UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2022

Title: Lexicographic Model-based Defeasible Reasoning

Author: Carl Combrinck

Project Abbreviation: MBDR

Supervisor(s): Professor Thomas Meyer

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	22
Experiment Design and Execution	*0*	*20*	0
System Development and Implementation	*0*	*20*	3
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	**80**

Lexicographic Model-based Defeasible Reasoning

Carl Combrinck
University of Cape Town
Cape Town, South Africa
CMBCAR007@myuct.ac.za

ABSTRACT

The primary forms of KLM-style defeasible entailment are defined syntactically, via formula-based manipulations, and semantically, using ranked models. Although the current literature describes formula-based algorithms for computing entailment, corresponding model-based approaches have not been explored. We present and analyse three model-based algorithms for computing a prominent form of defeasible entailment termed *lexicographic closure*. In each case, we define a suitable abstract representation of the ranked model, a construction algorithm, and a compatible adaptation of existing reasoning algorithms. We examine two definitions of lexicographic closure in the literature and show that these represent two distinct forms of entailment. Our analysis reveals that directly manipulating the underlying valuations is most performant, in producing the chosen lexicographic ordering.

CCS CONCEPTS

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

KEYWORDS

artificial intelligence, knowledge representation and reasoning, defeasible reasoning, rational closure, lexicographic closure

1 INTRODUCTION

A subfield of artificial intelligence (AI), knowledge representation and reasoning (KRR), formalises philosophical patterns of reasoning [7]. Symbolic representations of knowledge are collated into abstract structures termed *knowledge bases* and manipulated by several reasoning services to draw reasonable conclusions [14].

Classical propositional logic represents a simple yet expressive approach to KRR. Propositional logic possess several beneficial characteristics. However, it also has fundamental limitations precluding the expression of certain forms of human reasoning.

A crucial limitation is propositional logic's inability to capture *typicality* whereby specific implications may usually hold except for a few cases. Another is *monotonicity*, where conclusions cannot be retracted once made, even with the addition of new conflicting knowledge [10]. Such is vital in implementing forms of belief revision that involve reconsidering past conclusions with the potential for retraction.

The literature defines several defeasible approaches to reasoning as non-monotonic alternatives to classical entailment. In contrast to classical reasoning, there are no set means of defining defeasible reasoning.

Kraus, Lehmann and Magidor (KLM) [10] propose a set of properties as a thesis for how to define a 'sensible' or 'rational' form of defeasible entailment. Two such well-known examples, are *rational closure* [13] and *lexicographic closure* [12], each representing distinct, valid patterns of human reasoning. This paper focuses on lexicographic closure but notes the relationship between lexicographic and rational closure.

There are several distinct but equivalent characterisations of lexicographic closure. Lehmann first defined an ordering on valuations based on the natural lexicographic ordering of tuples associated with each valuation. These semantics are used to define formula-based algorithms for computing the same entailment relation. Casini et al. [3] define lexicographic closure, semantically, as a refinement of rational closure, noting that, like rational closure, it too can be characterised by a specific ranked model.

This paper investigates algorithms for constructing model-based representations of lexicographic closure as defined in [3]. We provide formal descriptions of each algorithm and motivate the representation and construction approaches. We aim to analyse the performance of these approaches with respect to existing methods. However, we will show that the definition provided by Casini et al., applied in our algorithms, represents a pattern of reasoning distinct from that proposed by Lehmann [12] in the original definition of lexicographic closure. We discuss this distinction using a counter-example, proving the inequality of the two orderings.

Lastly, we perform a space-time complexity analysis of each algorithm and discuss performance implications. In doing so, we establish baseline scalability constraints for this alternative definition of lexicographic closure.

2 BACKGROUND

2.1 Propositional Logic

2.1.1 Language and Semantics. We define a set $\mathcal{P}$ containing all atomic propositions, representing the most basic units of knowledge [1]. Formulas can consist of a single atom, the negations ($\neg$) of other formulas, or the combination of two other formulas using one of the binary connectives $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$. The set of all possible formulas is often referred to as $\mathcal{L}$ (the language of propositional logic). An interpretation is a function $\mathcal{I} : \mathcal{P} \rightarrow \{T, F\}$ which assigns truth values to each propositional atom. We denote the set of all propositional interpretations with $\mathscr{U}$. We say that an interpretation $\mathcal{I} \in \mathscr{U}$ satisfies a formula $\alpha \in \mathcal{L}$, denoted $\mathcal{I} \Vdash \alpha$, if α evaluates to true using the usual truth-functional semantics. We refer to a finite set of formulas as a knowledge base. We say that an interpretation $\mathcal{I}$ satisfies a knowledge base $\mathcal{K}$ if $\forall \alpha \in \mathcal{K}, \mathcal{I} \Vdash \alpha$. Interpretations that satisfy a knowledge base are referred to as models of that knowledge base. We use the notation $Mod(\mathcal{K})$ (or $[\![\mathcal{K}]\!]$) to refer to the set of models of a knowledge base $\mathcal{K}$ (similarly for a single formula).

2.1.2 *Entailment.* Using the above model-based semantics, entailment (or logical consequence), denoted using the $\models$ symbol, can be defined. A knowledge base $\mathcal{K}$ entails a formula α, written as $\mathcal{K} \models \alpha$, if and only if $Mod(\mathcal{K}) \subseteq Mod(\alpha)$. Intuitively, whenever all the formulas in $\mathcal{K}$ are true under a given interpretation, such will be the case for α and so we are able to conclude α whenever we have $\mathcal{K}$.

Example 2.1. Consider a knowledge base $\mathcal{K} = \{\mathsf{p} \vee \mathsf{q}, \neg\mathsf{p}\}$. $Mod(\mathcal{K}) = \{\overline{\mathsf{p}}\mathsf{q}\}$ ($\overline{\mathsf{p}}\mathsf{q}$ is shorthand for an interpretation that maps p to false and q to true). Consequently, $\mathcal{K} \models \mathsf{p} \rightarrow \mathsf{q}$ since $\overline{\mathsf{p}}\mathsf{q} \Vdash \mathsf{p} \rightarrow \mathsf{q}$ (equivalently, $\overline{\mathsf{p}}\mathsf{q} \in Mod(\mathsf{p} \rightarrow \mathsf{q})$) and so every model of $\mathcal{K}$ is also a model of $\mathsf{p} \rightarrow \mathsf{q}$.

2.2 Defeasible Reasoning

2.3 The KLM Framework and Extensions

Extending the preferential approach and semantics defined by Shoham [16, 17], and the associated proof-theoretic system defined by Gabbay [5], KLM [10] define a framework for defeasible reasoning referred to as the KLM framework. This framework is of particular interest because it has both a model and proof theory and computationally efficient algorithms for the associated reasoning services [9].

Initially, KLM [10] extended propositional logic by defining a consequence relation $\mid\!\sim$ representing defeasible implications in an attempt to reasonably represent *typicality*. Extensions of this framework instead define $\mid\!\sim$ as an additional connective (where $\alpha \mid\!\sim \beta$, with propositional formulas α, β, is read as 'typically, if α, then β' [3]). This extended language is defined as $\mathcal{L}_P \coloneqq \mathcal{L} \cup \{\alpha \mid\!\sim \beta \mid \alpha, \beta \in \mathcal{L}\}$ [9]. The semantics of $\mid\!\sim$ are then defined using *ranked interpretations* [13].

Definition 2.1. A ranked interpretation is a function $\mathcal{R} : \mathcal{U} \mapsto \mathcal{N} \cup \{\infty\}$, such that for every $i \in \mathcal{N}$, if there exists a $u \in \mathcal{U}$ such that $\mathcal{R}(u) = i$, then there must be a $v \in \mathcal{U}$ such that $\mathcal{R}(v) = j$ with $0 \leq j < i$, where $\mathcal{U}$ is the set of all possible propositional interpretations [9].

Ranked interpretations, therefore, assign to each propositional interpretation, a rank (with lower ranks corresponding, semantically, with more typical interpretations and higher ranks with less typical 'worlds'). Worlds with a rank of ∞, according to the ranked interpretation, are impossible, whereas worlds with finite ranks are possible.

2.3.1 *Satisfaction.* Given that ranked interpretations indicate the relative typicality of worlds, it makes sense to define whether a ranked interpretation satisfies a defeasible implication based on the most typical worlds in that interpretation. In order to define the 'most typical worlds', a definition of minimal worlds with respect to a formula in $\mathcal{L}$ is required.

Definition 2.2. Given a ranked interpretation $\mathcal{R}$ and any formula $\alpha \in \mathcal{L}$, it holds that $u \in [\![\alpha]\!]^{\mathcal{R}}$ (the models of α in $\mathcal{R}$) is minimal if and only if there is no $v \in [\![\alpha]\!]^{\mathcal{R}}$ such that $\mathcal{R}(v) < \mathcal{R}(u)$ [9].

This defines the concept of the 'best α worlds' (i.e. the lowest ranked, or most typical, of the worlds in which α is true).

Definition 2.3. Given a ranked interpretation $\mathcal{R}$ and a defeasible implication $\alpha \mid\!\sim \beta$, $\mathcal{R}$ satisfies $\alpha \mid\!\sim \beta$, written $\mathcal{R} \Vdash \alpha \mid\!\sim \beta$ if and only if for every s minimal in $[\![\alpha]\!]^{\mathcal{R}}$, $s \Vdash \beta$. If $\mathcal{R} \Vdash \alpha \mid\!\sim \beta$ then $\mathcal{R}$ is said to be a *model* of $\alpha \mid\!\sim \beta$ [9].

This says that in order for a ranked interpretation $\mathcal{R}$ to satisfy a defeasible implication $\alpha \mid\!\sim \beta$, it need only satisfy $\alpha \rightarrow \beta$ in the most typical (lowest ranked) α worlds of $\mathcal{R}$.

In the case of a propositional formula $\alpha \in \mathcal{L}$, it is required that every finitely-ranked world in $\mathcal{R}$ satisfies α in order for $\mathcal{R}$ to satisfy α. This is consistent with idea that propositional formulas, which do not permit exceptionality, should be satisfied in every plausible world of a ranked interpretation, if such a ranking is to satisfy the formula.

It is now possible to model knowledge that expresses typicality, and thus handles exceptional cases more reasonably.

We refer to a finite set of defeasible implications as a defeasible knowledge base. Note that we can express any classical propositional formula $\alpha \in \mathcal{L}$ using the defeasible representation $\neg\alpha \mid\!\sim \bot$. Henceforth, we assume that knowledge bases are defeasible unless specified otherwise.

In many cases, we require a propositional knowledge base containing the propositional analogue (or material counterpart) of each defeasible implication in a defeasible knowledge base. We term this the materialisation of a defeasible knowledge base.

Definition 2.4. The *material counterpart* of a defeasible implication $\alpha \mid\!\sim \beta$ is the propositional formula $\alpha \rightarrow \beta$. Given a defeasible knowledge base $\mathcal{K}$, the material counterpart of $\mathcal{K}$, denoted $\overrightarrow{\mathcal{K}}$, is the set of material counterparts, $\alpha \rightarrow \beta$, for every defeasible implication $\alpha \mid\!\sim \beta \in \mathcal{K}$. [9]

2.3.2 *Entailment.* We seek reasonable forms of non-monotonic entailment that permit the retraction of conclusions in cases where knowledge is added that contradicts these conclusions. Such entailment relations are defined by a set of postulates [10] which is extended to define more specific classes of entailment [3, 13]. We will look at two specific patterns of entailment, namely *rational closure* and *lexicographic closure* with a particular emphasis on their model-based semantics for the purposes of computing entailment.

2.4 Rational Closure

Rational closure represents a prototypical pattern of defeasible reasoning (one that is extremely conservative in abnormal cases) in the KLM framework. Lehmann and Magidor [13] propose that any other reasonable form of entailment, while possibly being more 'adventurous' in its conclusions, should endorse at least those assertions in the rational closure of the corresponding knowledge base.

There are 2 principle ways in which to compute the rational closure of a given knowledge base. The first is *minimal ranked entailment.* This approach defines rational closure and the semantics of the associated entailment relation using a unique ranked model for a given knowledge base. The second is an algorithmic approach involving the ranking of statements in the knowledge base [13].

Algorithm 1 `BaseRank`

Input: A knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0, ..., R_{n-1}, R_\infty, n)$
i := 0;
$E_0 := \overrightarrow{\mathcal{K}}$;
while $E_{i-1} \neq E_i$ **do**
  $E_{i+1} := \{\alpha \to \beta \in E_i \mid E_i \models \neg\alpha\}$;
  $R_i := E_i \setminus E_{i+1}$;
  $i := i + 1$;
end while
$R_\infty := E_{i-1}$;
$n := i - 1$;
return $(R_0, ..., R_{n-1}, R_\infty, n)$

Algorithm 2 `RationalClosure`

Input: A knowledge base $\mathcal{K}$, and a defeasible implication $\alpha \mid\!\sim \beta$
Output: **true**, if $\mathcal{K} \approx\!\!\!| \; \alpha \mid\!\sim \beta$, and **false** otherwise
$(R_0, ..., R_{n-1}, R_\infty, n)$:= BaseRank($\mathcal{K}$);
$i := 0$
$R := \bigcup_{i=0}^{j<n} R_j$;
while $R_\infty \cup R \models \neg\alpha$ and $R \neq \emptyset$ **do**
  $R := R \setminus R_i$;
  $i := i + 1$;
end while
return $R_\infty \cup R \models \alpha \to \beta$;

2.4.1 BaseRank and RationalClosure. Casini et al. [3] provide an algorithmic description of rational closure for computing entailment queries in terms of two sub-algorithms included as Algorithms 1 and 2. Algorithm 1 ranks the formulas of the knowledge base according to how exceptional their antecedents are, and Algorithm 2 then answers a given entailment query using the information provided by Algorithm 1.

2.4.2 Minimal Ranked Entailment. A partial order over all ranked models of a knowledge base $\mathcal{K}$, denoted $\preceq_{\mathcal{K}}$, is defined as follows [3]:

Definition 2.5. Given a knowledge base, $\mathcal{K}$, and $\mathcal{R}^{\mathcal{K}}$ the set of all ranked models of $\mathcal{K}$ (those ranked interpretations which satisfy $\mathcal{K}$), it holds for every $\mathcal{R}_1^{\mathcal{K}}, \mathcal{R}_2^{\mathcal{K}} \in \mathcal{R}^{\mathcal{K}}$ that $\mathcal{R}_1^{\mathcal{K}} \preceq_{\mathcal{K}} \mathcal{R}_2^{\mathcal{K}}$ if and only if for every $u \in \mathcal{U}$, $\mathcal{R}_1^{\mathcal{K}}(u) \leq \mathcal{R}_2^{\mathcal{K}}(u)$.

Intuitively, this partial order favours ranked models that have their worlds 'pushed down' as far as possible [9]. It has a unique minimal element, $\mathcal{R}_{RC}^{\mathcal{K}}$, as shown by Giordano et al. [6]. We now define minimal ranked entailment using this minimal element as follows:

Definition 2.6. Given a defeasible knowledge base $\mathcal{K}$, the minimal ranked interpretation satisfying $\mathcal{K}$, $\mathcal{R}_{RC}^{\mathcal{K}}$, defines an entailment relation, $\approx\!\!\!|$, called minimal ranked entailment, such that for any defeasible implication $\alpha \mid\!\sim \beta$, $\mathcal{K} \approx\!\!\!| \; \alpha \mid\!\sim \beta$ if and only if $\mathcal{R}_{RC}^{\mathcal{K}} \Vdash \alpha \mid\!\sim \beta$ [9].

Example 2.2. Consider the following knowledge base: $\mathcal{K} := \{\mathtt{bird} \mid\!\sim \mathtt{fly}, \mathtt{bird} \mid\!\sim \mathtt{wings}, \mathtt{kiwi} \to \mathtt{bird}\}$.

Intuitively, $\mathcal{K}$ suggests that birds usually fly, birds usually have wings, and kiwis are birds (kiwi here refers to the national bird of New Zealand). Using the partial order of ranked interpretations defined previously, the minimal ranked model, $\mathcal{R}_{RC}^{\mathcal{K}}$, of $\mathcal{K}$ is:

∞	$\bar{\mathtt{b}}\mathtt{fkw}$ $\overline{\mathtt{bf}}\mathtt{kw}$ $\bar{\mathtt{b}}\mathtt{fk}\bar{\mathtt{w}}$ $\overline{\mathtt{bf}}\mathtt{k}\bar{\mathtt{w}}$
1	$\mathtt{b}\bar{\mathtt{f}}\mathtt{kw}$ $\mathtt{bfk}\bar{\mathtt{w}}$ $\mathtt{b}\overline{\mathtt{fk}}\mathtt{w}$ $\mathtt{b}\bar{\mathtt{f}}\mathtt{k}\bar{\mathtt{w}}$ $\mathtt{bf}\overline{\mathtt{kw}}$ $\mathtt{b}\overline{\mathtt{fkw}}$
0	$\mathtt{bfkw}$ $\mathtt{bf}\bar{\mathtt{k}}\mathtt{w}$ $\bar{\mathtt{b}}\mathtt{f}\bar{\mathtt{k}}\mathtt{w}$ $\overline{\mathtt{bfk}}\mathtt{w}$ $\bar{\mathtt{b}}\mathtt{f}\overline{\mathtt{kw}}$ $\overline{\mathtt{bfkw}}$

Figure 1: Minimal ranked model of $\mathcal{K}$

For brevity, each proposition is represented by a single letter.

We see that $\mathcal{R}_{RC}^{\mathcal{K}} \Vdash \mathtt{kiwi} \mid\!\sim \mathtt{wings}$ since the circled minimal `kiwi` world has that `wings` is true, i.e. it follows that kiwis typically have wings ($\mathcal{K} \approx\!\!\!| \; \mathtt{kiwi} \mid\!\sim \mathtt{wings}$).

Example 2.3. Suppose the statement $\mathtt{kiwi} \to \neg\mathtt{fly}$ (that kiwis do not fly) was added to $\mathcal{K}$. The minimal ranked model $\mathcal{R}_{RC}^{\mathcal{K} \cup \{\mathtt{kiwi} \to \neg\mathtt{fly}\}}$ of $\mathcal{K} \cup \{\mathtt{kiwi} \to \neg\mathtt{fly}\}$, is:

∞	$\bar{\mathtt{b}}\mathtt{fkw}$ $\overline{\mathtt{bf}}\mathtt{kw}$ $\bar{\mathtt{b}}\mathtt{fk}\bar{\mathtt{w}}$ $\overline{\mathtt{bf}}\mathtt{k}\bar{\mathtt{w}}$ $\mathtt{bfkw}$ $\mathtt{bfk}\bar{\mathtt{w}}$
1	$\mathtt{b}\bar{\mathtt{f}}\mathtt{kw}$ $\mathtt{b}\overline{\mathtt{fk}}\mathtt{w}$ $\mathtt{b}\bar{\mathtt{f}}\mathtt{k}\bar{\mathtt{w}}$ $\mathtt{bf}\overline{\mathtt{kw}}$ $\mathtt{b}\overline{\mathtt{fkw}}$
0	$\mathtt{bf}\bar{\mathtt{k}}\mathtt{w}$ $\bar{\mathtt{b}}\mathtt{f}\bar{\mathtt{k}}\mathtt{w}$ $\overline{\mathtt{bfk}}\mathtt{w}$ $\bar{\mathtt{b}}\mathtt{f}\overline{\mathtt{kw}}$ $\overline{\mathtt{bfkw}}$

Figure 2: Minimal ranked model of $\mathcal{K} \cup \{\mathtt{kiwi} \to \neg\mathtt{fly}\}$

Now, notice that $\mathcal{R}_{RC}^{\mathcal{K} \cup \{\mathtt{kiwi} \to \neg\mathtt{fly}\}} \nVdash \mathtt{kiwi} \mid\!\sim \mathtt{wings}$, since the circled minimal `kiwi` worlds do not both have `wings` being true. This demonstrates the non-monotonicity of rational closure, as a previous conclusion was retracted with the addition of new information. Importantly, it also demonstrates the conservative nature of prototypical reasoning, formalized in rational closure. In $\mathcal{K} \cup \{\mathtt{kiwi} \to \neg\mathtt{fly}\}$, kiwis are atypical birds since they are birds that do not fly and hence don't conform to the prototype of a bird, warranting the retraction of the conclusion in example 2.2.

2.5 Lexicographic Closure

Lexicographic closure is a formalism of the presumptive pattern of reasoning introduced by Reiter [15] in the context of default logics. Presumptive reasoning is more 'adventurous' and willing to conclude statements so long as their is no evidence to the contrary (even in atypical cases). The semantics of lexicographic closure depends on an ordering that is defined based on two criteria: seriousness and cardinality.

Like rational closure, there are syntactic (formula-based) and semantic (model-based) descriptions of lexicographic closure [3, 12].

Lehmann first defined lexicographic closure using a modular partial ordering on valuations [12]. This ordering favoured valuations with lower violation tuples, according to the natural lexicographic ordering of tuples. A violation tuple of a valuation is derived from the subset of a given defeasible knowledge base containing all the formulas the valuation violates. The tuple records the counts of formulas violated by the valuation ordered by seriousness (in this case, the base rank of the formula).

Definition 2.7. Given a defeasible knowledge base $\mathcal{K}$ of order k, every subset $D \subseteq \mathcal{K}$ has a corresponding $k+1$ tuple of natural numbers, denoted n_D, $\langle n_0, ..., n_k \rangle$, where each is defined as such: $n_0 = |\{\alpha \mid\sim \beta \in D \mid br_{\mathcal{K}}(\alpha) = \infty\}|$, $n_1 = |\{\alpha \mid\sim \beta \in D \mid br_{\mathcal{K}}(\alpha) = k-1\}|$ and for any $i = 1, ..., k$, $n_i = |\{\alpha \mid\sim \beta \in D \mid br_{\mathcal{K}}(\alpha) = k-i\}|$. That is, n_0 is the number of defeasible implications of infinite base rank, or having no rank, in D, and each n_i for $0 < i \leq k$ is the number of defeasible implications of base rank $k-i$ in D [9].

Definition 2.8. Given a defeasible knowledge base $\mathcal{K}$, a seriousness ordering on subsets $D \subseteq \mathcal{K}$ is a modular partial ordering, denoted $\prec_S$, that is a lexicographic ordering over the tuples of ranks for each subset. That is, given two subsets, $D_1, D_2 \subseteq \mathcal{K}$, $D_1 \prec_S D_2$ if and only if $n_{D_1} \prec_S n_{D_2}$ using the natural lexicographic ordering over tuples of natural numbers, e.g. $\langle 1,0,2 \rangle \prec_S \langle 1,1,2 \rangle$. $\prec_S$ is a strict modular partial order over subsets of $\mathcal{K}$ [9].

Definition 2.9. Given a defeasible knowledge base $\mathcal{K}$, and valuations $m, n \in \mathcal{U}$, the preference order $\prec_{LC}^{\mathcal{K}}$ over $\mathscr{U}$ is defined as: $m \prec_{LC}^{\mathcal{K}} n$ if and only if $V(m) \prec_S V(n)$ where $V(m) \subseteq \mathcal{K}$ is the set of defeasible implications violated by $m \in \mathcal{U}$. $\prec_{LC}^{\mathcal{K}}$ is a modular partial order over $\mathscr{U}$, and so defines a ranked interpretation, denoted $\mathcal{R}_{LC}^{\mathcal{K}}$ [9].

A formula-based algorithm for computing lexicographic closure, based on Lehmann's definition [12], successively produces weakened formula representations of each base rank. We refer to this algorithm as the `LexicographicClosure` algorithm [4], defined in Algorithm 3. It proceeds in the same manner as the `RationalClosure` algorithm but weakens each rank, by considering incrementally smaller subsets of the rank, instead of completely discarding the entire rank at each iteration.

Algorithm 3 `LexicographicClosure`

Input: A knowledge base $\mathcal{K}$, and a defeasible implication $\alpha \mid\sim \beta$
Output: **true**, if $\mathcal{K} \approx_{LC} \alpha \mid\sim \beta$, and **false** otherwise
$(R_0, ..., R_{n-1}, R_\infty, n) :=$ BaseRank($\mathcal{K}$);
$i := 0$
$R := \bigcup_{i=0}^{j<n} R_j$;
while $R_\infty \cup R \models \neg\alpha$ and $R \neq \emptyset$ **do**
  $R := R \setminus R_i$;
  $m := \#R_i$ - 1;
  $R_{i,m} := \bigvee_{S \in \{T \subseteq R_i | \#T = m\}} \bigwedge_{s \in S} s$;
  while $R_\infty \cup R \cup \{R_{i,m}\} \models \neg\alpha$ and $m > 0$ **do**
    $m := m - 1$;
    $R_{i,m} := \bigvee_{S \in \{T \subseteq R_i | \#T = m\}} \bigwedge_{s \in S} s$
  end while
  $R := R \cup \{R_{i,m}\}$;
  $i := i + 1$;
end while
return $R_\infty \cup R \models \alpha \rightarrow \beta$;

Casini et al. provide another model-based definition of lexicographic closure in their framework of rational defeasible entailment relations [3]:

Definition 2.10. $m \prec_{LC}^{\mathcal{K}} n$ if and only if $\mathcal{R}_{RC}^{\mathcal{K}}(n) = \infty$, or $\mathcal{R}_{RC}^{\mathcal{K}}(m) < \mathcal{R}_{RC}^{\mathcal{K}}(n)$, or $\mathcal{R}_{RC}^{\mathcal{K}}(m) = \mathcal{R}_{RC}^{\mathcal{K}}(n)$ and m satisfies more formulas than n in $\mathcal{K}$.

This definition characterises the lexicographic closure ordering as a count-based refinement of the rational closure ordering, in that its ranked model respects the rankings of rational closure (which encode seriousness) but refines preference for worlds with the same rank based on the total number of formulas each satisfies.

We now define the lexicographic closure entailment relation $\approx_{LC}$ as follows [3]:

Definition 2.11. Given a defeasible knowledge base $\mathcal{K}$, the ranked model derived from the modular ordering in definition 2.10, $\mathcal{R}_{LC}^{\mathcal{K}}$, defines an entailment relation, $\approx_{LC}$, such that for any defeasible implication $\alpha \mid\sim \beta$, $\mathcal{K} \approx_{LC} \alpha \mid\sim \beta$ if and only if $\mathcal{R}_{LC}^{\mathcal{K}} \Vdash \alpha \mid\sim \beta$ [3].

Example 2.4. Returning to our kiwi example where $\mathcal{K} := \{$`bird` $\mid\sim$ `fly`, `bird` $\mid\sim$ `wings`, `kiwi` $\rightarrow$ `bird`, `kiwi` $\rightarrow \neg$`fly`$\}$, we can construct the model $\mathcal{R}_{LC}^{\mathcal{K}}$ corresponding to lexicographic closure by 'lifting up' worlds that satisfy fewer statements while preserving the original rational closure ordering.

∞	$\overline{b}fkw$ $\overline{bf}kw$ $\overline{b}fk\overline{w}$ $\overline{bf}k\overline{w}$ $bfkw$ $bfk\overline{w}$
2	$b\overline{f}k\overline{w}$ $b\overline{fkw}$
1	$b\overline{f}kw$ $b\overline{fk}w$ $bf\overline{kw}$
0	$bf\overline{k}w$ $\overline{b}f\overline{k}w$ $\overline{bfk}w$ $\overline{b}f\overline{kw}$ $\overline{bfkw}$

Figure 3: Ranked model for Lexicographic Closure of $\mathcal{K}$

Checking if the formula, `kiwi` $\mid\sim$ `wings`, is satisfied by $\mathcal{R}_{LC}^{\mathcal{K}}$, and hence in the lexicographic closure of $\mathcal{K}$, we find that $\mathcal{R}_{LC}^{\mathcal{K}} \Vdash$ `kiwi` $\mid\sim$ `wings` and hence that $\mathcal{K} \approx_{LC}$ `kiwi` $\mid\sim$ `wings`.

Notice that lexicographic closure endorses that kiwis have wings whereas rational closure would not. This speaks to the presumptive nature of lexicographic closure, as it is willing to assert that kiwis have wings despite the fact that kiwis are atypical birds (since there is no knowledge to the contrary).

This also demonstrates that there are multiple valid solutions to the problem of defeasible entailment. We may prefer the behaviour of rational closure in this case (since we know kiwis don't have wings), but may prefer lexicographic closure if kiwis are substituted for penguins, which do have wings.

It is important to note that we have mentioned two model-based definitions of lexicographic closure, one defined by Lehmann based on violation tuples, and the other defined by Casini et al. based on a count-based refinement of lexicographic closure. While these may appear to produce the same ranked model and thus the same entailment relation as claimed in [3], we will show this not to be the case. Henceforth, we use the definition for lexicographic closure proposed by Casini et al., unless otherwise stated, and, where necessary, refer to the associated ordering as the count-based lexicographic ordering.

3 ALGORITHM DEVELOPMENT

Our construction algorithms refine the output of the `ModelRank`, `FormulaRank`, and `CumulativeFormulaRank` algorithms, by Cohen, according the count-based lexicographic ordering.

3.1 Model Construction

3.1.1 Motivation. We wish to construct the ranked model corresponding to the lexicographic ordering in definition 2.10 using an approach similar to that of `ModelRank` by Cohen. `ModelRank` employs an ad hoc 'pushing up' of remaining worlds based on a similar algorithm defined in [2] for propositional typicality logic.

Given the rational closure model and counts for each model, representing the number of formulas satisfied, there does not seem to be a straightforward way of directly computing the lexicographic rank of a valuation. Because the number of refined ranks produced from a rational closure rank varies, a less complicated strategy would be to employ a procedure that ranks valuations iteratively, removing the need to place the worlds directly. As the lexicographic ordering gives preference to valuations on lower rational closure ranks and satisfying more formulas, respectively, a simple approach would be to consider, for each of the rational closure ranks in turn, every possible count of formulas that could be violated in ascending order. For each rational closure rank, the algorithm could place all valuations, if any exist, violating the current count, removing these from a set of remaining worlds to place. This resembles the 'pushing up' of worlds in `ModelRank` and would ensure that the relative order of worlds in the rational closure rank is maintained in the lexicographic model, while refining based on formulas violated to produced the required ordering.

We formalize this bottom-up construction of the lexicographic ranked model in the `LexicographicModelRank` algorithm and provide proofs of correctness in Appendix A.

Algorithm 4 `LexicographicModelRank`

Input: A defeasible knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0^{LC}, ..., R_{k-1}^{LC}, R_\infty^{LC}, k)$
$(R_0^{RC}, ..., R_{n-1}^{RC}, R_\infty^{RC}, n)$:= `ModelRank`$(\mathcal{K})$;
$i := 0$; ▷ rational closure rank
$k := 0$; ▷ lexicographic closure rank
while $i < n$ **do**
  $j := 0$; ▷ number of formulas to violate
  $\mathscr{U}_{ij} := R_i^{RC}$; ▷ remaining worlds to place
  while $\mathscr{U}_{ij} \neq \emptyset$ **do**
    $L_{ij} := \{u \in \mathscr{U}_{ij} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \not\Vdash k\} = j\}$;
    if $L_{ij} \neq \emptyset$ **then**
      $R_k^{LC} := L_{ij}$; ▷ place worlds violating j formulas
      $k := k + 1$;
    end if
    $\mathscr{U}_{i(j+1)} := \mathscr{U}_{ij} \setminus L_{ij}$; ▷ remove placed worlds
    $j := j + 1$;
  end while
  $i := i + 1$;
end while
$R_\infty^{LC} := R_\infty^{RC}$;
return $(R_0^{LC}, ..., R_{k-1}^{LC}, R_\infty^{LC}, k)$

3.1.2 Entailment. Given that `LexicographicModelRank` produces the ranked model corresponding to the lexicographic ordering in [3], we can use this to generate the lexicographic closure of a knowledge base according to definition 2.11. Algorithm 5 defines a straightforward procedural application of the definition for the satisfaction of a defeasible formula by a ranked interpretation, necessary for computing the lexicographic closure. Definition 2.11 states that a defeasible implication is in the lexicographic closure of a knowledge base if and only if the corresponding lexicographic ranked model satisfies the implication. Therefore, to answer a query, we construct the lexicographic ranked model using `LexicographicModelRank` and pass the result as input with the query to the `ModelSatisfaction` algorithm.

Algorithm 5 `ModelSatisfaction`

Input: A ranked interpretation $R^* := (R_0, ..., R_{n-1}, R_\infty)$, the number of ranks, n, and a query defeasible implication, $\alpha \mid\sim \beta$.
Output: **true** if $R^* \Vdash \alpha \mid\sim \beta$, otherwise **false**
$i := 0$;
while $R_i \cap [\![\alpha]\!] = \emptyset$ and $i < n$ **do** ▷ No α worlds found
  $i := i + 1$;
end while
return $R_i \cap [\![\alpha]\!] \subseteq [\![\beta]\!]$; ▷ All minimal α worlds are β worlds

The `ModelSatisfaction` algorithm checks whether the minimal worlds consistent with the query antecedent, also satisfy the query's consequent, as required by definition 2.3.

3.2 Formula Representation

3.2.1 Motivation. The motivation for considering a formula-based lexicographic algorithm is precisely that for developing a formula-based rational closure algorithm. We take issue with the implementation of approaches that directly manipulate models, as such requires the enumeration of all worlds, which may lead to performance issues with the exponential growth of worlds in the number of unique propositional atoms. We, therefore, adapt our `LexicographicModelRank` algorithm to represent the models on each rank syntactically, by constructing a formula F_i for each rank i such that $Mod(F_i) = L_i$, the worlds on the i^{th} rank of the lexicographic ranked model.

Using a formula-based version of the refinement strategy in `LexicographicModelRank`, we represent worlds satisfying a particular number of formulas n using all possible subsets of the knowledge with cardinality n. Combining these subsets disjunctively, we construct a formula with models satisfying at least n formulas, or equivalently, violating no more than $\#\mathcal{K} - n$ formulas. Therefore, when refining each rank, we start by constructing a formula with worlds on the rank violating no more than 0 formulas and, if any exist, removing these from the formula representing the remaining worlds. This process continues, as in `LexicographicModelRank`, until there are no remaining worlds and is repeated for each rank.

We prove, in Appendix B, that the models of each formula in this representation correspond exactly to the ranks of the model produced by the `LexicographicModelRank` algorithm.

Algorithm 6 `LexicographicFormulaRank`

Input: A defeasible knowledge base $\mathcal{K}$
Output: An ordered tuple $(F_0^{LC}, ..., F_{k-1}^{LC}, F_\infty^{LC}, k)$
$(F_0^{RC}, ..., F_{n-1}^{RC}, F_\infty^{RC}, n) :=$ `FormulaRank`$(\mathcal{K})$;
$i := 0$; ▷ rational closure rank
$k := 0$; ▷ lexicographic closure rank
while $i < n$ **do**
  $j := 0$; ▷ number of formulas to violate
  $\mathcal{U}_{ij} := F_i^{RC}$; ▷ remaining worlds to place
  while $\mathcal{U}_{ij} \not\models \bot$ **do**
    $L_{ij} := \mathcal{U}_{ij} \wedge \left(\bigvee_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}} - j\}} \bigwedge_{s \in S} s \right)$;
    if $L_{ij} \not\models \bot$ **then**
      $F_k^{LC} := L_{ij}$; ▷ place worlds violating j formulas
      $k := k + 1$;
    end if
    $\mathcal{U}_{i(j+1)} := \mathcal{U}_{ij} \wedge \neg L_{ij}$; ▷ remove placed worlds
    $j := j + 1$;
  end while
  $i := i + 1$;
end while
$F_\infty^{LC} := F_\infty^{RC}$;
return $(F_0^{LC}, ..., F_{k-1}^{LC}, F_\infty^{LC}, k)$

3.2.2 Entailment. We modify the `ModelSatisfaction` algorithm to use the formula representation returned by `LexicographicFormulaRank` by similarly converting the valuation set operations in the algorithm to the corresponding formula-based semantics.

Algorithm 7 `FormulaModelSatisfaction`

Input: A formula-based representation of a ranked interpretation $F^* := (F_0, ..., F_{n-1}, F_\infty)$, the number of ranks, n, and a query defeasible implication, $\alpha \mid\sim \beta$.
Output: **true** if $R^* \Vdash \alpha \mid\sim \beta$ where R^* is the ranked interpretation associated with F^*, otherwise **false**
$i := 0$;
while $F_i \wedge \alpha \models \bot$ and $i < n$ **do** ▷ No α worlds found
  $i := i + 1$;
end while
return $F_i \wedge \alpha \models \beta$; ▷ All minimal α worlds are β worlds

3.3 Cumulative Model Construction

3.3.1 Motivation. While the `LexicographicFormulaRank` algorithm removes the need to enumerate and manipulate the valuations directly, it still suffers from similar implementation-based constraints. When refining ranks, the algorithm uses the formula $\mathcal{U}_{ij}$ to represent the remaining yet-to-be-placed valuations. The algorithm defines $\mathcal{U}_{i(j+1)}$ as the conjunction of $\mathcal{U}_{ij}$ and the negation of the current lexicographic rank formula L_{ij}, analogous to excluding placed valuations from those remaining. The lexicographic rank formulas are defined in terms of the current $\mathcal{U}_{ij}$ formula.

Consequently, formulas on each rank grow exponentially in the number of knowledge base formulas syntactically included, resulting in an intractable solution for knowledge bases containing more than a few statements. While there are some ways of improving this, such as only including the negated current rank for satisfiable L_{ij}, such would not improve the worst-case exponential growth.

We obtain a more compact representation by removing the negation of prior ranks, intending to produce a cumulative model representation. Cohen takes this approach to produce a cumulative rational closure model representation. In this case, the result is strongly tied to the original formula-based algorithm for computing rational closure. The cumulative rank formulas corresponded precisely to the exceptional sets in the original algorithm. Therefore, this cumulative approach resembles a cached version of the `RationalClosure` algorithm in which the exceptional sets in each iteration are precomputed and stored.

We wish to formulate a similarly cumulative approach for computing lexicographic closure that exemplifies the relationship between the model-theoretic definition and the usual `LexicographicClosure` algorithm. Based on the cumulative rational closure approach findings, we expect that the cumulative model ranks correspond to various iterations of the original `LexicographicClosure` algorithm.

However, attempting to do so, we find that the count-based definition of lexicographic closure applied in our approaches is not equivalent to that initially proposed by Lehmann, contrary to what is stated in [3]. We describe this distinction fully in 4.

Consequently, there is no formula-based algorithm for specifically computing this form of lexicographic closure as the `LexicographicClosure` algorithm corresponds to the lexicographic closure ordering defined by Lehmann [12]. Further, the count-based ordering may not possess the same cumulative properties as the Lehmann ordering, allowing the `LexicographicClosure` algorithm to efficiently represent cumulative model ranks at each iteration. For example, it has not been shown that the set of valuations on rank i or less satisfying j or more formulas will also include valuations on rank $i-1$ or less satisfying any number of formulas. Equivalently, it may be possible to have two valuations, one on rank i satisfying j formulas and the other on a rank, say $i' > i$, satisfying $j' < j$ formulas (it is not necessarily the case that a lower rational closure rank implies a lower total formula satisfaction count).

Therefore, in defining a cumulative formula-based representation of the ranked model, we explicitly include a representation of the previous rational closure ranks to ensure the cumulative property is upheld. The algorithm is mechanically similar to `LexicographicFormulaRank` but is adjusted to account for cumulative ranks. In particular, the notion of remaining worlds to place is now checked by observing whether the last cumulative lexicographic formula's model set does not contain the models of the current cumulative rational closure formula. Similarly, refined cumulative formulas are only included if the last lexicographic cumulative formula's model set does not contain the models of the refined formula.

While justification for the adaption of the `LexicographicFormulaRank` is provided, we leave the proof of the cumulative model construction as future work, owing to the need for further exploration of the count-based lexicographic ordering. Such exploration may produce more efficient representations for computing the associated entailment relation and clarify the cumulative properties of the ordering.

Algorithm 8 `LexicographicCumulativeFormulaRank`

Input: A defeasible knowledge base $\mathcal{K}$
Output: An ordered tuple $(C_0^{LC}, ..., C_{k-1}^{LC}, C_\infty^{LC}, k)$
$(C_0^{RC}, ..., C_{n-1}^{RC}, C_\infty^{RC}, n) :=$ `CumulativeFormulaRank`$(\mathcal{K})$;
$C_{-1}^{RC} := \bot$;
$C_{-1}^{LC} := \bot$; ▷ required in first iteration of outer loop
$i := 0$; ▷ rational closure rank
$k := 0$; ▷ lexicographic closure rank
while $i < n$ **do**
  $j := 0$; ▷ number of formulas to violate
  while $C_i^{RC} \not\models C_{k-1}^{LC}$ **do** ▷ until no remaining worlds
    $L_{ij} := C_{i-1}^{RC} \vee \left(C_i^{RC} \wedge \left(\bigvee_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}} - j\}} \bigwedge_{s \in S} s\right)\right)$;
    if $L_{ij} \not\models C_{k-1}^{LC}$ **then** ▷ contains additional worlds
      $C_k^{LC} := L_{ij}$; ▷ place worlds violating $\leq j$ formulas
      $k := k + 1$;
    end if
    $j := j + 1$;
  end while
  $i := i + 1$;
end while
$C_\infty^{LC} := C_\infty^{RC}$;
return $(C_0^{LC}, ..., C_{k-1}^{LC}, C_\infty^{LC}, k)$

3.3.2 Entailment. The definition of ranked interpretation satisfaction only considers the minimal worlds consistent with the query antecedent. Both `ModelSatisfaction` and `FormulaModelSatisfaction` process ranks in order, checking for the existence of minimal worlds. Therefore, a cumulative representation does not affect the correctness of these algorithms as worlds inconsistent with the query antecedent from prior ranks have already been checked in prior iterations and must not constitute minimal α worlds.

We, therefore, use the `FormulaModelSatisfaction` algorithm for computing entailment with the cumulative formula representation without the need for modification.

Examples of this, and the other algorithms can be found in appendix C.

4 LEXICOGRAPHIC CLOSURE

We make an important observation regarding the distinction between definitions of lexicographic closure presented in [12] and [3].

The refinement definition of lexicographic closure applied in our model-based algorithms defines an ordering based on the criteria of seriousness and count (similar to the ordering defined by Lehmann). However, we find that this is, in fact, distinct from the ordering originally defined for lexicographic closure in [12].

We prove, via an example, how these definitions differ regarding the produced ranked models.

Example 4.1. Consider $\mathcal{K} = \{p \to b, b \mid\sim f, b \mid\sim w, p \mid\sim \neg f, p \mid\sim w\}$.

This represents the knowledge that all penguins are birds, birds typically fly, birds typically have wings, penguins typically don't fly, and penguins typically have wings.

The base rank of the formulas in $\mathcal{K}$ and the corresponding minimal ranked (rational closure) model are shown in figure 4.

∞	$p \to b$
1	$p \mid\sim \neg f, p \mid\sim w$
0	$b \mid\sim f, b \mid\sim w$

(a) Base Rank

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

(b) Minimal Ranked Model

Figure 4: Base Rank and Minimal Ranked Model of $\mathcal{K}$

We construct the ranked models in figure 5 according to the, purportedly equivalent, Lehmann [12] and Casini et al. [3] definitions of lexicographic closure.

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

(a) Lexicographic Closure [12] Model

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

(b) Lexicographic Closure [3] Model

Figure 5: The Two Lexicographic Ranked Models

We observe that both lexicographic models respect the relative order of the worlds in the rational closure model.

However, we notice a difference in the refinement of the second rational closure rank in producing the two lexicographic models. In particular, consider valuations $\mathsf{bfp\overline{w}}$ and $\mathsf{b\overline{f}p\overline{w}}$(circled in the rational closure model). The tuples of violated formula counts ordered by base rank, as defined in the Lehmann definition of lexicographic closure [12], are $\langle 0,2,1\rangle$ and $\langle 0,1,2\rangle$, respectively. We define similar tuples based on the refinement criteria of the Casini et al. lexicographic ordering [3] to include the rational closure rank and the number of formulas violated in $\mathcal{K}$ (as the ordering favours valuations with a lower rational closure rank, violating fewer formulas in $\mathcal{K}$). Both valuations, in this case, have the same rational closure rank of two and violate three formulas in $\mathcal{K}$. Hence the corresponding tuple associated with these valuations is $\langle 2,3\rangle$. Noting that both partial orders can be defined by comparing the corresponding tuples using the natural lexicographic ordering of tuples, the Lehmann ordering places the valuations on different ranks while the Casini et al. ordering places them on the same rank.

Example 4.1 demonstrates that refining the rational closure model ranks by count alone does not necessarily produce the Lehmann lexicographic closure ranked model. The rational closure model separates valuations according to the number of trailing zeroes in their formula violation tuples or, equivalently, the highest base rank among the formulas each violates (based on the connection between the base rank of a formula and the rank of its minimal world in the minimal ranked model [6]). Therefore, to refine such to produce the Lehmann lexicographic closure model, valuations on the same rational closure rank must be separated, not based on the number of formulas violated in total [3], but rather by considering each of the remaining violation counts in turn (essentially completing the lexicographic tuple comparison).

Nonetheless, the ordering defined in [3] still represents a valid form of lexicographic closure in which the tuples used to compare valuations are of order two, consisting of a valuation's rational closure rank and formula violation count. Notably, the ranked models corresponding to each ordering constitute refinements of the rational closure model and will, therefore, fall within the rational defeasible entailment framework described in [3].

While it would be possible to construct a cumulative version of the `LexicographicFormulaRank` algorithm by instead refining cumulative rational closure ranks without the negation of prior ranks (much like in the `CumulativeFormulaRank` algorithm), we would need to show that such represents the cumulative model. The output of the `CumulativeFormulaRank` algorithm is cumulative as the formulas on each rank become strictly weaker as the rank increases. While such is also the case for iterations of the `LexicographicClosure` algorithm, it may not hold for the modified `LexicographicFormulaRank`, as described above. Therefore, our cumulative algorithm explicitly includes the prior cumulative rational closure rank when constructing a rank's representative formula, as it is possible for a lower-ranked valuation to violate more formulas than a higher-ranked valuation.

5 ALGORITHM ANALYSIS

We consider each of the developed approaches' worst-case run-time and space complexity for an input defeasible knowledge base of size n with p unique propositional atoms.

5.1 LexicographicModelRank

5.1.1 Time Complexity. The `LexicographicModelRank` algorithm refines each of the ranks produced by the `ModelRank` algorithm. In the worst case, the rational closure ranked model has $n+1$ finite ranks and an infinite rank, totalling $n+2$ ranks. This upper bound is a consequence of the worst-case output of the `BaseRank` algorithm for the knowledge base, in which each rank contains a single formula corresponding to a ranked model with at most $n+1$ finite ranks.

Each rational closure rank is refined by placing, and removing from the set of remaining worlds, those worlds which violate an incrementally increasing number of formulas in the knowledge base. In the worst case, this number, starting from 0, reaches the size of the knowledge base n before the termination condition is satisfied. For a particular violation count, we need to consider each of the, at most, 2^p worlds in the remaining worlds. For a given valuation, the algorithm performs a satisfaction ($\Vdash$) check for each of the n formulas in the knowledge.

Therefore, a conservative upper bound on the number of satisfaction checks is $(n+1)\times(n+1)\times 2^p\times n\approx n^3\times 2^p$ (excluding those performed in the `ModelRank` algorithm).

We note that this represents a conservative upper bound on the number of satisfaction checks. The estimated operation count includes repeated operations performed when placing worlds. Noting that all worlds will be considered at some iteration of the algorithm, precomputing and storing the violation counts for each world reduces this complexity to $O(n\times 2^p)$ (we will see that this does not impact space complexity).

If we assume that an entailment check corresponds to $O(2^p)$ satisfaction checks for the given knowledge base, then we have that the algorithm is $O(n)$ for the entailment operation. Noting that the `ModelRank` algorithm performs $O(n^2)$ such entailment operations, as shown by Cohen, the `LexicographicModelRank` algorithm performs $O(n+n^2)=O(n^2)$ classical entailment checks.

5.1.2 Space Complexity. Irrespective of the input knowledge base size and whether we choose to precompute and store violation counts for each world, the space complexity of the algorithm is $O(2^p)$ corresponding to the number of possible valuations. These must be stored as the output of the `ModelRank` algorithm, potentially stored with the number of formulas each violates, and returned in the output.

5.2 LexicographicFormulaRank

5.2.1 Time Complexity. Similarly, in the `LexicographicFormulaRank` algorithm, there are at most $n+1$ finite ranks, each requiring at most $n+1$ refinements. Of interest is the number of entailment checks performed. In this case, the algorithm performs 2 per refinement (firstly, in the inner while loop condition, and secondly, in the if statement condition). Therefore, the total classical entailment checks performed is $(n+1)\times(n+1)\times 2\approx 2\times n^2$, and so the time complexity is $O(n^2)$ in entailment operations. Including the time complexity of the FormulaRank algorithm initially invoked, we obtain a final time complexity of $O(n^2 + n^2) = O(n^2)$.

5.2.2 Space Complexity. The models' formula representations store one formula per rank. Therefore, there will be at most $n + 2$ such formulas produced by the FormulaRank algorithm, and $(n + 1) \times (n + 1) \approx n^2$ produced during rank refinement. This results in a space complexity of $O(n^2 + n) = O(n^2)$ for representative rank formulas.

However, it is not sufficient to consider the number of rank formulas in isolation. We note that these formulas may differ substantially in length and complexity and may affect space complexity. Such will likely impact run-time performance but should not affect the run-time complexity class.

We consider the length of the representative formulas produced in the algorithm in terms of the number of knowledge base formulas each syntactically comprises. For example, a representative formula that is the conjunction of 4 knowledge base formulas will be considered to comprise 4 knowledge base formulas. The primary assumption is that knowledge base formulas have similar complexity and size.

Any given representative formula comprises the formula representing the remaining worlds on the current rank and the formulas from all the possible subsets of a given size. In the worst case, the remaining world's formula comprises $O(n \times 2^n)$ knowledge base formulas, as Cohen discusses. Subsets approximately half the size of the knowledge yield the worst case in terms of formula count. Therefore the total number of formulas in the subset portion of the representative formula is $O(\lfloor \frac{n}{2} \rfloor \times 2^n) = O(n \times 2^n)$, as each subset contains $\lfloor \frac{n}{2} \rfloor$ formulas. There are more accurate approximations of $\binom{n}{\lfloor \frac{n}{2} \rfloor}$, but for simplicity, we use 2^n as an upper bound.

Therefore, we have at most each formula consists of $O(n \times 2^n + n \times 2^n) = O(n \times 2^n)$ formulas. With $O(n^2)$ such formulas, a more representative space complexity is $O(n^2 \times n \times 2^n) = O(n^3 \times 2^n)$ in the number of knowledge base formulas.

5.3 LexicographicCumulativeFormulaRank

5.3.1 Time Complexity. As is the case in the LexicographicFormulaRank algorithm, the LexicographicCumulativeFormulaRank requires at most $O(n^2)$ classical entailment checks and stores $O(n^2)$ formulas.

5.3.2 Space Complexity. However, the formulas no longer include the negation of prior rank formulas. The cumulative formulas are shown to comprise $O(n)$ knowledge base formulas, while the subset conjunction-disjunction still requires $O(n \times 2^n)$ space. Space complexity adjusted for knowledge base formula count is, therefore, $O(n^2 \times (n + n \times 2^n)) = O(n^3 \times 2^n)$. Despite the reduction in formulas, the space complexity class is unchanged from the LexicographicFormulaRank algorithm, due to the disjunctively-combined subsets.

5.4 ModelSatisfaction

For the ModelSatisfaction and FormulaModelSatisfaction, we take note of the time complexity, ignoring the space complexity of the input as this is accounted for in the ranked model construction algorithms. In both cases, no additional space is required when answering an entailment query.

In determining the minimal worlds, the ModelSatisfaction algorithm implicitly uses satisfaction checks on the valuations in the input ranked interpretation. Should any exist on the current rank, a second satisfaction check performed implicitly, in computing the algorithm output, verifies whether all are consistent with the query.

In the worst case, all worlds are minimal and consistent with the query antecedant, requiring a total of 2×2^p or $O(2^p)$ satisfaction checks. Using the same assumption in 5.1.1, this is equivalent to $O(1)$ entailment checks.

	Model	Formula	Cumulative
Time	$O(n^2)$	$O(n^2)$	$O(n^2)$
Space	$O(2^p)$	$O(n^3 \times 2^n)$	$O(n^3 \times 2^n)$

Figure 6: Algorithm Time and Space Complexities

	Model	Formula
Time	$O(1)$	$O(n)$

Figure 7: Entailment Algorithm Time Complexities

5.5 FormulaModelSatisfaction

At each iteration, an entailment check is performed, followed by one final check in the return statement. Therefore, in the worst case, the representation has $n + 1$ finite ranks, all of which are checked, for a total of $n + 2$ or $O(n)$ entailment checks.

6 RESULTS AND DISCUSSION

In all three cases, we observe that the time complexity for each model construction algorithm is the same ($O(n^2)$ in classical entailment). Using the resulting model representations, entailment requires a $O(1)$ entailment checks per query in the case of ModelSatisfaction, and $O(n)$ checks in the case of FormulaModelRank.

Noting that the problem of boolean satisfiability for propositional logic is solvable in non-deterministic polynomial time (NP) [13], our time complexity results for the various algorithms suggest that the decision problem of whether a query is in the count-based lexicographic closure of a knowledge base is not much more complex than that of classical entailment. Combining model construction with a single defeasible entailment query check requires $O(n^2)$ classical entailment checks in all cases.

However, LexicographicModelRank is a member of a space complexity class distinct from the formula representation-based algorithms.

It is essential to note the distinction between the quoted space complexity classes. The LexicographicModelRank space complexity refers to the number of atoms, whereas that of the remaining algorithms refers to the number of knowledge base formulas. It is not easy to directly compare these space complexity measures as the extent to which these correlate depends on the knowledge base.

Therefore, the choice of algorithm for a given knowledge base will depend on the knowledge base size and unique atom count. Further, there does not seem to be a canonical data set from which to derive a general relationship between these variables.

Suppose we assume that knowledge bases generally have more formulas than atoms, representing the case where we have rich, defeasible information about a few entities and that the size of valuations stored is not much more than the formulas themselves. In that case, we can safely conclude that `LexicographicModelRank` is the optimal choice of algorithm.

Implementations of these algorithms using the Tweety Project libraries [18] and Sat4j [11] classical reasoner in Java confirm our space complexity results as the constraining factor on algorithm scalability. In each case, space complexity is exponential or super-exponential, leading to out-of-memory errors during model construction. We find that the `LexicographicModelRank` was able to handle a knowledge base size of 38 formulas (20 unique atoms) while `LexicographicFormulaRank` and `LexicographicCumulativeFormulaRank` only managed 4 and 13 formulas, respectively. In these cases, construction times were in the order of minutes, likely too time-consuming for any substantial practical application. Entailment times significantly favoured model representations compared to the formula representations, owing to the distinct time complexity classes in figure 7, strengthening the case for the purely model-based algorithms.

As we show that the count-based lexicographic ordering is distinct from the Lehmann ordering, these results represent a scalability baseline for future comparison. Noting the reasonably performant nature of formula-based algorithms for rational closure in [8], we attribute the poor performance of our formula-based representations to the difficulty of concisely expressing cardinalities syntactically in propositional logic. We use the disjunction of knowledge base subsets of size k to express the notion of 'satisfying at least k formulas' (despite being a rather verbose representation).

7 RELATED WORK

In collaboration with Cohen, we propose the foundational algorithmic strategies and model representations for computing rational and lexicographic closure. Cohen details the approaches for rational closure, each of which is refined in the proposed count-based lexicographic closure algorithms in this paper.

The count-based lexicographic ordering used to define the form of lexicographic closure in this paper is first introduced in [3]. It is part of a broader framework of rational defeasible entailment relations represented by specific knowledge base ranked models that constitute refinements of rational closure.

Lehmann first defined lexicographic closure using the natural ordering of violation tuples consisting of ordered formula base rank counts [12]. While we show that the ordering defined by [3] differs from that of Lehmann, it is still important to note the similarities between the two orderings. It is also important to note that the ranked model derived from the Lehmann ordering is part of the rational defeasible entailment framework in [3].

Scalability of the syntax-based approaches for computing rational closure and the Lehmann lexicographic closure are explored in [8].

8 CONCLUSIONS

We defined three new algorithms for constructing abstract representations of the count-based lexicographic modular partial ordering defined in [3]. The first of these constructs precisely the ranked model corresponding to this ordering via the direct manipulation of valuations. The second algorithms represents the rank valuations syntactically. Similarly, the third approach represents the cumulative rank valuations using more concise syntax.

We formulate a straightforward algorithm, based on the definition for ranked interpretation satisfaction, for computing the entailment relation associated with a ranked interpretation. We adapt this approach to be compatible with both of our formula-based representations.

In attempting to formulate a cumulative algorithm for lexicographic closure, corresponding to the original formula-based approaches, we find that the ordering defined in [3] for lexicographic closure differs from that originally defined in [12]. While both constitute refinements of the rational closure model, they represent distinct forms of rational defeasible entailment [3].

Analysing the theoretical performance of our approaches, we find that all suffer from intractable space complexities attributable to the difficulty of enforcing the count-based refinement criterion of the ordering. While we initially aimed to compare these results to existing approaches, showing potential improvements, the newly-discovered difference in lexicographic orderings means that our results represent a baseline for the count-based lexicographic closure in [3].

9 FUTURE WORK

In light of the distinction made between the two lexicographic orderings in this paper, further exploration of the count-based ordering is required. The philosophical pattern of reasoning, if any, to which this ordering corresponds, is not yet known. Additionally, exploration of its cumulative properties may facilitate the development of a formula-based algorithm and an associated cumulative model-based algorithm similar to `LexicographicCumulativeFormulaRank`.

Future work could include the development of similar semantic algorithms for computing the Lehmann lexicographic closure, as was the original aim of this paper. We would expect corresponding algorithms for the Lehmann ordering to mirror the findings by Cohen for rational closure, in particular the relationship between the cumulative model and `LexicographicClosure` algorithm.

Finally, there is room to explore whether these algorithms, those developed for rational closure by Cohen, and their corresponding model representations, may be generalised for the purposes of computing any rational defeasible entailment relation [3].

10 ACKNOWLEDGEMENTS

This paper would not have been possible without our supervisor, Professor Thomas Meyer, whom we thank for his guidance and insight in conducting this research.

We also express gratitude to our research partner, Jaron Cohen, with whom we collaborated to produce much of the work in our two papers.

REFERENCES

[1] Mordechai Ben-Ari. 2012. *Propositional Logic: Formulas, Models, Tableaux*. Springer London, London, 1, 7–47.

[2] Richard Booth, Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2015. On the Entailment Problem for a Logic of Typicality. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 2805–2811.

[3] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2019. Taking Defeasible Entailment Beyond Rational Closure. In *Logics in Artificial Intelligence*. Springer International Publishing, Cham, 182–197.

[4] Lloyd Everett, Emily Morris, and Tommie Meyer. 2022. *Explanation for KLM-Style Defeasible Reasoning* (1551 ed.). Springer, Cham. https://doi.org/10.1007/978-3-030-95070-5_13

[5] Dov M. Gabbay. 1985. Theoretical Foundations for Non-Monotonic Reasoning in Expert Systems. In *Logics and Models of Concurrent Systems*, Krzysztof R. Apt (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 439–457.

[6] L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence* 226 (2015), 1–33. https://doi.org/10.1016/j.artint.2015.05.001

[7] Crina Grosan and Ajith Abraham. 2011. *Knowledge Representation and Reasoning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 131–147. https://doi.org/10.1007/978-3-642-21004-4_6

[8] Joel Hamilton, Joonsoo Park, Aidan Bailey, and Thomas Meyer. 2022. *An Investigation into the Scalability of Defeasible Reasoning Algorithms*. SACAIR 2021 Organising Committee, Online, 235–251. https://protect-za.mimecast.com/s/OFYSCpgo02fL1l9gtDHUkY

[9] Adam Kaliski. 2020. *An Overview of KLM-Style Defeasible Entailment*. Master's thesis. Faculty of Science, University of Cape Town, Rondebosch, Cape Town, 7700.

[10] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44, 1 (1990), 167–207. https://doi.org/10.1016/0004-3702(90)90101-5

[11] Daniel Le Berre and Anne Parrain. 2010. The SAT4J library, Release 2.2, System Description. *Journal on Satisfiability, Boolean Modeling and Computation* 7 (2010), 59–64. https://doi.org/10.3233/SAT190075

[12] Daniel Lehmann. 1999. Another perspective on Default Reasoning. *Annals of Mathematics and Artificial Intelligence* 15 (11 1999). https://doi.org/10.1007/BF01535841

[13] Daniel Lehmann and Menachem Magidor. 1992. What does a conditional knowledge base entail? *Artificial Intelligence* 55, 1 (1992), 1–60. https://doi.org/10.1016/0004-3702(92)90041-U

[14] Hector J. Levesque. 1986. Knowledge Representation and Reasoning. *Annual Review of Computer Science* 1, 1 (1986), 255–287. https://doi.org/10.1146/annurev.cs.01.060186.001351 arXiv:https://doi.org/10.1146/annurev.cs.01.060186.001351

[15] Raymond Reiter. 1980. A logic for default reasoning. *Artificial Intelligence* 13, 1 (1980), 81–132. https://doi.org/10.1016/0004-3702(80)90014-4 Special Issue on Non-Monotonic Logic.

[16] Yoav Shoham. 1987. Nonmonotonic Logics: Meaning and Utility. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence - Volume 1* (Milan, Italy) *(IJCAI'87)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 388–393.

[17] Yoav Shoham. 1987. *A Semantical Approach to Nonmonotonic Logics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 227–250.

[18] Matthias Thimm. 2017. The Tweety Library Collection for Logical Aspects of Artificial Intelligence and Knowledge Representation. *Künstliche Intelligenz* 31, 1 (March 2017), 93–97.

A `LEXICOGRAPHICMODELRANK` PROOFS

Proposition A.1. *The* `LexicographicModelRank` *algorithm terminates.*

Proof. The outermost while loop executes exactly n times and should not affect termination. Therefore, termination will depend entirely on whether the inner while loop terminates for each value of i.

For any $i < n$:
Since $\{L_{ij} \mid 0 \leq j \leq \#\mathcal{K}\} \setminus \{\emptyset\}$ partitions $\mathscr{U}_{i0}$, $\bigcup_{j=0}^{\#\mathcal{K}} L_{ij} = \mathscr{U}_{i0} = R_i^{RC}$.
Now, the algorithm recursively defines $\mathscr{U}_{ij}$ as $\mathscr{U}_{i(j-1)} \setminus L_{i(j-1)}$, resulting in the following derivation:

$$\begin{aligned} \mathscr{U}_{ij} &= \mathscr{U}_{i0} \setminus L_{i0} \setminus ... \setminus L_{i(j-1)} \\ \implies \mathscr{U}_{ij} &= \mathscr{U}_{i0} \setminus \bigcup_{k=0}^{j-1} L_{ik} \\ \implies \mathscr{U}_{ij} &= R_i^{RC} \setminus \bigcup_{k=0}^{j-1} L_{ik} \end{aligned}$$

But for $j = \#\mathcal{K} + 1$, we have $\bigcup_{k=0}^{\#\mathcal{K}} L_{ik} = \mathscr{U}_{i0} = R_i^{RC}$, since the L_{ik}'s partition the rank.
Thus, $\mathscr{U}_{ij} = R_i^{RC} \setminus R_i^{RC} = \emptyset$, the required condition for termination of the inner loop.
Therefore, we have that the innermost loop will terminate after at most $\#\mathcal{K} + 1$ iterations, for each value of i, and hence the algorithm terminates. □

Proposition A.2. *The* `LexicographicModelRank` *algorithm produces the lexicographic [3] ranked model of* $\mathcal{K}$*.*

Proof. Suppose `LexicographicModelRank` produces
$\mathcal{R}^* = (R_0, ..., R_{n-1}, R_\infty)$.

We will prove the above in two parts:

(1) $\mathcal{R}^*$ is a ranked interpretation:
We show that all worlds are assigned a unique rank, and that there are no empty ranks in the model.

We have that $\mathcal{R}_{\mathcal{K}}^{RC} = (R_0^{RC}, ..., R_{n-1}^{RC}, R_\infty^{RC})$ produced by `ModelRank` is a ranked interpretation.
Consider $u \in R_i^{RC}$:
There is some j such that $u \in L_{ij}$, since $\bigcup_{k=0}^{\#\mathcal{K}} L_{ik} = R_i^{RC}$.
Since $L_{ij} \neq \emptyset$, there is some k such that $R_k = L_{ij}$ and hence $\mathcal{R}^*(u) = k$.
This rank is unique since $\nexists L_{i'j'} : u \in L_{i'j'}, i' \neq i \textit{ or } j' \neq j$.
This follows from the fact that the rational closure ranks partition $\mathscr{U}$ and each R_i^{RC} is partitioned by the L_{ij}'s (ignoring potentially empty L_{ij}'s).
And so, $\nexists k' : \mathcal{R}^*(u) = k' \textit{ and } k' \neq k$, since $R_k = L_{ij}$.

Consider R_i for some i:
$\exists j, k : R_i = L_{jk} \textit{ and } L_{jk} \neq \emptyset$, by construction, and such L_{jk}'s are placed consecutively.
Therefore, there cannot be an empty rank in the interpretation, which is sufficient in satisfying the required convexity property of ranked interpretations.

(2) $\mathcal{R}^*$ conforms to the lexicographic ordering [3] defined on $\mathcal{K}$:
We consider the 3 cases in the defined ordering: $m \prec_{LC}^{\mathcal{K}} n$ if and only if $\mathcal{R}_{RC}^{\mathcal{K}}(n) = \infty$, or $\mathcal{R}_{RC}^{\mathcal{K}}(m) < \mathcal{R}_{RC}^{\mathcal{K}}(n)$, or $\mathcal{R}_{RC}^{\mathcal{K}}(m) = \mathcal{R}_{RC}^{\mathcal{K}}(n)$ and m satisfies more formulas than n in $\mathcal{K}$.

Consider arbitrary $u, v \in \mathscr{U}$:

(a) $\mathcal{R}_{RC}^{\mathcal{K}}(v) = \infty$:
Since $R_\infty = R_\infty^{RC}$, $\mathcal{R}^*(v) = \infty$, and hence $u \prec_{\mathcal{R}^*} v$.

(b) $\mathcal{R}_{RC}^{\mathcal{K}}(u) < \mathcal{R}_{RC}^{\mathcal{K}}(v)$:
Then $u \in L_{ij}$ and $v \in L_{kl}$ for some i, j, k, l such that $i < j$. Since $R_m = L_{ij}$ and $R_n = L_{kl}$ for some $m < n$, we have that

$R^*(u) < R^*(v)$ and therefore than $u \prec_{\mathcal{R}^*} v$.

(c) $\mathcal{R}_{RC}^{\mathcal{K}}(u) = \mathcal{R}_{RC}^{\mathcal{K}}(v)$ and u satisfies more formulas than v in $\mathcal{K}$:
Let $i = \mathcal{R}_{RC}^{\mathcal{K}}(u) = \mathcal{R}_{RC}^{\mathcal{K}}(v)$. Then, $u \in L_{ij}$ and $v \in L_{ik}$ for some $j < k$, since u satisfies more formulas and hence violates fewer formulas than v in $\mathcal{K}$. Since $R_m = L_{ij}$ and $R_n = L_{ik}$ with $j < k$, we have $m < n$, and hence that $R^*(u) < R^*(v)$ and $u \prec_{\mathcal{R}^*} v$.

We now have that $\prec_{\mathcal{R}^*}$ satisfies all the properties of the lexicographic closure modular ordering, and since it is a ranked interpretation, it must be the unique ranked interpretation obeying such an ordering. From [3], we know that the ranked interpretation corresponding to lexicographic closure is a model of $\mathcal{K}$, and hence $\mathcal{R}^*$ is the lexicographic ranked model of $\mathcal{K}$, as defined by the ordering in [3].

□

B LEXICOGRAPHICFORMULARANK PROOFS

Proposition B.1. *For each rank L'_k in the output of the* `LexicographicFormulaRank` *algorithm,* $\mathrm{Mod}(L'_k) = L_k$ *where L_k is the corresponding rank in the output of the* `LexicographicModelRank` *algorithm, with both algorithms returning the same number of ranks.*

Proof. We will first show, inductively, that for each refined rank L'_{ij} in the `LexicographicFormulaRank` algorithm, for any arbitrary i, is such that $Mod(L'_{ij}) = L_{ij}$ where L_{ij} is defined in the `LexicographicModelRank` algorithm, and similarly, that $Mod(\mathscr{U}'_{ij}) = \mathscr{U}_{ij}$. We also show that $\mathscr{U}'_{ij}$ is defined if and only if $\mathscr{U}_{ij}$ is defined.

We first note that $Mod(L'_\infty) = Mod(F_\infty^{RC}) = R_\infty^{RC} = L_\infty$ (we explicitly assign the infinite rank in both algorithms, ensuring correspondence).

Let $i < n$ be any finite rank in any rational closure model with $n-1$ finite ranks.
Base Case:

(1) $Mod(\mathscr{U}'_{i0}) = Mod(F_i^{RC}) = R_i^{RC} = \mathscr{U}_{i0}$

(2) $\mathscr{U}_{i0} \neq \emptyset$, since the rational closure ranks are non-empty, therefore $\mathscr{U}_{i1}$ and L_{i0} will be defined. Similarly, $\mathscr{U}'_{i0} \not\models \bot$, since $\mathscr{U}'_{i0} \not\models \bot \iff Mod(\mathscr{U}'_{i0}) = \mathscr{U}_{i0} \neq \emptyset$. Therefore, $\mathscr{U}'_{i1}$ and L'_{i0} will be defined.

(3)

$$\begin{aligned}
Mod(L'_{i0}) &= Mod\left(F_i^{RC} \wedge \left(\bigvee_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}} - 0\}} \bigwedge_{s \in S} s\right)\right) \\
&= Mod(F_i^{RC}) \cap \bigcup_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}}\}} Mod(S) \\
&= R_i^{RC} \cap Mod(\overrightarrow{\mathcal{K}}) (\text{the only subset of size } \#\overrightarrow{\mathcal{K}} \text{ is } \overrightarrow{\mathcal{K}}) \\
&= \mathscr{U}_{i0} \cap Mod(\overrightarrow{\mathcal{K}}) \\
&= \{u \in \mathscr{U}_{i0} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \not\Vdash k\} = 0\} \\
&= L_{i0}
\end{aligned}$$

Inductive Step:
Assume for some j such that L_{ij}, L'_{ij} and $\mathscr{U}_{ij}, \mathscr{U}'_{ij}$ are defined, that $L_{ij} = Mod(L'_{ij})$ and $\mathscr{U}_{ij} = Mod(\mathscr{U}'_{ij}) \neq \emptyset$.

(1)

$$\begin{aligned}
Mod(\mathscr{U}'_{i(j+1)}) &= Mod(\mathscr{U}'_{ij} \wedge \neg L'_{ij}) \\
&= Mod(\mathscr{U}'_{ij}) \cap \overline{Mod(L'_{ij})} \\
&= \mathscr{U}_{ij} \setminus L_{ij} \\
&= \mathscr{U}_{i(j+1)}
\end{aligned}$$

(2) Now,

$$\begin{aligned}\mathscr{U}_{i(j+1)} = \emptyset &\iff Mod(\mathscr{U}'_{i(j+1)}) = \emptyset \\ &\iff \mathscr{U}'_{i(j+1)} \models \bot\end{aligned}$$

Therefore,

$$\begin{aligned}L_{i(j+1)} \text{ is defined } &\iff \mathscr{U}_{i(j+1)} \neq \emptyset \\ &\iff \mathscr{U}'_{i(j+1)} \not\models \bot \\ &\iff L'_{i(j+1)} \text{ is defined .}\end{aligned}$$

(3) If $\mathscr{U}_{i(j+1)} = \emptyset$, we are done (both $L_{i(j+1)}$ and $L'_{i(j+1)}$ will not be defined, with L_{ij}, L'_{ij} the last defined ranks for the refinement of rational closure rank i.
Else, $\mathscr{U}_{i(j+1)} \neq \emptyset$ and so $L_{i(j+1)}, L'_{i(j+1)}$ are defined.

$$\begin{aligned}Mod(L'_{i(j+1)}) &= Mod\left(\mathscr{U}'_{i(j+1)} \wedge \left(\bigvee_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}} - (j+1)\}} \bigwedge_{s \in S} s\right)\right) \\ &= Mod\left(\mathscr{U}'_{ij} \wedge \neg L'_{ij} \wedge \left(\bigvee_{S \in \{T \subseteq \overrightarrow{\mathcal{K}} \mid \#T = \#\overrightarrow{\mathcal{K}} - (j+1)\}} \bigwedge_{s \in S} s\right)\right) \\ &= Mod(\mathscr{U}'_{ij}) \cap \overline{Mod(L'_{ij})} \cap \{u \in \mathscr{U} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \Vdash k\} \geq \#\overrightarrow{\mathcal{K}} - (j+1)\} \\ &= \mathscr{U}_{ij} \cap \mathscr{U} \setminus L_{ij} \cap \{u \in \mathscr{U} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \nVdash k\} \leq j+1\} \\ &= \mathscr{U}_{ij} \setminus L_{ij} \cap \{u \in \mathscr{U} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \nVdash k\} \leq j+1\} \\ &= \mathscr{U}_{i(j+1)} \cap \{u \in \mathscr{U} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \nVdash k\} \leq j+1\} \\ &= \{u \in \mathscr{U}_{i(j+1)} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \nVdash k\} \leq j+1\} \\ &= \{u \in \mathscr{U}_{i(j+1)} \mid \#\{k \in \overrightarrow{\mathcal{K}} \mid u \nVdash k\} = j+1\} \\ &= L_{i(j+1)}\end{aligned}$$

Thus, by induction, the refined ranks in each algorithm correspond as required.

Using this result, since

$$\begin{aligned}L_k = L_{ij} &\iff L_{ij} \neq \emptyset \\ &\iff L'_{ij} \not\models \bot \\ &\iff L'_k = L'_{ij}\end{aligned}$$

we must have that $\forall k \leq n, Mod(L'_k) = L_k$.

□

C EXAMPLES

We use our kiwi example to illustrate the output of each of the three construction algorithms.

Recall we have $\mathcal{K} := \{k \rightarrow b, b \mid\sim f, b \mid\sim w, k \rightarrow \neg f\}$.

C.1 LexicographicModelRank

The ranked model produced by `ModelRank` is:

∞	$\overline{b}fkw$ $\overline{bf}kw$ $\overline{b}fk\overline{w}$ $\overline{bf}k\overline{w}$ $bfkw$ $bfk\overline{w}$
1	$b\overline{f}kw$ $b\overline{fk}w$ $b\overline{f}k\overline{w}$ $bf\overline{kw}$ $b\overline{fkw}$
0	$bf\overline{k}w$ $\overline{b}f\overline{k}w$ $\overline{bfk}w$ $\overline{b}f\overline{kw}$ $\overline{bfkw}$

Figure 8: Rational closure model of $\mathcal{K}$

Refining this output, our `LexicographicModelRank` produces:

∞	$\overline{b}fkw$ $\overline{bf}kw$ $\overline{b}fk\overline{w}$ $\overline{bf}k\overline{w}$ $bfkw$ $bfk\overline{w}$
2	$b\overline{f}k\overline{w}$ $b\overline{fkw}$
1	$b\overline{f}kw$ $b\overline{fk}w$ $bf\overline{kw}$
0	$bf\overline{k}w$ $\overline{b}f\overline{k}w$ $\overline{bfk}w$ $\overline{b}f\overline{kw}$ $\overline{bfkw}$

Figure 9: Lexicographic closure model of $\mathcal{K}$

C.2 LexicographicFormulaRank

The formula representation of the ranked model produced by `FormulaRank` is:

∞	$\neg((k \rightarrow b) \wedge (k \rightarrow \neg f))$
1	$((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge \neg(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))) \wedge \top$
0	$((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))$

Figure 10: Formula rational closure model of $\mathcal{K}$

The refined formula representation output by `LexicographicFormulaRank` is:

∞	$\neg((k \rightarrow b) \wedge (k \rightarrow \neg f))$
2	$(\neg(((b \rightarrow f) \vee (b \rightarrow w)) \wedge (((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge \neg(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))) \wedge \top) \wedge \top) \wedge \top) \wedge$ $(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge \neg(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))) \wedge \top) \wedge \top$
1	$((b \rightarrow f) \vee (b \rightarrow w)) \wedge (((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge \neg(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))) \wedge \top) \wedge \top$
0	$(((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))) \wedge \top \wedge ((b \rightarrow f) \wedge (b \rightarrow w))$

Figure 11: Formula lexicographic closure model of $\mathcal{K}$

Taking the models of each of the rank formulas yields the ranked model in figure 9.

C.3 LexicographicCumulativeFormulaRank

The cumulative formula representation of the ranked model produced by `CumulativeFormulaRank` is:

∞	$\top$
1	$((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge \top$
0	$((k \rightarrow b) \wedge (k \rightarrow \neg f)) \wedge ((b \rightarrow f) \wedge (b \rightarrow w))$

Figure 12: Cumulative formula rational closure model of $\mathcal{K}$

The corresponding refined lexicographic cumulative formula model produced by `LexicographicCumulativeFormulaRank` is:

∞	$\top$
2	$(((k \to b) \wedge (k \to \neg f)) \wedge ((b \to f) \wedge (b \to w))) \vee ((((k \to b) \wedge (k \to \neg f)) \wedge \top) \wedge \top)$
1	$(((k \to b) \wedge (k \to \neg f)) \wedge ((b \to f) \wedge (b \to w))) \vee ((((k \to b) \wedge (k \to \neg f)) \wedge \top) \wedge ((b \to f) \vee (b \to w)))$
0	$((k \to b) \wedge (k \to \neg f)) \wedge ((b \to f) \wedge (b \to w)) \vee \bot$

Figure 13: Cumulative formula lexicographic closure model of $\mathcal{K}$

The models of each formula in the representation correspond to the cumulative models from the model in figure 9.