Similarly, Coetzer and Britz [12] have shown how visual debugging tools can aid ontology refinement and error detection. Earlier systems such as DEIMOS [1], further highlight the benefits of human-readable explanations for building trust and accessibility, particularly for non-expert users.

Finally, the computational complexity associated with rational and lexicographic closure necessitates ongoing performance optimization. Methods such as minimal ranked model caching, binary search over rank structures, and Horn-formula compression have proven effective in improving scalability [3, 5, 17]. Future work that combines these optimizations with novel entailment strategies, supported by automated knowledge base generation and user-centric interfaces, will be critical to building scalable, explainable, and robust reasoning systems capable of operating in real-world, exception-rich domains.

REFERENCES

[1] Antoniou, G., Billington, D., Governatori, G., Maher, M. J., and Rock, A. A family of defeasible reasoning logics and its implementation. In *Proceedings of the Australian Conference on Artificial Intelligence* (2000), Springer, pp. 335–346.

[2] Barros, M., and Couto, F. M. Knowledge representation and management: a linked data perspective. *Yearbook of Medical Informatics 25*, 1 (2016), 178–183.

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

[4] 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.

[5] Borrego-Díaz, J., and Galán Páez, J. Knowledge representation for explainable artificial intelligence. *Complex & Intelligent Systems 8* (2022), 1579–1601.

[6] Brachman, R. J., and Levesque, H. J. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

[14] Davis, R., Shrobe, H., and Szolovits, P. What is a knowledge representation? *AI Magazine 14*, 1 (1993), 17–33.

[15] Fikes, R., and Garvey, T. Knowledge representation and reasoning—a history of darpa leadership. *AI Magazine 41*, 2 (2020), 9–21.

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

[17] 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.

[18] 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 *Proceedings of the RuleML-2010 Challenge at the 4th International Web Rule Symposium (RuleML 2010)* (2010), vol. 649 of *CEUR Workshop Proceedings*, CEUR-WS.org.

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

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

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

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

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

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

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

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

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

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