UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

Computer Science Honours
Final Paper
2017

Title: Preferential reasoning for ontologies with a user-defined ranking of defeasible subsumption statements

Author: Reid Swan

Project Abbreviation: PRO

Supervisor(s): Professor Thomas Meyer

Category	Min	Max	Chosen
Requirements Analysis and Design	0	20	0
Theoretical Analysis	0	25	25
Experiment Design and Execution	0	20	0
System Development and Implementation	0	15	0
Results, Findings and Conclusion	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**

Preferential reasoning for ontologies with a user-defined ranking of defeasible subsumption statements

Reid Swan
Computer Science Honours
University of Cape Town
SWNREI001@myuct.ac.za

ABSTRACT

In this paper, we present the idea of a user-defined ranking of statements in the DTBox of a knowledge base for defeasible reasoning. We show that this satisfies the KLM postulates and Rational Monotonicity, powerful and successful logical formalisms for non-monotonic reasoning. These user-defined rankings are characterized in terms of the statements they entail, when they are equivalent, and when ranked statements can be considered redundant and hence removed. Finally, the rankings are related to the rational closure of a conditional knowledge base, where it is proven that these two ideas are essentially equivalent. The results of this paper should enable ontology engineers to more easily model complex scenarios which are not suited to monotonic reasoning, while providing a finer level of control over entailment in a more intuitive manner than is provided by rational closure.

1 INTRODUCTION

Automated reasoning is the process of an algorithm taking knowledge represented in some form and using it to automatically derive new knowledge from it. Description logics are significant and powerful formalism for representing this knowledge, employing some decidable subset of first order logic and using it to express facts. There exist many powerful classical reasoners, including HermiT, Fact++ and Konclude.

Classical reasoners, however, adhere to the monotonicity property, limiting their reasoning power. To this end, we wish to implement a reasoner which can reason non-monotonically. Although non-monotonic reasoners already exist – for example, the Rational Closure-based non-monotonic reasoner implemented by Moodley [9] – they are limited in the amount of control they provide to the ontology engineer: the Rational Closure algorithm automatically generates a fixed ranking, when instead a ranking could be provided by the ontology engineer, allowing for more control and the communication of more information.

This paper represents the theoretical side of an attempt to develop a non-monotonic reasoner which relies on statement rankings provided by the user, wherein we develop algorithms and the theory behind reasoning with a user-defined statement ranking. A related paper by Michael Harrison dicusses the actual implementation details of the project.

The intention of this paper was to determine under which circumstances reasoning with a user-defined ranking adheres to the KLM postulates and Rational Monotonicity, properties desired of any non-monotonic reasoner, originally defined by Kraus, Lehmann and Magidor [6]. In fact, we show that when reasoning with a user-defined ranking, any ranking satisifies the KLM postulates and Rational Monotonicity. From this, we then prove a number of results about these rankings, including when they equivalent, when statements are redundant, and that non-monotonic reasoning with a user-defined ranking is equally as powerful as Rational Closure.

2 BACKGROUND

Description Logics

Description logics are a family of logic languages used to represent knowledge in terms of concepts, roles and individuals. A set of description logic statements form a knowledge base. Using a precise semantics, it is possible for an automated system to use the information represented in the knowledge base to derive conclusions; this is in contrast with a database, where information is simply stored for retrieval, and implicit information is only extracted if it is enriched with extraneous functionality. To paraphrase Baader, et al. [1], description logics are decidable fragments of first order logic used for representing knowledge in a domain so that the knowledge can be reasoned about by an automated system.

In standard, monotonic description logics, a knowledge base is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a set of terminological axioms, known as the TBox, defining concepts, roles, and their relations to one another, and $\mathcal{A}$ is a set of assertional axioms, known as the ABox, defining individuals (atoms) and the concepts to which they belong.

Description logics have a formal set-based semantics which precisely define the meaning of each axiom. Informally, the semantics consists of an interpretation, which is a pair containing the domain Δ and a mapping function $\cdot^{\mathcal{I}}$ which maps individuals to items in the domain, concepts to sets of those items, and roles to relations between these items. A knowledge base is consistent if and only if there exists at least one $(\Delta, \cdot^{\mathcal{I}})$ pair which are consistent with the semantics of the axioms in the knowledge base; such a pair is called a *model* of $\mathcal{K}$.

This semantics defines whether a description logic statement is true based on the axioms contained in the knowledge base; this process is known as entailment. In classical entailment, a description logic statement A is entailed by a knowledge base $\mathcal{K}$, denoted $\mathcal{K} \models A$, if and only if A is true in every model of $\mathcal{K}$.

$\mathcal{ALC}$, or *Attributive Language with Concepts*, is a prototypical description logic first described by Schmidt-Schauß and Smolka [10], and will be the description logic employed throughout this paper. For a thorough treatment of $\mathcal{ALC}$ and entailment, please consult Baader, et al.'s 'An Introduction to Description Logic' [2].

Defeasible Reasoning

Classical entailment in description logics satisfies the monotonicity property: if a statement is entailed by a set of statements, any

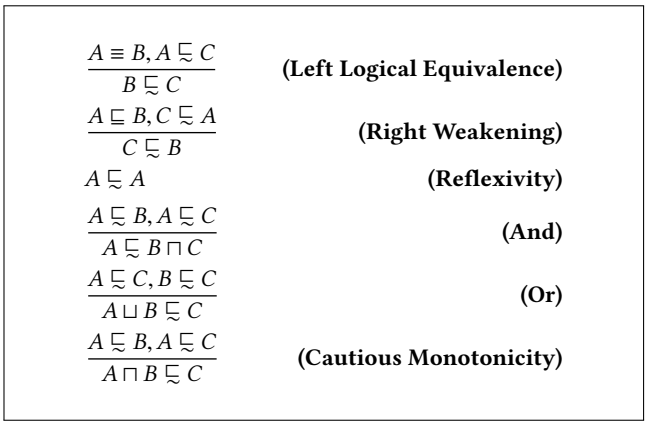

Figure 1: The KLM Postulates

superset of those statements will entail that statement; so it is not possible to withdraw a conclusion when new information is learned. In symbols, if $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ where $\mathcal{T} \subseteq \mathcal{T}'$ and $\mathcal{A} \subseteq \mathcal{A}'$, then for all statements S, $\mathcal{K} \models S \implies \mathcal{K}' \models S$. By contrast, *defeasible* reasoning allows conclusions to be withdrawn in light of new knowledge.

While monotonicity can be a powerful tool in reasoning with knowledge bases, it is also a limitation on the reasoning power of a knowledge base. Consider the following example which demonstrates the limitation of monotonicity: mammals dwell on land (`Mammal` $\sqsubseteq$ `LandDweller`); whales are mammals (`Whale` $\sqsubseteq$ `Mammal`); whales are sea dwellers (`Whale` $\sqsubseteq$ `SeaDwellers`); sea dwellers are not land dwellers (`SeaDwellers` $\sqsubseteq \neg$`LandDwellers`). While a human would be able to revise their knowledge to attain a consistent mental model of this situation, in classical description logics, the concept `Whale` is inconsistent – in other words, an automated reasoner would conclude there are no `Whales`. Due to monotonicity, there is no way to extend this knowledge base to allow for `Whale` to be consistent.

Non-monotonic reasoning allows us to reason with such a system in a rigorous, formally-defined manner, thereby extending the power of automatic reasoners and artificial intelligence systems. It is therefore desirable to enrich description logics with the ability to reason defeasibly. To this end, a number of new systems of logic or extensions to existing systems of logic have been proposed.

The KLM Postulates. One such system is the *preferential* reasoning of Kraus, Lehmann and Magidor [6]; initially proposed as an extension to propositional logic, it has since been extended to description logics. This paper takes a number of notational conventions and semantic constructs from the extension proposed by Britz, et al., [3]; a similar extension was proposed by Casini and Straccia, [5].

Kraus, et al., [6] define a set of postulates, which have come to be known as the KLM postulates, which they argue should be satisfied by any system of non-monotonic reasoning. These are reproduced in figure 1 using $\mathcal{ALC}$ and $\mathrel{\tilde{\sqsubset}}$, the defeasible subsumption operator introduced by Britz, et al., [3], where $A \mathrel{\tilde{\sqsubset}} B$ can be understood to mean 'an A is *typically* a B'. In addition to these KLM postulates, they also propose Rational Monotonicity:

$$\frac{A \sqcap B \not\mathrel{\tilde{\sqsubset}} C, A \not\mathrel{\tilde{\sqsubset}} \neg B}{A \not\mathrel{\tilde{\sqsubset}} C} \quad \textbf{(Rational Monotonicity)}$$

Kraus, et al., argue that Rational Monotonicity is a desirable property of any reasonable non-monotonic entailment procedure. Rational Monotonicity can be understood as follows: if something that is an A and a B is not typically a C, and A is not typically *not* a B, then we should not conclude that A is typically a C. This is a reasonable conclusion to draw – we do not believe that the typical A is not a B, and we also do not believe that the typical A-and-B is not a C, so because it is possible that a typical A is also a B, we cannot conclude that it is a C. For example, `Animal` $\sqcap \exists$`eats.Plant` $\not\mathrel{\tilde{\sqsubset}}$ `Predator` (an animal that eats plants is not typically a predator) and `Animal` $\not\mathrel{\tilde{\sqsubset}} \neg \exists$`eats.Plant` (an animal is not typically not a plant-eater), then `Animal` $\not\mathrel{\tilde{\sqsubset}}$ `Predator` (an animal is not typically a predator), because a typical `Animal` may be a plant-eater, which is not typically a `Predator`.

A consequence relation that satisfies the KLM postulates Left Logical Equivalence to Cautious Monotonicity is said to be a preference relation. For it to be rational, it must additionally satisfy Rational Monotonicity.

Lehmann and Magidor [7] additionally define the notion of *rational closure*, an entailment procedure which can be thought of as the minimum rational preferential relation in the sense that any other rational preferential consequence relation must entail some superset of the set of statements entailed by rational closure.

Defeasible Description Logics. A conditional knowledge base, $\mathcal{K}$, is a pair $(\mathcal{T}, \mathcal{D})$ consisting of a TBox $\mathcal{T}$ of terminological axioms, which are strict facts, expressed in pure $\mathcal{ALC}$ notation, and a DTBox $\mathcal{D}$ of defeasible terminological axioms, using $\mathcal{ALC}$ with the defeasible subsumption operator ($\mathrel{\tilde{\sqsubset}}$) replacing the classical subsumption operator ($\sqsubseteq$), which contains non-strict statements that may be revoked when reasoning defeasibly.

Casini, et al., [4] present a rational closure algorithm that is equivalent to that presented by Lehmann and Magidor, but for the context of defeasible description logics. The algorithm for `RationalClosure` is reproduced from [4] in Appendix A. Using this definition of `RationalClosure`, Casini, et al. [5] prove that defeasible subsumption with entailment defined by `RationalClosure` is a rational preference relation.

User-defined ranking

Algorithm 1, `RationalClosure` (appendix A), has been shown to represent a rational preference relation for defeasible description logics. By using algorithm 3, `ComputeRanking`, it automatically generates a ranking of the statements in $\mathcal{D}$ from a conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. This is only one of many possible rankings of statements in $\mathcal{D}$. An ontology engineer may have some additional knowledge of the situation being modelled that can be communicated by defining her own ranking of the DTBox statements. It would be beneficial to be able to take an arbitrary, user-defined ranking of DTBox statements and determine an entailment procedure that satisfies the KLM postulates and Rational Monotonicity. If such a defeasible entailment procedure is found, what are some properties of these rankings that can be exploited?

The aim of this paper is to determine under which circumstances a given ranking represents a rational preference relation, and to derive some of the properties of these rankings in general.

3 RELATED WORK

Kody Moodley [9] discusses the theory and implementation of a Rational Closure-based reasoner as an extension of the Protégé ontology editor. Also discussed is preferential reasoning, as well as a number of alternative non-monotonic logic systems.

Kraus, Lehmann and Magidor [6] introduce preferential reasoning and rational consequence relations, defining the KLM postulates and Rational Monotonicity and proving some results about preferential reasoning systems. A sequel paper by Lehmann and Magidor [7] extends this discussion, defining rational closure and an algorithm for computing it.

Britz, et al., [3] define a semantics for defeasible description logics based on the preferential reasoning of Kraus, Lehmann and Magidor. They also present rational closure for defeasible description logics and present results related to it.

A similar paper by Casini and Straccia [5] also presents preferential reasoning in terms of description logics, lifting Rational Closure to description logics and presenting an algorithm for computing rational closure for a defeasible $\mathcal{ALC}$.

For a precise treatment of description logics, Schmidt-Schaußand Smolka [10] introduce the concept language ALC which is used in this paper, and 'An Introduction to Description Logics' by Baader, et al., [2] is an excellent reference for a precise introductory treatment of description logics.

4 PROPERTIES OF RANKINGS

For a given conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, a ranking is a partitioning of the DTBox $\mathcal{D}$ into a number of ranks, each given a non-negative integer index. More precisely:

Definition 4.1 (Ranking). A ranking is a tuple $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ defined relative to a conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ satisfying the following properties:

- for every integer $i \in [0, n]$, $\mathcal{E}_i \subseteq \mathcal{D}$;
- for every integer $i \in [0, n)$, $\mathcal{E}_{i+1} \subseteq \mathcal{E}_i$
- $\mathcal{E}_0 = \mathcal{D}$

A *rank* refers to any of the sets which compose a given ranking $\mathcal{E}$. For all integers $i \in [0, n]$, $\mathcal{E}_i$ is a rank of $\mathcal{E}$. The *size* of $\mathcal{E}$, denoted $|\mathcal{E}|$, is n where $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$.

All of the statements in a ranking $\mathcal{E}$ are defeasible subsumption statements. Casini, et al. [4] define the following two ways in which to translate these to classical $\mathcal{ALC}$ concepts: the *materialization* of $\mathcal{E}$ and the *classical translation* of $\mathcal{E}$.

Definition 4.2 (Materialization). The *materialization* of a rank $\mathcal{E}_i$, denoted $\overline{\mathcal{E}_i}$, is defined as

$$\overline{\mathcal{E}_i} = \{\neg C \sqcup D \mid C \precsim D \in \mathcal{E}_i\}$$

Definition 4.3 (Classical translation). The *classical translation* of a rank $\mathcal{E}_i$, denoted $\mathcal{E}_i^{\sqsubseteq}$, is defined as

$$\mathcal{E}_i^{\sqsubseteq} = \{C \sqsubseteq D \mid C \precsim D \in \mathcal{E}_i\}$$

The reasoning behind classical translation should be quite clear: a set of defeasible subsumption statements are translated to their exact classical counterparts. Materialization may require slightly more motivation. Consider a statement $C \sqsubseteq D$; since C is subsumed by D, $\neg C$ subsumes $\neg D$; so $\top \sqsubseteq \neg D \sqcup D \sqsubseteq \neg C \sqcup D$. For any concept A, taking $(\neg C \sqcup D) \sqcap A$ is equivalent to A. For defeasible subsumption statements, however, it is not necessarily the case that $\top \sqsubseteq \neg C \sqcup D$; but for any concept A, $(\neg C \sqcup D) \sqcap A$ can be considered the same as the concept A if $C \precsim D$ was a strict fact. Similarly, $\bigsqcap \overline{\mathcal{E}_i}$ is the conjunction of the materialization of every concept $C \precsim D \in \mathcal{E}_i$, and the conjunction of $\bigsqcap \overline{\mathcal{E}_i}$ with any concept A is the same as the concept A if the defeasible statements in $\mathcal{E}_i$ were strict facts.

A property of the materialization of ranks of $\mathcal{E}$ is shown in the following lemma.*

Lemma 4.4. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to the conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. For every integer $i \in [0, n)$, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}}$.*

It follows that if $0 \leq j < i \leq n$, then $\bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$.

Algorithm 1, `RationalClosure`, computes an automatic ranking using the procedure of algorithm 3, `ComputeRanking`, which is used with algorithm 2, `IsEntailed`, to determine if a given defeasible subsumption statement is in the Rational Closure of the given knowledge base. However, it is possible to supply an arbitrary ranking $\mathcal{E}$ to `IsEntailed`. The following theorem shows that this entailment procedure on an arbitrary ranking represents a preference relation.

Theorem 4.5. *Given a conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and a ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, using the entailment procedure defined by algorithm 2, $\precsim$ is a preference relation.*

Moreover, under the entailment procedure defined by `IsEntailed` with knowledge base $\mathcal{K}$ and arbitrary ranking $\mathcal{E}$, $\precsim$ satisfies Rational Monotonicity and is a *rational* preference relation.

Theorem 4.6. *Given a conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, using the entailment procedure defined by* `IsEntailed`*, $\precsim$ is a rational preference relation.*

Theorem 4.6 indicates that $\precsim$ is a rational preference relation in general for the entailment procedure defined by `IsEntailed`. An important consequence of this theorem is that when creating a conditional knowledge base, a user can provide further knowledge by providing their own ranking of statements in the knowledge base, and that ranking will still adhere to the desirable property Rational Monotonicity. The user can communicate further information about the system by defining their own ranking, as well as gaining finer-grained control over the entailment procedure, without sacrificing any of the reasoning power desired in a defeasible automatic reasoner.

Now that it has been established that `IsEntailed` represents a rational preference relation with an aribitrary ranking, for purposes of convenience and clarity, we can define the following notation:

$(\mathcal{K}, \mathcal{E}) \models A \precsim B$ if and only if `IsEntailed`$(\mathcal{K}, \mathcal{E}, A \precsim B)$ = `true`

We now present a number of characteristics of rankings. We start by defining some terminology and notations.

*Full proofs and relevant algorithms are present in the appendices

Definition 4.7 (The rk function). Let $\mathcal{C}$ be the set of all possible $\mathcal{ALC}$ concepts and $\mathbb{Z}_0$ be the set of all non-negative integers. Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to a conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. We define the function $rk_{\mathcal{E}} : \mathcal{C} \to \mathbb{Z}_0 \cup \infty$ as follows:

$$rk_{\mathcal{E}}(A) = \begin{cases} \min\{i \mid \mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot\}, & \text{the set is not empty} \\ \infty, & \text{the set is empty} \end{cases}$$

$rk_{\mathcal{E}}(A)$ will also be referred to as the rank of concept A.

`IsEntailed`$(\mathcal{K}, \mathcal{E}, A \sqsubsetneq B)$, can be understood as performing a search for $i = rk_{\mathcal{E}}(A)$ and then checking if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$.

A version of the rk function was originally defined by Lehmann and Magidor [7], and the version given here was defined by Britz, et al., [3]. If $\mathcal{E}$ is understood as the ranking used in `RationalClosure` produced by `ComputeRanking` (algorithms 1 and 3, respectively), then the original definition of rational closure defined by Lehmann and Magidor required that $A \sqsubsetneq B \iff rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$. The following theorem shows that this holds for an arbitrary ranking:

Theorem 4.8. *Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$; then $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$ if and only if $rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$ or $A \sqsubseteq B$.*

Thus it is possible to use the rk function to succinctly prove that a statement $A \sqsubsetneq B$ is entailed by comparing the value of $rk_{\mathcal{E}}(A \sqcap B)$ with $rk_{\mathcal{E}}(A \sqcap \neg B)$. Since $rk_{\mathcal{E}}$ can be a powerful tool in reasoning with rankings, we can further characterize the nature of the $rk_{\mathcal{E}}$ function by observing how it acts on composite concepts. The following relates $rk_{\mathcal{E}}(C)$ for a composite concept C with its subconcepts:

Theorem 4.9. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a conditional knowledge base with ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$. Let j be the smallest integer such that $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_j} \not\sqsubseteq \bot$.*

$$rk_{\mathcal{E}}(A \sqcap B) \geq \max(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B)) \tag{1}$$

$$rk_{\mathcal{E}}(A \sqcup B) = \min(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B)) \tag{2}$$

$$rk_{\mathcal{E}}(\neg A) \begin{cases} = j, & rk_{\mathcal{E}}(A) > j \\ \geq j, & rk_{\mathcal{E}}(A) = j \end{cases} \tag{3}$$

Full proofs of all theorems are in Appendix B, but it may be beneficial to motivate the second case of (3). It is not possible to find a stronger relation $rk_{\mathcal{E}}(\neg A)$ in terms of $rk_{\mathcal{E}}(A)$ if the latter is equal to j, because there exists a ranking in which $rk_{\mathcal{E}}(A) = rk_{\mathcal{E}}(\neg A)$. For example:

Let $\mathcal{K} = (\emptyset, \mathcal{D})$ with ranking $\mathcal{E}$ where $\mathcal{D} = \mathcal{E}_0 = \{A \sqsubsetneq B, \neg A \sqsubsetneq C\}$; then $\overline{\mathcal{E}_0} = \{\neg A \sqcup B, A \sqcup C\}$

$$\begin{aligned} \bigsqcap \overline{\mathcal{E}_0} \equiv & (\neg A \sqcup B) \sqcap (A \sqcup C) \\ \equiv & (\neg A \sqcap (A \sqcup C)) \sqcup (B \sqcap (A \sqcup C)) \\ \equiv & ((\neg A \sqcap A) \sqcup (\neg A \sqcap C)) \sqcup ((B \sqcap A) \sqcup (B \sqcap C)) \\ \equiv & (\bot \sqcup (\neg A \sqcap C)) \sqcup (B \sqcap A) \sqcup (B \sqcap C) \\ \equiv & (\neg A \sqcap C) \sqcup (B \sqcap A) \sqcup (B \sqcap C) \end{aligned}$$

Then $\bigsqcap \overline{\mathcal{E}_0} \sqcap A \equiv (B \sqcap A) \sqcup (B \sqcap C \sqcap A)$ and $\bigsqcap \overline{\mathcal{E}_0} \sqcap \neg A \equiv (\neg A \sqcap C) \sqcup (B \sqcap C \sqcap \neg A)$. Neither of these are subsumed by $\bot$, hence A and $\neg A$ have $rk_{\mathcal{E}}(A) = rk_{\mathcal{E}}(\neg A) = 0$

To demonstrate the power of the $rk_{\mathcal{E}}$ function, we can use theorems 4.8 and 4.9 to prove the following result:

Proposition 4.10. *Let $\mathcal{K}$ be a conditional knowledge base with ranking $\mathcal{E}$. If $rk_{\mathcal{E}}(A) < rk_{\mathcal{E}}(\neg B)$ then $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$*

Proof. Since $rk_{\mathcal{E}}(A) < rk_{\mathcal{E}}(\neg B)$, we must have that the smallest j such that $\mathcal{T} \bigsqcap \overline{\mathcal{E}_j} \not\sqsubseteq \bot$ is less than $rk_{\mathcal{E}}(\neg B)$, so by theorem 4.9(3), $rk_{\mathcal{E}}(B) \leq rk_{\mathcal{E}}(A)$.

Let $i = rk_{\mathcal{E}}(A)$; by theorem 4.9,

$$rk_{\mathcal{E}}(A \sqcap \neg B) \geq \max(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(\neg B)) > rk_{\mathcal{E}}(A) = i$$

so $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap \neg B \sqsubseteq \bot$, but $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$, so it follows that $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq \bot$. So $rk_{\mathcal{E}}(A \sqcap B) \leq i$, and $rk_{\mathcal{E}}(A \sqcap \neg B) > i$, hence $rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$. By theorem 4.8, $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$. □

5 RANKING EQUIVALENCE

Now that we have some results characterizing a single ranking, we may wish to compare two rankings. To that end, this section investigates when two rankings entail the same set of statements, which we shall call *equivalent* rankings.

Results about the equivalence of rankings are important, because as per Lutz [8], entailment checking in $\mathcal{ALC}$ is ExpTime-complete, and since the algorithms for defeasible entailment used in this paper (in particular, `IsEntailed`) reduce to $\mathcal{ALC}$ entailment checks, reducing the number of entailment checks required can have a significant impact on the run time of these algorithms. These results allow us to find knowledge bases and rankings that are equivalent in terms of the statements entailed but with a smaller size, allowing faster run times without sacrificing reasoning power.

Definition 5.1 (Equivalent). Two rankings $\mathcal{E}, \mathcal{F}$, relative to knowledge bases $\mathcal{K}$ and $\mathcal{K}'$ respectively, are *equivalent*, denoted $\mathcal{E} \equiv \mathcal{F}$, iff $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$ if and only if $(\mathcal{K}', \mathcal{F}) \models A \sqsubsetneq B$.

It is easy to see that $\equiv$ is an equivalence relation: `IsEntailed` is a deterministic algorithm, so for a given $(\mathcal{K}, \mathcal{E}, A \sqsubsetneq B)$, `IsEntailed` always returns the same result; additionally, 'if and only if' is an equivalence relation. Together, this results in $\equiv$ being an equivalence relation.

Equivalence can be related to the rk functions of $\mathcal{E}$ and $\mathcal{F}$ by the following theorem:

Theorem 5.2. *Let $\mathcal{E}$ and $\mathcal{F}$ be two rankings relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}', \mathcal{D}')$. If $\mathcal{E} \equiv \mathcal{F}$ then*

1. *$rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$ if and only if $rk_{\mathcal{F}}(A \sqcap B) < rk_{\mathcal{F}}(A \sqcap \neg B)$*
2. *$rk_{\mathcal{E}}(A \sqcap B) = rk_{\mathcal{E}}(A \sqcap \neg B)$ if and only if $rk_{\mathcal{F}}(A \sqcap B) = rk_{\mathcal{F}}(A \sqcap \neg B)$*

Lemmas 5.4 and 5.5 allow us to conditionally 'shrink' a ranking in order to produce a ranking which is smaller but equivalent. For notational convenience, we define 'subrankings', borrowing syntax from the Python programming language.

Definition 5.3 (Subranking). Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. For $0 \leq i \leq j \leq n$, an i-to-j subranking of $\mathcal{E}$, denoted $\mathcal{E}[i : j]$, is $(\mathcal{E}_i, \mathcal{E}_{i+1}, ..., \mathcal{E}_j)$.

Note that by definition, a ranking $\mathcal{E}$ must have $\mathcal{E}_0 = \mathcal{D}$, because we want every statement in $\mathcal{D}$ to be ranked in $\mathcal{E}$. But for any $i \geq 1$, $\mathcal{E}[i:j]$ has $\mathcal{E}[i:j]_0 = \mathcal{E}_i \neq \mathcal{D}$, so it is a ranking relative to a different knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{E}_i)$.

Lemma 5.4. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a conditional knowledge base with the statements in $\mathcal{D}$ ranked into $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$. If there is some $i \in [0..n]$ for which $\mathcal{T} \models \bigcap \overline{\mathcal{E}_i} \sqsubseteq \bot$ then $\mathcal{E} \equiv \mathcal{E}[i+1:n]$, where $\mathcal{E}[i+1:n]$ is the subranking of $\mathcal{E}$ relative to conditional knowledge base $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ where $\mathcal{D}' = \bigcup_{j=i+1}^{n} \mathcal{E}_j$*

Lemma 5.5. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the smallest integer such that $\bigcap \overline{\mathcal{E}_i} \equiv \top$. Then $\mathcal{E}' \equiv \mathcal{E}$, where $\mathcal{E}' = \mathcal{E}[0:i-1]$.*

These lemmas allow for ranks to be dropped from the top or bottom of a ranking without effecting the set of entailed statements, but what about dropping statements from the middle of the ranking? This is discussed further in section 6, but the following definition allows us to provide some initial results:

Definition 5.6 (Dense ranking). Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking; $\mathcal{E}$ is *dense* iff for all $0 \leq i < n$, $\bigcap \overline{\mathcal{E}_i} \not\equiv \bigcap \overline{\mathcal{E}_{i+1}}$.

Dense rankings are significant because every rank of a dense ranking can be involved in entailment. By contrast, a ranking which is not dense (a *sparse* ranking) has the property that for some $\mathcal{E}_i$, $\mathcal{E}_{i+1}$, $\bigcap \overline{\mathcal{E}_i} \equiv \bigcap \overline{\mathcal{E}_{i+1}}$. Then for all concepts A, $\bigcap \overline{\mathcal{E}_i} \sqcap A \equiv \bigcap \overline{\mathcal{E}_{i+1}} \sqcap A$; so A is non-empty at i if and only if it is non-empty at $i+1$. Because of this, when using algorithm 2, `IsEntailed`, to check if some $A \subsetneq B$ is defeasibly entailed by $(\mathcal{K}, \mathcal{E})$, if $\bigcap \overline{\mathcal{E}_i} \sqcap A$ is non-empty then $\bigcap \overline{\mathcal{E}_{i+1}} \sqcap A$ is non-empty, but the algorithm will stop at i, so $\mathcal{E}_{i+1}$ will not be used; and if $\bigcap \overline{\mathcal{E}_i} \sqcap A$ is empty, then $\bigcap \overline{\mathcal{E}_{i+1}} \sqcap A$ is also empty, so it will be passed over for the next highest rank. So $\mathcal{E}_{i+1}$ never contributes to entailment checking.

This can be thought of as a 'gap' in the ranking – an empty space that does not contribute to entailment in a significant way. The terms 'sparse' and 'dense' are chosen because they relay the idea of the presence or absence of these 'gaps'. These gaps add to the total number of iterations of the while loop in algorithm 2, increasing running time without representing any unique knowledge. We would therefore prefer to work with dense rankings.

To this end, algorithm 5, `CollapseRanking`, takes a knowledge base $\mathcal{K}$ and ranking $\mathcal{E}$ and returns a ranking $\mathcal{E}'$ of the same statements such that $\mathcal{E}'$ is dense and $\mathcal{E}' \equiv \mathcal{E}$.

It is clear that the algorithm will terminate for all inputs as long as a ranking is finite, since i is incremented on every iteration of the while loop, and the loop ends when $i > n$.

Theorem 5.7. *The ranking $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$ is dense and equivalent to $\mathcal{E}$.*

Corollary 5.8. *$\mathcal{E} \equiv \mathcal{F}$ if and only if $\texttt{CollapseRanking}(\mathcal{E}) \equiv \texttt{CollapseRanking}(\mathcal{F})$.*

Corollary 5.9. *Every finite ranking $\mathcal{E}$ has an equivalent dense ranking.*

Another motivation for working with dense rankings is the result of theorem 5.11, which provides the strong and surprising result that if two dense ranks are equivalent, then the conjunction of the materialization of each of their ranks must be equivalent.

One requirement of the proof of theorem 5.11 is that the top rank of the rankings under consideration not have the conjunction of its materialization equivalent to $\top$; but by lemma 5.5, it is always possible to transform a ranking into an equivalent ranking for which this is true.

Lemma 5.10 is split from theorem 5.11 because it allows us to prove a slighter stronger result without some of the restrictions of 5.11, namely the requirement of dense rankings. Also, it enables us to reduce the length of theorem 5.11's proof.

Lemma 5.10. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be rankings relative to the same knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ with $m = n$ such that for all integers i with $0 \leq i \leq n$, $\bigcap \overline{\mathcal{E}_i} \equiv \bigcap \overline{\mathcal{F}_i}$; then $\mathcal{E} \equiv \mathcal{F}$.*

Theorem 5.11. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be dense rankings relative to the same knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ such that $\bigcap \overline{\mathcal{E}_m} \not\equiv \top$ and $\bigcap \overline{\mathcal{F}_n} \not\equiv \top$[†]; then $\mathcal{E} \equiv \mathcal{F}$ if and only if $m = n$ and for all integers i such that $0 \leq i \leq n$, $\bigcap \overline{\mathcal{E}_i} \equiv \bigcap \overline{\mathcal{F}_i}$.*

It may be noted that theorem 5.11 requires that $\mathcal{E}$ and $\mathcal{F}$ be rankings relative to the same conditional knowledge base, but that equivalence does not require this in general. What, then, can be said about dense equivalent rankings with different knowledge bases? In fact, it requires only a generalization of the base case of the proof of theorem 5.11 to yield the following theorem.

Theorem 5.12. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be dense rankings relative to the knowledge bases $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ respectively such that $\bigcap \overline{\mathcal{E}_0} \not\equiv \bot$ and $\bigcap \overline{\mathcal{F}_0} \not\equiv \bot$ and $\bigcap \overline{\mathcal{E}_m} \not\equiv \top$ and $\bigcap \overline{\mathcal{F}_n} \not\equiv \top$; then $\mathcal{E} \equiv \mathcal{F}$ if and only if $m = n$ and for all integers i such that $0 \leq i \leq n$, $\bigcap \overline{\mathcal{E}_i} \equiv \bigcap \overline{\mathcal{F}_i}$.*

The results of this section on equivalence can be summarize by the following observation, which also serves to generalize the results to arbitrary (i.e., dense or sparse) rankings.

Let $\mathcal{E}$ and $\mathcal{F}$ be two rankings of statements relative to conditional knowledge bases $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ respectively. Let $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$, $\mathcal{F}' = \texttt{CollapseRanking}(\mathcal{K}', \mathcal{F})$. Let $n = |\mathcal{E}'|$ and $m = |\mathcal{F}'|$. We assign e_l, e_h, f_l, f_h as follows:

$$e_l = \begin{cases} 0, & \bigcap \overline{\mathcal{E}'_0} \not\equiv \bot \\ 1, & \bigcap \overline{\mathcal{E}'_0} \equiv \bot \end{cases} \quad e_h = \begin{cases} n, & \bigcap \overline{\mathcal{E}'_n} \not\equiv \top \\ n-1, & \bigcap \overline{\mathcal{E}'_n} \equiv \top \end{cases}$$

$$f_l = \begin{cases} 0, & \bigcap \overline{\mathcal{F}'_0} \not\equiv \bot \\ 1, & \bigcap \overline{\mathcal{F}'_0} \equiv \bot \end{cases} \quad f_h = \begin{cases} n, & \bigcap \overline{\mathcal{F}'_m} \not\equiv \top \\ n-1, & \bigcap \overline{\mathcal{F}'_m} \equiv \top \end{cases}$$

Let $\mathcal{E}'' = \mathcal{E}'[e_l : e_h]$ and $\mathcal{F}'' = \mathcal{F}'[f_l : f_h]$. Finally, let $n' = |\mathcal{E}''|$ and $m' = |\mathcal{F}''|$. Then we can prove the following:

Corollary 5.13. *$\mathcal{E} \equiv \mathcal{F}$ if and only if $n' = m'$ and for all integers i such that $0 \leq i \leq n$, $\bigcap \overline{\mathcal{E}''_i} \equiv \bigcap \overline{\mathcal{F}''_i}$.*

Using this, it is possible to define an algorithm which checks equivalence between ranks in general. This is presented as algorithm 6, `IsEquivalent`.

[†]This is not a limiting requirement because by lemma 5.5, if $\bigcap \overline{\mathcal{E}_m} \equiv \top$, we can attain an equivalent ranking by removing the rank $\mathcal{E}_m$, and similarly for $\mathcal{F}_n$.

6 STATEMENT REDUNDANCY

Redundant statements refer to those statements which, when added to or removed from a knowledge base and ranking, do not effect the set of entailed statements. Naturally, redundancy is related to ranking equivalence, because we require that, on adding or removing redundant statements, the new and old rankings must be equivalent. In this section, redundancy is discussed. In order to do so, two forms of redundancy – namely, local and total redundancy – are presented and related to one another.

Definition 6.1 (Locally redundant). Let $\mathcal{E}$ be a ranking of defeasible subsumption statements in a DTBox $\mathcal{D}$. Let $C \sqsubseteq D$ be some statement in $\mathcal{D}$. Then $C \sqsubseteq D$ is *locally redundant* at i if $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_i'}$ where $\mathcal{E}_i' = \mathcal{E}_i \backslash \{C \sqsubseteq D\}$.

It follows trivially from the definition that if $C \sqsubseteq D$ is locally redundant at i and $\mathcal{E}_i' = \mathcal{E}_i \backslash \{C \sqsubseteq D\}$, then for all concepts A, $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_i'} \sqcap A$; so for any concept B, $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$ iff $\bigsqcap \overline{\mathcal{E}_i'} \sqcap A \sqsubseteq B$.

What does local redundancy at i imply about $i-1$?

Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, and $C \sqsubseteq D$ locally redundant at i. If $i = 0$ then there is no $\mathcal{E}_{i-1}$ so we assume $i > 0$. Let $R = \{A \sqsubseteq B | A \sqsubseteq B \in \mathcal{E}_{i-1}$ and $A \sqsubseteq B \notin \mathcal{E}_i\}$. Note that $C \sqsubseteq D \in \mathcal{E}_i$ and $\mathcal{E}_i \subseteq \mathcal{E}_{i-1}$, so $C \sqsubseteq D \notin R$.

$\mathcal{E}_i \cup R = \mathcal{E}_{i-1}$, so $\bigsqcap \overline{\mathcal{E}_i} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$. By assumption, $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_i'}$, so $\bigsqcap \overline{\mathcal{E}_i'} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_i} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$. But $\bigsqcap \overline{\mathcal{E}_i'} \sqcap \bigsqcap \overline{R}$ is the conjunction of the materialization of every $A \sqsubseteq B \in \mathcal{E}_{i-1}$ except $C \sqsubseteq D$, so it is equivalent to $\bigsqcap \overline{\mathcal{E}_{i-1} \backslash \{C \sqsubseteq D\}}$. Then $\bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \bigsqcap \overline{\mathcal{E}_{i-1} \backslash \{C \sqsubseteq D\}}$, so $C \sqsubseteq D$ is locally redundant at $i-1$.

A simple inductive argument yields:

Theorem 6.2. *If $C \sqsubseteq D$ is locally redundant at i then $C \sqsubseteq D$ is locally redundant at j for all $0 \leq j \leq i$.*

What does local redundancy at i imply about $i+1$?

Local redundancy at i does not imply local redundancy at $i+1$, because there exists a ranking for which $C \sqsubseteq D$ is locally redundant at i but not at $i+1$.

For example, let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ where $\mathcal{T} = \{A \sqsubseteq \neg C, A \sqsubseteq D, B \sqsubseteq C \sqcap \neg D\}$, $\mathcal{D} = \{C \sqsubseteq D, \top \sqsubseteq A\}$ and the ranking $\mathcal{E}$ is as follows:

$\mathcal{E}_0$	$C \sqsubseteq D, \top \sqsubseteq A$
$\mathcal{E}_1$	$C \sqsubseteq D$

$$\begin{aligned}\bigsqcap \overline{\mathcal{E}_0} =& (\neg C \sqcup D) \sqcap (\neg \top \sqcup A)\\ \equiv& (\neg C \sqcup D) \sqcap A\\ \equiv& (\neg C \sqcap A) \sqcup (A \sqcap D)\\ \equiv& A \sqcup A\\ \equiv& A\end{aligned}$$

$$\begin{aligned}\bigsqcap \overline{\mathcal{E}_0 \backslash \{C \sqsubseteq D\}} =& (\neg \top \sqcup A)\\ \equiv& A\end{aligned}$$

Hence $\bigsqcap \overline{\mathcal{E}_0} \equiv \bigsqcap \overline{\mathcal{E}_0 \backslash \{C \sqsubseteq D\}}$, so $C \sqsubseteq D$ is locally redundant at 0.

However, $C \sqsubseteq D$ is not locally redundant at 1 because it is the only statement, and $\mathcal{T} \not\models \neg C \sqcup D \equiv \top$[‡].

So local redundancy at i does not imply local redundancy at $i+1$.

Another form of statement redundancy that we can consider is total redundancy:

Definition 6.3 (Totally redundant). For a ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ and conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, a statement $C \sqsubseteq D$ is *totally redundant* if and only if $\mathcal{E} \equiv \mathcal{E}'$, where $\mathcal{E}' = (\mathcal{E}_0 \backslash \{C \sqsubseteq D\}, ..., \mathcal{E}_n \backslash \{C \sqsubseteq D\})$ relative to $\mathcal{K} = (\mathcal{T}, \mathcal{D} \backslash \{C \sqsubseteq D\})$

Now we have these two disparate definitions of statement redundancy. Can they be related, and if so, how? The following theorem shows that for dense rankings, total redundancy can be defined entirely in terms of local redundancy.

Theorem 6.4. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a dense ranking relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the largest integer such that $C \sqsubseteq D \in \mathcal{E}_i$. Then $C \sqsubseteq D$ is totally redundant if and only if it is locally redundant at i.*

7 RELATING RATIONAL CLOSURE TO RANKINGS

One may wonder whether using an arbitrary ranking is more powerful than rational closure. Clearly rankings are at least as powerful as rational closure, since setting a ranking equal to the output of `ComputeRanking` is equivalent to the procedure employed by rational closure. Is the converse true, or is it possible to create a ranking which cannot be simulated by the rational closure of some conditional knowledge base?

The following theorem shows that this is not the case; in other words, for an arbitrary ranking $\mathcal{E}$ relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, it is always possible to create a knowledge base $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ such that $\mathcal{E} \equiv \mathcal{E}'$ where $\mathcal{E}' = \texttt{ComputeRanking}(\mathcal{K}')$. Since `RationalClosure` uses the ranking output by `ComputeRanking`, we have for all statements $C \sqsubseteq D$, $\texttt{RationalClosure}(\mathcal{K}', C \sqsubseteq D) = \texttt{IsEntailed}(\mathcal{K}, \mathcal{E}, C \sqsubseteq D)$.

Theorem 7.1. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $\mathcal{D}' = \{\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}\} \cup \{\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \leq i < n\}$. Let $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ and $\mathcal{E}' = \texttt{ComputeRanking}(\mathcal{K}')$. Then $\mathcal{E} \equiv \mathcal{E}'$.*

The proof of this uses the following lemmas:

Lemma 7.2. $\bigsqcap \overline{\mathcal{E}_0'} \equiv \bigsqcap \overline{\mathcal{D}'} \equiv \bigsqcap \overline{\mathcal{E}_0}$.

Lemma 7.3. *For all $i \in [1, n]$,*

$$\bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \leq j < n\}} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$$

Lemma 7.4. *For all integers $i \in [0, n]$,*

$$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}_i') = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \leq j < n\}$$

8 COMPUTATIONAL COMPLEXITY CONSIDERATIONS

In this section, we briefly discuss the order of complexity of the algorithms presented in appendix A.

[‡] $\neg(\neg C \sqcup D) \equiv C \sqcap \neg D \sqsupseteq B$ and B is not necessarily empty

`RationalClosure`

Algorithm 1, `RationalClosure`, and its subroutines, algorithms 2, 3 and 4 are all adapted from those presented by Casini, et al. [4]. They note that as long as the underlying entailment procedure has at least polynomial complexity, `RationalClosure` will have the same complexity class as that entailment procedure. This paper uses the description logic $\mathcal{ALC}$. Per Lutz [8], entailment in $\mathcal{ALC}$ is ExpTime-complete, so the `RationalClosure` used in this paper is in the complexity class ExpTime.

`CollapseRanking`

Algorithm 5, `CollapseRanking`, is dominated in running time by the while loop of lines 8 to 15. i starts the while loop equal to 1, and the loop increments i on every iteration, and stops when $i > n$; so the loop will run exactly n times.

Aside from the entailment check on line 9, the while loop only performs a constant number of $O(1)$ assignments. Per Lutz [8], $\mathcal{ALC}$ entailment checking is ExpTime-complete.

We therefore have a polynomial number of ExpTime-complete entailment checks, so we conclude that `CollapseRanking` is in the complexity class ExpTime.

`Equivalent`

Algorithm 6, `Equivalence`, can be considered in terms of the initialization of the variables in lines 4 to 22, and the core of the algorithm in lines 23 to 31.

Lines 4 to 22 contain 6 ExpTime statements on lines 4, 5, 8, 11, 14 and 17; the remainder of the lines contain $O(1)$ or $O(n)$ computations. In total, then, these lines are dominated by the ExpTime calls, of which there are a constant number. We conclude that this section takes worst case ExpTime.

Lines 23 to 31 contain an if statement followed by a for loop. In the worst case, the if statement fails and the for loop runs. The if statement represents a simple numerical equality check, so runs in $O(n)$ time. This section is therefore dominated by the for loop of lines 26 to 27. Clearly, the worst case is when all $n' + 1$ loop iterations run. Each iteration has 1 $\mathcal{ALC}$ entailment check, which, per Lutz [8], is ExpTime-complete. Since $n' <= n + m$ which is directly proportional to the length of the input, we can see that there are only polynomially many such entailment checks before the algorithm completes. Hence we conclude that this section has ExpTime running time.

Together, this indicates that the algorithm is in the ExpTime complexity class.

Although these results may seem discouraging, it should be noted that while the Rational Closure algorithm for $\mathcal{ALC}$ has ExpTime worst case running time, Casini, et al., [4] note that the real world running time of Rational Closure is much better than ExpTime would indicate due to the optimizations applied to the underlying reasoners. A similar result may hold for the algorithms presented here, and as such should be investigated.

9 CONCLUSION

In this paper, we presented the idea of allowing a user to define a ranking of statements in the DTBox of a knowledge base for defeasible reasoning. This was proven to be a reasonable proposal, since using the entailment procedure defined by `IsEntailed`, it satisfies the KLM postulates and rational monotonicity, a set of conditionals for non-monotonic entailment which have had significant success both in propositional logic and description logics. The paper than proceeded to characterize these user-defined rankings by investigating their properties, in terms of the statements they entail. Strong results about the equivalence of rankings were presented. We also defined when a statement can be considered redundant and can be removed without effecting entailment, and presented results relating to exactly when that is the case for a given statement. Finally, the discussion was rounded off by proving that user-defined rankings are equivalent to rational closure in that they can simulate one another, before the computational complexity classes of algorithms presented in this paper was indicated.

Future work

The results in this paper can be expanded upon in a number of ways. Primarily, the results in this paper should be extended to the case of reasoning with conditional knowledge bases that include assertional axioms (ABoxes). The results can also be generalized to description logics other than $\mathcal{ALC}$, in particular $\mathcal{SROIQ}$, which is a highly popular description logic for the semantic web. The results on statement redundancy can be extended to multiple redundant statements - in particular, when can multiple redundant statements be removed, and how do we determine the largest set of redundant statements that can be removed together without effecting the set of entailed statements? Section 7 related rankings to rational closure for dense rankings - can this be extended to sparse rankings? Is there a more natural construction for doing so? Can we do the same for other forms of closure, for example, lexicographic closure?

REFERENCES

[1] Franz Baader, Diego Calvanese, Deborah L McGuinness, Daniele Nardi, and Peter F Patel-Schneider. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Vol. 32. 622 pages.

[2] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. 2017. *An Introduction to Description Logic*. Cambridge University Press.

[3] Katarina Britz, Thomas Meyer, and Ivan Varzinczak. 2011. Semantic foundation for preferential description logics. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 491–500.

[4] Giovanni Casini, Thomas Meyer, Kody Moodley, Uli Sattler, and Ivan Varzinczak. 2015. Introducing defeasibility into OWL ontologies. In *International Semantic Web Conference*. Springer, 409–426.

[5] Giovanni Casini and Umberto Straccia. 2010. Rational closure for defeasible description logics. In *European Workshop on Logics in Artificial Intelligence*. Springer, 77–90.

[6] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44, 1-2 (1990), 167–207.

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

[8] Carsten Lutz. 2008. The complexity of conjunctive query answering in expressive description logics. *Automated Reasoning* (2008), 179–193.

[9] Kody Moodley. 2015. *Practical Reasoning For Defeasible Description Logics*. Ph.D. Dissertation. University of KwaZulu-Natal.

[10] Manfred Schmidt-Schauß and Gert Smolka. 1991. Attributive concept descriptions with complements. *Artificial intelligence* 48, 1 (1991), 1–26.

APPENDIX A ALGORITHMS

`RationalClosure` returns `true` if and only if a defeasible subsumption statement $C \sqsubseteq\!\!\!\sim D$ is in the Rational Closure of the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Adapted from Casini, et al. [4].

Algorithm 1 RationalClosure

procedure RationalClosure
  Input: $\mathcal{K} = (\mathcal{T}^*, \mathcal{D}^*)$, query $C \sqsubseteq\!\!\!\sim D$
  Output: `true` iff $C \sqsubseteq\!\!\!\sim D$ is in the Rational Closure of $\mathcal{K}$
  $\mathcal{E} = \mathit{ComputeRanking}(\mathcal{K})$
  return IsEntailed$(\mathcal{K}, \mathcal{E}, C \sqsubseteq\!\!\!\sim D)$
end procedure

`IsEntailed` is a function which returns `true` if and only if a defeasible subsumption statement $C \sqsubseteq\!\!\!\sim D$ is defeasibly entailed by conditional knowledge base $\mathcal{K}$ with ranking $\mathcal{E}$. Adapted from Casini, et al. [4].

Algorithm 2 IsEntailed

procedure IsEntailed
  Input: $\mathcal{K} = (\mathcal{T}^*, \mathcal{D}^*)$, ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, query $C \sqsubseteq\!\!\!\sim D$
  Output: `true` iff $C \sqsubseteq\!\!\!\sim D$ is defeasibly entailed by $\mathcal{K}$
  i := 0
  while $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$ and $i \leq n$ **do**
    $i := i + 1$
  end while
  if $i \leq n$ **then**
    return $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq D$
  else
    return $\mathcal{T}^* \models C \sqsubseteq D$
  end if
end procedure

`ComputeRanking` is a helper function of `RationalClosure` used to generate an automatic ranking of statements for use defeasible entailment checking. Adapted from Casini, et al. [4].

`Exceptional` is a helper function of `ComputeRanking` which, given a knowledge base $\mathcal{K}$ and a set D of defeasible subsumption statements, returns the subset of D' of D which is exceptional for D. Adapted from Casini, et al. [4].

`CollapseRanking` takes a ranking $\mathcal{E}$ and its related knowledge base $\mathcal{K}$ and returns a ranking $\mathcal{E}'$ relative to the same knowledge base which is equivalent to $\mathcal{E}$ and is dense.

`Equivalent`$(\mathcal{K}, \mathcal{E}, \mathcal{K}', \mathcal{F}')$ returns `true` if and only if $\mathcal{E} \equiv \mathcal{F}$; the correctness of this follows from corollary 5.13.

Algorithm 3 ComputeRanking

procedure ComputeRanking
  Input: $\mathcal{K} = (\mathcal{T}^*, \mathcal{D}^*)$
  Output: ranking $\mathcal{E}$ of statements in $\mathcal{D}$.
  $\mathcal{T}' := \mathcal{T}$, $\mathcal{D}' = \mathcal{D}$, $\mathcal{E} = \emptyset$, $\mathcal{D}'_\infty := \mathcal{D}$
  while $\mathcal{D}'_\infty \neq \emptyset$ **do**
    i := 0
    $\mathcal{E}_0 = \mathcal{D}'$
    $\mathcal{E}_1$ = Exceptional$(\mathcal{T}', \mathcal{E}_0)$
    while $\mathcal{E}_{i+1} \neq \mathcal{E}_i$ **do**
      $i := i + 1$
      $\mathcal{E}_{i+1}$:= Exceptional$(\mathcal{T}', \mathcal{E}_i)$
    end while
    $\mathcal{D}'_\infty := \mathcal{E}_i$
    $\mathcal{T}' := \mathcal{T}' \cup \{C \sqsubseteq D \mid C \sqsubseteq\!\!\!\sim D \in \mathcal{D}'_\infty\}$
    $\mathcal{D}' := \mathcal{D}' \backslash \mathcal{D}'_\infty$
  end while
  return $\mathcal{E}$
end procedure

Algorithm 4 Exceptional

procedure Exceptional
  Input: $\mathcal{K} = (\mathcal{T}^*, \mathcal{D}^*), D \subseteq \mathcal{D}$
  Output: $D' \subseteq D$ such that D' is exceptional w.r.t. D
  for $A \sqsubseteq\!\!\!\sim B \in D$ **do**
    if $\mathcal{T} \models \bigsqcap \overline{D} \sqsubseteq \neg A$ **then**
      $D' := D' \cup \{A \sqsubseteq\!\!\!\sim B\}$
    end if
  end for
  return D'
end procedure

Algorithm 5 CollapseRanking

procedure CollapseRanking$(\mathcal{K}, \mathcal{E})$
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, finite ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$
  Output: $\mathcal{E}'$ where $\mathcal{E}'$ is dense and equivalent to $\mathcal{E}$.
  $\mathcal{E}'_0 = \mathcal{E}_0$
  $i := 1$
  $j := 1$
  prev := $\bigsqcap \overline{\mathcal{E}_0}$
  while $i \leq n$ **do**
    if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \not\equiv$ prev **then**
      $\mathcal{E}'_j := \mathcal{E}_i$
      prev := $\bigsqcap \overline{\mathcal{E}_i}$
      $j := j + 1$
    end if
    $i := i + 1$
  end while
  return $\mathcal{E}'$
end procedure

Algorithm 6 Equivalent

procedure EQUIVALENT($\mathcal{K}, \mathcal{E}, \mathcal{K}', \mathcal{F}$)
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ with finite ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ with finite ranking $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$
  Output: `true` if and only if $\mathcal{E} \equiv \mathcal{F}$
  $\mathcal{E}' :=$ `CollapseRanking`$(\mathcal{K}, \mathcal{E})$
  $\mathcal{F}' :=$ `CollapseRanking`$(\mathcal{K}, \mathcal{F})$
  $n := |\mathcal{E}'|$, $m := |\mathcal{F}'|$
  $e_l := 0$, $e_h := n$, $f_l = 0$, $f_h = m$
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_0} \equiv \bot$ **then**
    $e_l := e_l + 1$
  end if
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_0} \equiv \bot$ **then**
    $f_l := f_l + 1$
  end if
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_n} \equiv \top$ **then**
    $e_h := e_h - 1$
  end if
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_m} \equiv \top$ **then**
    $f_h := f_h - 1$
  end if
  $\mathcal{E}'' := \mathcal{E}'[e_l : e_h]$
  $\mathcal{F}'' := \mathcal{F}'[f_l : f_h]$
  $n' := |\mathcal{E}''|$, $m' = |\mathcal{F}''|$
  if $n' \neq m'$ **then**
    return `false`
  end if
  for i from 0 to n' **do**
    if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}''} \not\equiv \bigsqcap \overline{\mathcal{F}''}$ **then**
      return `false`
    end if
  end for
  return `true`
end procedure

APPENDIX B PROOFS

Lemma 4.4: $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}}$.

Proof. Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. By definition, $\mathcal{E}_{i+1} \subseteq \mathcal{E}_i$. Let $E = \mathcal{E}_i \backslash \mathcal{E}_{i+1}$. Then $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap \bigsqcap \overline{E}$. By monotonicity, we can conclude that $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}}$ □

Theorem 4.5: for any ranking $\mathcal{E}$, under the entailment procedure defined by algorithm 2, `IsEntailed`$(\mathcal{K}, \mathcal{E}, A \sqsubsetneq B)$, $\sqsubsetneq$ is a preference relation.

By definition, an entailment procedure defines a preference relation if and only if it satisfies the KLM postulates of section 2. We therefore show that under the algorithm `IsEntailed`, $\sqsubsetneq$ satisfies these postulates.

Left logical equivalence:

$$\frac{A \equiv B, A \sqsubsetneq C}{B \sqsubsetneq C}$$

Proof.

Case (1):

There is some $i \leq n$ such that $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. Then since $A \sqsubsetneq C$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$. Since $\mathcal{T}^* \models A \equiv B$, by classical entailment we can replace every occurrence of A with B and conclude that $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap B \sqsubseteq \bot$ and $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap B \sqsubseteq C$, and so $B \sqsubsetneq C$.

Case (2):

There is no $i \leq n$ such that $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. Since $\mathcal{T}^* \models A \equiv B$, the same must be true for B. Then since $A \sqsubsetneq C$, $\mathcal{T}^* \models A \sqsubseteq C$. Using $\mathcal{T}^* \models A \equiv B$, we can conclude by classical entailment that $\mathcal{T}^* \models B \sqsubseteq C$, and so $B \sqsubsetneq C$. □

Right weakening:

$$\frac{A \sqsubseteq B, C \sqsubsetneq A}{C \sqsubsetneq B}$$

Proof.

Case (1):

There is some $i \leq n$ such that $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$, and since $C \sqsubsetneq A$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq A$. $\mathcal{T}^* \models A \sqsubseteq B$, so by the monotonicity of classical entailment, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq A \sqsubseteq B$ and so $C \sqsubsetneq B$.

Case (2):

There is no $i \leq n$ such that $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$. Since $C \sqsubsetneq A$, it must be the case that $\mathcal{T}^* \models C \sqsubseteq A$, and by classical entailment monotonicity, $\mathcal{T}^* \models C \sqsubseteq A \sqsubseteq B$, and so $C \sqsubsetneq B$. □

Reflexivity

$$A \sqsubsetneq A$$

Proof. If for all $i \leq n$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$, then the algorithm returns `true` iff $\mathcal{T}^* \models A \sqsubseteq A$ which is trivially true. If there is some $i \leq n$ for which $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$ then the algorithm returns `true` iff $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq A$, which is true by the monotonicity of classical entailment. □

And

$$\frac{A \sqsubsetneq B, A \sqsubsetneq C}{A \sqsubsetneq B \sqcap C}$$

Proof.

Case (1):

There is some $i \leq n$ for which $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$; this is the same i for all three defeasible subsumption statements since they all have the form $A \sqsubsetneq X$ for some X. From this we can conclude that $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$ and $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$. Combining these, we get

$$\mathcal{T}^* \models (\bigsqcap \overline{\mathcal{E}_i} \sqcap A) \sqcap (\bigsqcap \overline{\mathcal{E}_i} \sqcap A) \sqsubseteq B \sqcap C$$
$$\implies \mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B \sqcap C$$

Thus $A \sqsubsetneq B \sqcap C$.

Case (2):

For all $i \leq n$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. Since $A \sqsubsetneq B$ and $A \sqsubsetneq C$ we have $\mathcal{T}^* \models A \sqsubseteq B$ and $\mathcal{T}^* \models A \sqsubseteq C$. $\mathcal{T}^* \models A \sqsubseteq B \sqcap C$ follows from classical entailment and so $A \sqsubsetneq B \sqcap C$. □

Or

$$\frac{A \sqsubsetneq C, B \sqsubsetneq C}{A \sqcup B \sqsubsetneq C}$$

Proof.

Case (1):

Suppose for both A and B, for all $i \leq n$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$ and $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap B \sqsubseteq \bot$. This must also be the case for $A \sqcup B$, since

$$\begin{aligned}\mathcal{T}^* \models & \bigsqcap \overline{\mathcal{E}_i} \sqcap (A \sqcup B) \\ \equiv & (\bigsqcap \overline{\mathcal{E}_i} \sqcap A) \sqcup (\bigsqcap \overline{\mathcal{E}_i} \sqcap B) \\ \sqsubseteq & (\bot) \sqcup (\bot) \\ \equiv & \bot\end{aligned}$$

Then $\mathcal{T}^* \models A \sqsubseteq C$ and $\mathcal{T}^* \models B \sqsubseteq C$, which together give $\mathcal{T}^* \models A \sqcup B \sqsubseteq C$.

Case (2):

For A, there is some $i \leq n$ for which $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$ and for B, for all $i \leq n$ $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap B \sqcap \bot$.

Then $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$ and $\mathcal{T}^* \models B \sqsubseteq C$. $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap (A \sqcup B) \sqsubseteq \bot$ because $\overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$. So we check if $\mathcal{T}^* \models \overline{\mathcal{E}_i} \sqcap (A \sqcup B) \sqsubseteq C$. This is `true` because $\mathcal{T}^* \models \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$ and $\mathcal{T}^* \models \overline{\mathcal{E}_i} \sqcap B \sqsubseteq C$, so if we take the union we get

$$\begin{aligned}\mathcal{T}^* \models & \overline{\mathcal{E}_i} \sqcap (A \sqcup B) \\ \equiv & (\bigsqcap \overline{\mathcal{E}_i} \sqcap A) \sqcap (\bigsqcap \overline{\mathcal{E}_i} \sqcap B) \\ \sqsubseteq & C \sqcap \bot \\ \equiv & C\end{aligned}$$

and so $A \sqcup B \sqsubsetneq C$. (Note that switching A and B in the previous proof will have no effect of the result.)

Case (3):

Suppose that for some $i \leq n$ and for some $j \leq n$, $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$ and $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_j} \sqcap B \sqsubseteq \bot$. Let $k = \min(i, j)$. Then $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_k} \sqcap (A \sqcup B) \sqsubseteq \bot$. By assumption, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$ and $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_j} \sqcap B \sqsubseteq C$. For $\bigsqcap \overline{\mathcal{E}_k}$, either $i = j = k$ and so $\bigsqcap \overline{\mathcal{E}_k} \sqcap (A \sqcup B) \equiv$

$(\bigsqcap \overline{\mathcal{E}_i} \sqcap A) \sqcup (\bigsqcap \overline{\mathcal{E}_j} \sqcap B) \sqsubseteq C$, or for some $X \in \{A, B\}$, $\bigsqcap \overline{\mathcal{E}_k} \sqcap X \sqsubseteq \bot$ and so $\bigsqcap \overline{\mathcal{E}_k} \sqcap (A \sqcup B) \sqsubseteq \bot \sqcup C \equiv C$. In both cases we conclude $A \sqcup B \precsim C$. □

Cautious Monotonicity

$$\frac{A \precsim B, A \precsim C}{A \sqcap B \precsim C}$$

Proof.

Case (1):

Suppose for all $i \leq n$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. Then $\mathcal{T}^* \models A \sqsubseteq B$ and $A \sqsubseteq C$. $\mathcal{T} \models \bigsqcap A \sqsubseteq \bot$, so $\mathcal{T} \models \bigsqcap A \sqcap B \sqsubseteq \bot$. We therefore check if $\mathcal{T} \models \bigsqcap A \sqcap B \sqsubseteq C$. This follows from the monotonicity of classical entailment.

Case (2):

Suppose there is some $i \leq n$ for which $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubset \bot$. Then by assumption, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$ and $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubset C$.

From

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot \text{ and } \mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubset B$$

and we can derive that

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq \bot$$

so the algorithm returns `true` iff

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq C$$

. But $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$ and by the monotonicity of classical entailment,

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq \bigsqcap \overline{\mathcal{E}_i} \sqcap A$$

and so

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq C$$

Thus $A \sqcap B \precsim C$. □

Theorem 4.6: Under the entailment procedure defined by `IsEntailed` with a knowledge base $\mathcal{K}, \mathcal{E}$, $\precsim$ is a rational preference relation.

Proof. By definition, a rational preference relation is a preference relation that satisifes Rational Monotonicity. By theorem 4.5, $\precsim$ is a preference relation under the entailment procedure `IsEntailed`; it remains to show that it satisfies Rational Monotonicity:

$$\frac{A \sqcap B \not\precsim C, A \not\precsim \neg B}{A \not\precsim C}$$

Consider the contrapositive of Rational Monotonicity:

$$\frac{A \precsim C}{A \sqcap B \precsim C \textbf{ or } A \precsim \neg B}$$

Suppose that $A \precsim C$ and $A \not\precsim \neg B$.

Claim: $A \precsim C \wedge A \not\precsim \neg B \implies A \sqcap B \precsim C$

Proof. There are two cases to consider:

Case (1): There is some $i \leq n$ for which $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. By assumption, $A \precsim C$ and $A \not\precsim \neg B$, so

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$$

and

$$\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \neg B$$
$$\iff \mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \neg B$$

From the second line, since $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ is not empty and is not subsumed by $\neg B$, the intersection of $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ with B is non-empty, hence

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq \bot$$

From the monotonicity of classical entailment, $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B$ is subsumed by $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ which is subsumed by C, so we conclude that

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq C$$

and so $A \sqcap B \precsim C$.

Case (2): For all $i \leq n$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$, so

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq \bot \sqcap B \equiv \bot$$

Thus for the query $A \sqcap B \precsim C$ the rational closure algorithm returns `true` iff $\mathcal{T}^* \models A \sqcap B \sqsubseteq C$. By assumption, $A \precsim C$, so

$$\mathcal{T}^* \models A \sqsubseteq C$$

and

$$A \sqcap B \sqsubseteq A \sqsubseteq C$$

so $A \sqcap B \precsim C$. □

Now suppose that $A \precsim C$ but $A \sqcap B \not\precsim C$. Again there are two cases.

Case (1): For all $i \leq n$, $\mathcal{T}^* \models \bigsqcap \overline{E_i} \sqcap A \sqsubseteq \bot$. Then $\mathcal{T}^* \models \bigsqcap \overline{E_i} \sqcap A \sqcap B \sqsubseteq \bigsqcap \overline{E_i} \sqcap A \sqsubseteq \bot$. In the rational closure algorithm, the query $A \precsim \neg B$ reduces to $\mathcal{T} \models A \sqsubseteq \neg B$. From the assumptions, we have

$$\mathcal{T}^* \models A \sqcap B \not\sqsubseteq C$$
$$\mathcal{T}^* \models A \sqsubseteq C$$

This indicates a TBox inconsistency.

Case (2): There is some minimal $i \leq n$ for which $\mathcal{T}^* \not\models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$. Since we assume $A \precsim C$, we have that $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$. There are two subcases of this case:

Case (2.1): $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq \bot$. By assumption,

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$$

but when $\sqcap$'d with B, this is subsumed by $\bot$. So all of $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ is "outside of" B, which means it is "inside" $\neg B$. Thus $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \neg B$, and so $A \precsim \neg B$.

Case (2.2): $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq \bot$. Since i was chosen to be the smallest value for which $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$, for any $j < i$, $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq \bot$ and so $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq \bot$. Since $A \sqcap B \not\precsim C$, the rational closure algorithm has determined that

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq C$$

But we also have $A \precsim C$, which means

$$\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq C$$

This indicates a TBox inconsistency. □

Theorem 4.8: Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$; then $(\mathcal{K}, \mathcal{E}) \models A \subsetneq B$ if and only if $rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$.

Proof. ($X \implies Y$) Suppose we have that $A \subsetneq B$ and that there is some minimal $i \in [0..n]$ such that

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$$

By assumption, $A \subsetneq B$, so

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$$

which means

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \not\sqsubseteq \bot$$

Furthermore,

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqcap \neg B \sqsubseteq \bot$$

Clearly then $rk(A \sqcap B) < rk(A \sqcap \neg B)$.

If we suppose instead that there is no $i \in [0..n]$ for which $\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$, then $A \subsetneq B$ iff $\mathcal{T}^* \models A \sqsubseteq B$, i.e. it follows purely from strict facts.

($\neg X \implies \neg Y$) Suppose that $A \not\subsetneq B$ and that there is some minimal $i \in [0..n]$ such that

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$$

By assumption, $A \not\subsetneq B$, so

$$\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq B$$
$$\implies \mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqcap \neg B \not\sqsubseteq \bot$$

From this last line we can conclude that $rk(A \sqcap \neg B) \leq i$. Since we chose i minimally, at values of $j < i$, $\bigcap \mathcal{E}_j \sqcap A \sqsubseteq \bot$ which implies $\bigcap \mathcal{E}_j \sqcap A \sqcap \neg B \sqsubseteq \bot$, hence $rk(A \sqcap \neg B) = i$. It also implies $\bigcap \mathcal{E}_j \sqcap A \sqcap B \sqsubseteq \bot$ for $j < i$, hence $rk(A \sqcap B) \geq i$. Taken together, $rk(A \sqcap \neg B) \leq rk(A \sqcap B)$.

Alternatively, if there is no $i \in [0..n]$ for which $\mathcal{T}^* \models \bigcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$ then $A \not\subsetneq B$ implies $\mathcal{T}^* \models A \not\sqsubseteq B$ as required. □

Theorem 4.9: Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a conditional knowledge base with ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$. Let j be the smallest integer such that $\mathcal{T} \models \bigcap \overline{\mathcal{E}_j} \not\sqsubseteq \bot$.

$$rk_{\mathcal{E}}(A \sqcap B) \geq \max(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B)) \quad (1)$$
$$rk_{\mathcal{E}}(A \sqcup B) = \min(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B)) \quad (2)$$
$$rk_{\mathcal{E}}(\neg A) \begin{cases} = j, & rk_{\mathcal{E}}(A) > j \\ \geq j, & rk_{\mathcal{E}}(A) = j \end{cases} \quad (3)$$

Proof. In all of the following, assume knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ with ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$; let $j \in [0, n]$ be the smallest integer such that $\bigcap \overline{\mathcal{E}_j} \not\sqsubseteq \bot$.

(1) Let A and B be concepts with $rk_{\mathcal{E}}(A) = a$ and $rk_{\mathcal{E}}(B) = b$. This implies that for all $i < a$,

$$\mathcal{T} \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$$

and for all $j < b$,

$$\mathcal{T} \models \bigcap \overline{\mathcal{E}_j} \sqcap B \sqsubseteq \bot$$

Assume without loss of generality that $a \leq b$; then for all $i < b$,

$$\mathcal{T} \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqcap B \sqsubseteq \bigcap \overline{\mathcal{E}_i} \sqcap B \sqsubseteq \bot$$

So $rk_{\mathcal{E}}(A \sqcap B) \geq \max(a, b)$.

(2) As above $rk_{\mathcal{E}}(A) = a$, $rk_{\mathcal{E}}(B) = b$. So $\mathcal{T} \models \bigcap \overline{\mathcal{E}_a} \sqcap A \not\sqsubseteq \bot$ and $\mathcal{T} \models \bigcap \overline{\mathcal{E}_b} \sqcap B \not\sqsubseteq \bot$

For concept A,

$$\bigcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bigcap \overline{\mathcal{E}_i} \sqcap A \sqcup \bigcap \overline{\mathcal{E}_i} \sqcap B \equiv \bigcap \overline{\mathcal{E}_i} \sqcap (A \sqcup B)$$

(note, the same holds for B); assume without loss of generality that $a \leq b$; then because

$$\mathcal{T} \models \bigcap \overline{\mathcal{E}_a} \sqcap A \not\sqsubseteq \bot$$

we can conclude that

$$\mathcal{T} \models \bigcap \overline{\mathcal{E}_a} \sqcap (A \sqcup B) \not\sqsubseteq \bot$$

so $rk_{\mathcal{E}}(A \sqcup B) \leq \min(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B))$

If $i < a$, then $\mathcal{T} \models \bigcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq \bot$ and $\mathcal{T} \models \bigcap \overline{\mathcal{E}_i} \sqcap B \sqsubseteq \bot$; since they are both empty, it must follow that their disjunction is also empty: $\mathcal{T} \models (\bigcap \overline{\mathcal{E}_a} \sqcap A) \sqcup (\bigcap \overline{\mathcal{E}_a} \sqcap B) \equiv \bigcap \overline{\mathcal{E}_a} \sqcap (A \sqcup B) \sqsubseteq \bot$. Hence $rk_{\mathcal{E}}(A \sqcup B) \geq a$.

Together, this gives $rk_{\mathcal{E}}(A \sqcup B) = \min(rk_{\mathcal{E}}(A), rk_{\mathcal{E}}(B))$.

(3)(i) Let concept A have $rk(A) = a > j$. Then $\mathcal{T} \models \bigcap \overline{\mathcal{E}_0} \sqcap A \sqsubseteq \bot$; this is equivalent to $\mathcal{T} \models \bigcap \overline{\mathcal{E}_0} \sqsubseteq \neg A$. We know that $\mathcal{T} \models \bigcap \overline{\mathcal{E}_j} \not\sqsubseteq \bot$; so since $\bigcap \overline{\mathcal{E}_j}$ is non-empty and subsumed by $\neg A$, its intersection with $\neg A$ is non-empty: $\mathcal{T} \models \bigcap \overline{\mathcal{E}_j} \sqcap \neg A \not\sqsubseteq \bot$. Hence $rk_{\mathcal{E}}(\neg A) leq j$. Furthermore, by our choice of j, for all $i < j$, $\bigcap \overline{\mathcal{E}_i} \sqsubseteq \bot$, so $\bigcap \overline{\mathcal{E}_i} \sqcap \neg A \sqsubseteq \bot$, so $rk_{\mathcal{E}}(\neg A) = j$.

(ii) Let concept A have $rk(A) = 0$. Then we have $\mathcal{T} \models \bigcap \overline{\mathcal{E}_0} \sqcap A \not\sqsubseteq \bot$; this does *not* imply $\mathcal{T} \models \bigcap \overline{\mathcal{E}_0} \sqcap \neg A \sqsubseteq \bot$; it is possible that $\mathcal{T} \models \bigcap \overline{\mathcal{E}_0} \sqcap A \not\sqsubseteq \bot$, and such an example does exist, and is presented in the body of the paper. □

Theorem 5.2: Let $\mathcal{E}$ and $\mathcal{F}$ be two rankings relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}', \mathcal{D}')$ respectively. If $\mathcal{E} \equiv \mathcal{F}$ then

(1) $rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$ if and only if $rk_{\mathcal{F}}(A \sqcap B) < rk_{\mathcal{F}}(A \sqcap \neg B)$
(2) $rk_{\mathcal{E}}(A \sqcap B) = rk_{\mathcal{E}}(A \sqcap \neg B)$ if and only if $rk_{\mathcal{F}}(A \sqcap B) = rk_{\mathcal{F}}(A \sqcap \neg B)$

Proof. Suppose $\mathcal{E}, \mathcal{F}$ are rankings such that $\mathcal{E} \equiv \mathcal{F}$ and $(\mathcal{K}, \mathcal{E}) \models A \subsetneq B$. Then $(\mathcal{K}', \mathcal{F}) \models A \subsetneq B$. So by theorem 4.8, $rk_{\mathcal{E}}(A \sqcap B) < rk_{\mathcal{E}}(A \sqcap \neg B)$ if and only if $rk_{\mathcal{F}}(A \sqcap B) < rk_{\mathcal{F}}(A \sqcap \neg B)$, proving (1).

Suppose instead then that $(\mathcal{K}, \mathcal{E}) \models A \not\subsetneq B$. Then $(\mathcal{K}', \mathcal{F}) \models A \not\subsetneq B$. This implies that $rk_{\mathcal{E}}(A \sqcap B) \geq rk_{\mathcal{E}}(A \sqcap \neg B)$ and $rk_{\mathcal{F}}(A \sqcap B) \geq$

$rk_{\mathcal{F}}(A \sqcap \neg B)$. If $rk_{\mathcal{E}}(A \sqcap B) > rk_{\mathcal{E}}(A \sqcap \neg B)$ then $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq \neg B$, so $(\mathcal{K}', \mathcal{F}) \models A \sqsubsetneq \neg B$ and $rk_{\mathcal{F}}(A \sqcap \neg B) < rk_{\mathcal{F}}(A \sqcap B)$.

Otherwise, $rk_{\mathcal{E}}(A \sqcap B) = rk_{\mathcal{E}}(A \sqcap \neg B)$; but then $(\mathcal{K}, \mathcal{E}) \models A \not\sqsubseteq B$ and $(\mathcal{K}, \mathcal{E}) \models A \not\sqsubseteq \neg B$, hence $(\mathcal{K}', \mathcal{F}) \models A \not\sqsubseteq B$ and $(\mathcal{K}', \mathcal{F}) \models A \not\sqsubseteq \neg B$, so $rk_{\mathcal{F}}(A \sqcap B) = rk_{\mathcal{F}}(A \sqcap \neg B)$. □

Lemma 5.4: Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a conditional knowledge base with the statements in $\mathcal{D}$ ranked into $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$. If there is some $i \in [0..n]$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bot$ then $\mathcal{E} \equiv \mathcal{E}[i+1:n]$, where $\mathcal{E}[i+1:n]$ is the subranking of $\mathcal{E}$ relative to conditional knowledge base $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ where $\mathcal{D}' = \bigcup_{j=i+1}^{n} \mathcal{E}_j$

Proof. Suppose $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bot$. Then for all concepts C we have $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$. In other words, no concepts will have rank i. Furthermore, since $\mathcal{E}_i \subseteq \mathcal{E}_j$ if $j \leq i$, we have $\mathcal{T}^* \models \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bot$. From this we can conclude that for all concepts C, $rk(C) > i$.

From theorem 4.8, we know that a statement $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$ if and only if $rk(A \sqcap B) < rk(A \sqcap \neg B)$. But $i < rk(A \sqcap B) < rk(A \sqcap \neg B)$. Hence the ranks $\mathcal{E}_j$ with $j \leq i$ do not participate in deciding if a defeasible subsumption is entailed by $\mathcal{K}$, and so we can discard all those $\mathcal{E}_j$ with $j \leq i$ and the set of statements entailed will be the same. This is exactly the set $\mathcal{D}'$ with ranking $\mathcal{E}_{i+1}, ..., \mathcal{E}_n$ described. □

Lemma 5.5: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the smallest integer such that $\bigsqcap \overline{\mathcal{E}_i} \equiv \top$. Then $\mathcal{E}' \equiv \mathcal{E}$, where $\mathcal{E}' = \mathcal{E}[0 : i-1]$.

Proof. ($\Longrightarrow$) Suppose $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$; let $j = rk_{\mathcal{E}}(A)$.

If $j < i$ then $\bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq B$; but $\mathcal{E}'_j = \mathcal{E}_j$, so $\bigsqcap \overline{\mathcal{E}'_j} \sqcap A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$.

If $j \geq i$, then $\bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq B$. By assumption, $\bigsqcap \overline{\mathcal{E}_i} \equiv \top$; but by lemma 4.4, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_j}$, so $top \equiv \bigsqcap \overline{\mathcal{E}_j}$. Therefore this is the same as $\top \sqcap A \sqsubseteq B$, hence $\mathcal{T} \models A \sqsubseteq B$. So then it must be the case that $(\mathcal{K}, \mathcal{E}') \models A \sqsubseteq B$.

($\Longleftarrow$) Suppose $(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$; let $j = rk_{\mathcal{E}'}(A)$.

If $j < i$ then $\bigsqcap \overline{\mathcal{E}'_j} \sqcap A \sqsubseteq B$; but $\mathcal{E}'_j = \mathcal{E}_j$, so $\bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$.

If $j \geq i$, then $\bigsqcap \overline{\mathcal{E}'_k} \sqcap A$ is empty for all integers $k \in [0, i-1]$. But $(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$, which means that $\mathcal{T} \models A \sqsubseteq B$. But then it must also be the case that $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$. □

Theorem 5.7: The ranking $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$ is dense and equivalent to $\mathcal{E}$.

Proof. Let $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$.

$\mathcal{E}'$ is dense:

For `CollapseRanking`, at every stage, `prev` $= \bigsqcap \overline{\mathcal{E}'_{j-1}}$. At initialization, `prev` is set equal to $\bigsqcap \overline{\mathcal{E}_0}$ and j is set equal to 1, so it is true here. In the while loop, j is incremented whenever $\mathcal{E}'_j$ is assigned to, and `prev` is set to $\bigsqcap \overline{\mathcal{E}_i}$ where $\mathcal{E}'_j$ was set to $\mathcal{E}_i$ for the previous j; hence `prev` $= \bigsqcap \overline{\mathcal{E}'_{j-1}}$.

The `if` statement in the `while` loop ensures that $\mathcal{E}'_j$ is only assigned to when `prev` $\not\equiv \bigsqcap \overline{\mathcal{E}_i}$; but `prev` $\equiv \bigsqcap \overline{\mathcal{E}'_{j-1}}$ and $\mathcal{E}'_j$ is set to $\mathcal{E}_i$, which means $\bigsqcap \overline{\mathcal{E}'_{j-1}} \not\equiv \bigsqcap \overline{\mathcal{E}'_j}$ hence for all $0 \leq j < n$, $\bigsqcap \overline{\mathcal{E}'_j} \not\equiv \bigsqcap \overline{\mathcal{E}'_{j+1}}$, so $\mathcal{E}'$ satisfies the definition of a dense ranking.

$\mathcal{E}' \equiv \mathcal{E}$:

`CollapseRanking` preserves the relative order of statements in $\mathcal{E}$; in other words, ranks of $\mathcal{E}$ are added to $\mathcal{E}'$ in the order they appear in $\mathcal{E}$; so if $\mathcal{E}_i$, $\mathcal{E}_j$ are such that $i < j$ and $\mathcal{E}_i = \mathcal{E}'_{i'}$ and $\mathcal{E}_j = \mathcal{E}'_{j'}$, then $i' < j'$.

We show that $\mathcal{E}'$ is equivalent to $\mathcal{E}$ by showing that $(\mathcal{K}, \mathcal{E})$ entails some $A \sqsubsetneq B$ if and only if $(\mathcal{K}, \mathcal{E}')$ entails that $A \sqsubsetneq B$. We do this first in the forward direction, then the reverse.

$(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B \implies (\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$:

Suppose $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$. Let $i = rk_{\mathcal{E}}(A)$. Then $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$.

If $i = 0$, then $\mathcal{E}_i = \mathcal{E}'_i = \mathcal{E}_0$; so $\bigsqcap \overline{\mathcal{E}'_i} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$.

If $i > 0$, then for all $0 \leq k < i$, $\bigsqcap \mathcal{E}_k \sqcap A \sqsubseteq \bot$. But $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$, hence $\bigsqcap \overline{\mathcal{E}_i} \not\equiv \bigsqcap \overline{\mathcal{E}_k}$. In `CollapseRanking`, when constructing $\mathcal{E}'$, `prev` $= \bigsqcap \overline{\mathcal{E}_{k'}}$ for some $k' < i$; but then `prev` $\not\equiv \bigsqcap \overline{\mathcal{E}_i}$, so $\mathcal{E}_i$ is added to $\mathcal{E}'$ at j; furthermore, for all $j' < j$, $\mathcal{E}'_{j'}$ is taken from $\mathcal{E}$ where it must have rank less than i, hence $\bigsqcap \overline{\mathcal{E}'_{j'}} \sqcap A \sqsubseteq \bot$. So $rk_{\mathcal{E}'}(A) = j$. Furthermore, since $\mathcal{E}'_j = \mathcal{E}_i$, it must be the case that $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_j} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$.

$(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B \implies (\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$:

Suppose $(\mathcal{K}, \mathcal{E}') \models A \sqsubsetneq B$. Let $j = rk_{\mathcal{E}'}(A)$. Then $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_j} \sqcap A \not\sqsubseteq \bot$ and $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_j} \sqcap A \sqsubseteq B$, and for all $0 \leq k < j$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_k} \sqcap A \sqsubseteq \bot$.

Since ranks of $\mathcal{E}'$ are only taken directly from $\mathcal{E}$, there is some i such that $\mathcal{E}_i = \mathcal{E}'_j$; also, for every $0 \leq k < i$ there is some k' with $0 \leq k' < j$ for which $\bigsqcap \overline{\mathcal{E}_k} \equiv \bigsqcap \overline{\mathcal{E}'_{k'}}$. This is the case because the `CollapseRanking` only adds a rank $\mathcal{E}_i$ to $\mathcal{E}'$ when $\bigsqcap \overline{\mathcal{E}_i} \not\equiv$ `prev`; but `prev` is always equal to $\bigsqcap \overline{\mathcal{E}'_j}$ where $\mathcal{E}'_j$ is the highest rank in $\mathcal{E}'$; so if a rank $\mathcal{E}_i$ is *not* added to $\mathcal{E}'$, then it is equivalent to `prev`, which is already in the ranking, and if it *is* added, then it is present in the ranking $\mathcal{E}'$ and equivalent to itself. Furthermore, as previously noted, the relative ranking of these statements is preserved.

Now we have that for all $0 \leq k < i$, there is some k' with $0 \leq k' < j$ for which $\bigsqcap \overline{\mathcal{E}_k} \equiv \bigsqcap \overline{\mathcal{E}'_{k'}}$; but for all $0 \leq k' < j$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_{k'}} \sqcap A \sqsubseteq \bot$, hence $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_k} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}'_{k'}} \sqcap A \sqsubseteq \bot$; so $rk_{\mathcal{E}}(A) = i$. Furthermore, since $\mathcal{E}_i = \mathcal{E}'_j$, we have that $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}'_j} \sqcap A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{E}) \models A \sqsubsetneq B$. □

Corollary 5.8: $\mathcal{E} \equiv \mathcal{F}$ if and only if $\texttt{CollapseRanking}(\mathcal{E}) \equiv \texttt{CollapseRanking}(\mathcal{F})$.

Proof. Let

$$\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{E})$$

$$\mathcal{F}' = \texttt{CollapseRanking}(\mathcal{F})$$

($\implies$) Suppose $\mathcal{E} \equiv \mathcal{F}$. By theorem 5.7, $\mathcal{E} \equiv \mathcal{E}'$ and $\mathcal{F} \equiv \mathcal{F}'$. So $\mathcal{F}' \equiv \mathcal{F} \equiv \mathcal{E} \equiv \mathcal{E}'$.

($\impliedby$) Suppose $\mathcal{E}' \equiv \mathcal{F}'$. By theorem 5.7, $\mathcal{E} \equiv \mathcal{E}'$ and $\mathcal{F} \equiv \mathcal{F}'$. So $\mathcal{E} \equiv \mathcal{E}' \equiv \mathcal{F}' \equiv \mathcal{F}$. □

Lemma 5.10: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be rankings relative to the same knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ with $m = n$ such that for all integers i with $0 \leq i \leq n$, $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$; then $\mathcal{E} \equiv \mathcal{F}$.

Proof. ($\implies$): Suppose $(\mathcal{K}, \mathcal{E}) \models A \subsetneq B$. Let $i = rk_{\mathcal{E}}(A)$.

If $i \leq n$, then $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$, and for all integers j such that $0 \leq j < i$, $\bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq \bot$, but $\bigsqcap \overline{\mathcal{E}_i} \sqcap A \not\sqsubseteq \bot$.

By assumption, $\mathcal{F}$ is such that for all $0 \leq j \leq n$, $\bigsqcap \overline{\mathcal{F}_j} \equiv \bigsqcap \overline{\mathcal{E}_j}$. If $j < i$, then $\bigsqcap \overline{\mathcal{F}_j} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq \bot$, and if $j = i$, then $\bigsqcap \overline{\mathcal{F}_j} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_j} \sqcap A \not\sqsubseteq \bot$, so $rk_{\mathcal{F}}(A) = rk_{\mathcal{E}}(A) = i$. Furthermore $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}_j} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{F}) \models A \subsetneq B$.

If $i > n$ then $(\mathcal{K}, \mathcal{E}) \models A \subsetneq B \implies \mathcal{T} \models A \sqsubseteq B$, and for all $0 \leq j \leq n$, $\bigsqcap \overline{\mathcal{E}_j} \sqcap A \sqsubseteq \bot$; but $\bigsqcap \overline{\mathcal{E}_j} \equiv \bigsqcap \overline{\mathcal{F}_j}$, so $\bigsqcap \overline{\mathcal{F}_j} \sqcap A \sqsubseteq \bot$. $\mathcal{T} \models A \sqsubseteq B$, so $(\mathcal{K}, \mathcal{F}) \models A \subsetneq B$.

($\impliedby$): The same argument can be applied as in the $\implies$ case, but with $\mathcal{E}$ switched with $\mathcal{F}$. □

Theorem 5.11: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be dense rankings relative to the same knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ such that $\bigsqcap \overline{\mathcal{E}_m} \not\equiv \top$ and $\bigsqcap \overline{\mathcal{F}_n} \not\equiv \top$[§]; then $\mathcal{E} \equiv \mathcal{F}$ if and only if $m = n$ and for all integers i such that $0 \leq i \leq n$, $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$.

Proof.

($\impliedby$): Follows from lemma 5.10.

($\implies$): We prove this by induction on i, the number of the current rank.

Base case: $n = 0$. Since $\mathcal{E}$ and $\mathcal{F}$ are rankings relative to the same DTBox $\mathcal{D}$, we must have $\mathcal{E}_0 = \mathcal{F}_0 = \mathcal{D}$, so $\bigsqcap \overline{\mathcal{E}_0} \equiv \bigsqcap \overline{\mathcal{D}} \equiv \bigsqcap \overline{\mathcal{F}_0}$.

Inductive step: Suppose that for all $0 \leq j < i$, $\bigsqcap \overline{\mathcal{E}_j} \equiv \bigsqcap \overline{\mathcal{F}_j}$. Then in particular, $\bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \bigsqcap \overline{\mathcal{F}_{i-1}}$.

There are two cases to consider:

Case 1: $\mathcal{E}_i$ and $\mathcal{F}_i$ exist.

By the inductive hypothesis, $\bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \bigsqcap \overline{\mathcal{F}_{i-1}}$. Let concept $E = \neg \bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \neg \bigsqcap \overline{\mathcal{F}_{i-1}}$.

$$\mathcal{T} \models \bigsqcap \overline{\mathcal{F}_{i-1}} \sqcap E \equiv \bigsqcap \overline{\mathcal{E}_{i-1}} \sqcap \neg \bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \bot$$

so $rk_{\mathcal{E}}(E) > i-1$ and $rk_{\mathcal{F}}(E) > i-1$. Furthermore, since $\mathcal{E}_i \subseteq \mathcal{E}_{i-1}$, we have $\bigsqcap \overline{\mathcal{E}_i} \sqsupseteq \bigsqcap \overline{\mathcal{E}_{i-1}}$; $\mathcal{E}$ is dense, so $\bigsqcap \overline{\mathcal{E}_i} \not\equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$. Thus $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubset \bigsqcap \overline{\mathcal{E}_i}$. But then

$$\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap E \equiv \bigsqcap \overline{\mathcal{E}_i} \sqcap \neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \not\equiv \bot$$

hence $rk_{\mathcal{E}}(E) \leq i$. The previous argument applies equally to $\mathcal{F}$, so $rk_{\mathcal{F}}(E) \leq i$. Therefore $rk_{\mathcal{E}}(E) = rk_{\mathcal{F}}(E) = i$.

Clearly, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, so by the `RationalClosure` algorithm, $(\mathcal{K}, \mathcal{E}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_i}$. But $\mathcal{E} \equiv \mathcal{F}$, so $(\mathcal{K}, \mathcal{F}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_i}$, so $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$.

Similarly, $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_i}$, so by the `RationalClosure` algorithm, $(\mathcal{K}, \mathcal{F}) \models E \subsetneq \bigsqcap \overline{\mathcal{F}_i}$. But $\mathcal{E} \equiv \mathcal{F}$, so $(\mathcal{K}, \mathcal{E}) \models E \subsetneq \bigsqcap \overline{\mathcal{F}_i}$, so $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_i}$.

So we have (1) $\bigsqcap \overline{\mathcal{E}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_i}$ and (2) $\bigsqcap \overline{\mathcal{F}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$. From (1), we can derive:

$$\begin{aligned}
&\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_i} \\
\implies &\left(\bigsqcap \overline{\mathcal{E}_i} \sqcap E \right) \sqcup \neg E \sqsubseteq \bigsqcap \overline{\mathcal{F}_i} \sqcup \neg E \\
\iff &\left[\bigsqcap \overline{\mathcal{E}_i} \sqcap \neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \sqcup \neg\neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \sqsubseteq \bigsqcap \overline{\mathcal{F}_i} \sqcup \neg\neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \\
\iff &\left[\bigsqcap \overline{\mathcal{E}_i} \sqcap \neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \sqsubseteq \bigsqcap \overline{\mathcal{F}_i} \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right)
\end{aligned}$$

By the inductive hypothesis, $\bigsqcap \overline{\mathcal{F}_{i-1}} \equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$ and $\bigsqcap \overline{\mathcal{F}_{i-1}} \sqsubset \bigsqcap \overline{\mathcal{F}_i}$; hence $\bigsqcap \overline{\mathcal{F}_i} \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \equiv \bigsqcap \overline{\mathcal{F}_i}$.

We also have the following:

$$\begin{aligned}
&\left[\bigsqcap \overline{\mathcal{E}_i} \sqcap \neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \\
\equiv &\left[\bigsqcap \overline{\mathcal{E}_i} \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \sqcap \left[\neg \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \\
\equiv &\left[\bigsqcap \overline{\mathcal{E}_i} \sqcup \left(\bigsqcap \overline{\mathcal{E}_{i-1}} \right) \right] \sqcap \top \\
\equiv &\bigsqcap \overline{\mathcal{E}_i} \sqcup \bigsqcap \overline{\mathcal{E}_{i-1}}
\end{aligned}$$

But $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, so $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqcup \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_i}$.

This together with the derivation from (1) gives

$$\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{F}_i}$$

Using the same argument, (2) would allow us to derive $\bigsqcap \overline{\mathcal{F}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$. This is only satisfied if

$$\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$$

Case 2: Only one of $\mathcal{E}_i$ and $\mathcal{F}_i$ exists.

Assume without loss of generality that $\mathcal{E}_i$ exists but $\mathcal{F}_i$ does not. This can only be the case when $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ and $n < i \leq m$.

From case 1, we know that $E = \neg \bigsqcap \overline{\mathcal{E}_{i-1}}$ has $rk_{\mathcal{E}}(E) = i$; furthermore, $(\mathcal{K}, \mathcal{E}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_i}$. $\mathcal{E} \equiv \mathcal{F}$, so $(\mathcal{K}, \mathcal{F}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_i}$. However, $i > n$ which is the maximum rank of $\mathcal{F}$, so by the `RationalClosure` algorithm, this means that $\mathcal{T} \models \neg \bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$. But $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, so $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_{i-1}} \sqcup \neg \bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$ which implies that $\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$. But this violates the original premise that $\bigsqcap \overline{\mathcal{E}_n} \not\equiv \top$, a contradiction. □

Theorem 5.12: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ and $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ be dense rankings relative to the knowledge bases $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ respectively such that $\bigsqcap \overline{\mathcal{E}_0} \not\equiv \bot$ and $\bigsqcap \overline{\mathcal{F}_0} \not\equiv \bot$ and $\bigsqcap \overline{\mathcal{E}_m} \not\equiv \top$ and $\bigsqcap \overline{\mathcal{F}_n} \not\equiv \top$; then $\mathcal{E} \equiv \mathcal{F}$ if

and only if $m = n$ and for all integers i such that $0 \leq i \leq n$, $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$.

Proof. ($\Longleftarrow$): Follows from lemma 5.10.

($\Longrightarrow$): We prove this by induction on i, the number of the current rank.

Base case: $n = 0$. Let $E = \top$. Since $\bigsqcap \overline{\mathcal{E}_0} \not\equiv \bot$, $\bigsqcap \overline{\mathcal{E}_0} \sqcap E \equiv \bigsqcap \overline{\mathcal{E}_0} \not\sqsubseteq \bot$. So $rk_{\mathcal{E}}(E) = 0$. Similarly, $\bigsqcap \overline{\mathcal{F}_0} \sqcap E \not\sqsubseteq \bot$, so $rk_{\mathcal{F}}(E) = 0$.

Clearly, $\bigsqcap \overline{\mathcal{E}_0} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}$, so $(\mathcal{K}, \mathcal{E}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_0}$; $\mathcal{E} \equiv \mathcal{F}$, so $(\mathcal{K}', \mathcal{F}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_0}$. But then $\bigsqcap \overline{\mathcal{F}_0} \equiv \bigsqcap \overline{\mathcal{F}_0} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}$, so $\bigsqcap \overline{\mathcal{F}_0} \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}$.

Similarly, $\bigsqcap \overline{\mathcal{F}_0} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_0}$, so $(\mathcal{K}', \mathcal{F}) \models E \subsetneq \bigsqcap \overline{\mathcal{F}_0}$; $\mathcal{F} \equiv \mathcal{E}$, so $(\mathcal{K}, \mathcal{E}) \models E \subsetneq \bigsqcap \overline{\mathcal{E}_0}$. But then $\bigsqcap \overline{\mathcal{E}_0} \equiv \bigsqcap \overline{\mathcal{E}_0} \sqcap E \sqsubseteq \bigsqcap \overline{\mathcal{F}_0}$, so $\bigsqcap \overline{\mathcal{E}_0} \sqsubseteq \bigsqcap \overline{\mathcal{F}_0}$.

Together, this gives $\bigsqcap \overline{\mathcal{E}_0} \equiv \bigsqcap \overline{\mathcal{F}_0}$.

Inductive step: This is identical to the inductive step in the proof of theorem 5.11. □

Corollary 5.13: $\mathcal{E} \equiv \mathcal{F}$ if and only if $n' = m'$ and for all integers i such that $0 \leq i \leq n$, $\bigsqcap \overline{\mathcal{E}_i''} \equiv \bigsqcap \overline{\mathcal{F}_i''}$.

$[m, n, m', n', \mathcal{E}', \mathcal{F}', \mathcal{E}'', \mathcal{F}''$ are all defined in the section on equivalence where the corollary is initially presented.]

Proof.

($\Longrightarrow$) Suppose $\mathcal{E} \equiv \mathcal{F}$.

By corollary 5.8,

$$\texttt{CollapseRanking}(\mathcal{K}, \mathcal{E}) \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{F})$$

Then by lemmas 5.4 and 5.5,

$$\mathcal{E}'' \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$$

and

$$\mathcal{F}'' \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{F})$$

so $\mathcal{E}'' \equiv \mathcal{F}''$. Furthermore, by theorem 5.7, $\mathcal{E}''$ and $\mathcal{F}''$ are both dense. The result follows from theorem 5.12.

($\Longleftarrow$) Suppose $n' = m'$ and for all integers i such that $0 \leq i \leq n$, $\bigsqcap \overline{\mathcal{E}_i''} \equiv \bigsqcap \overline{\mathcal{F}_i''}$.

Then by theorem 5.12, $\mathcal{E}'' \equiv \mathcal{F}''$. By lemmas 5.4 and 5.5,

$$\mathcal{E}'' \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$$

and

$$\mathcal{F}'' \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{F})$$

By theorem 5.7,

$$\mathcal{E} \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$$

and

$$\mathcal{F} \equiv \texttt{CollapseRanking}(\mathcal{K}, \mathcal{F})$$

Hence $\mathcal{E} \equiv \mathcal{F}$. □

Theorem 6.4: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a dense ranking relative to knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the largest integer such that $C \subsetneq D \in \mathcal{E}_i$. Then $C \subsetneq D$ is totally redundant if and only if it is locally redundant at i.

Proof. ($\Longrightarrow$) Let $C \subsetneq D$ be totally redundant for ranking $\mathcal{E}$ with conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Then $\mathcal{E}' = (\mathcal{E}_0 \backslash \{C \subsetneq D\}, ..., \mathcal{E}_n \backslash \{C \subsetneq D\})$ with $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ is such that $\mathcal{E} \equiv \mathcal{E}'$.

$\mathcal{E}'$ is dense: by theorem 5.7, $\mathcal{E}'' = \texttt{CollapseRanking}(\mathcal{K}', \mathcal{E}')$ is dense and equivalent to $\mathcal{E}'$, so it is equivalent to $\mathcal{E}$; but then by theorem 5.12, $|\mathcal{E}''| = |\mathcal{E}|$, but $|\mathcal{E}| = |\mathcal{E}'|$, so $|\mathcal{E}'| = |\mathcal{E}''|$. CollapseRanking constructs $\mathcal{E}''$ by taking statements from $\mathcal{E}'$. Since $|\mathcal{E}'| = |\mathcal{E}''|$, all statements of $|\mathcal{E}'|$ were taken for $|\mathcal{E}''|$, so $\mathcal{E}'$ is exactly $\mathcal{E}''$ which is dense.

Then by theorem 5.12, for all $j \in [0, n]$, $\bigsqcap \overline{\mathcal{E}_j} \equiv \bigsqcap \overline{\mathcal{E}_j'}$. In particular, $\bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_i'}$. But $\mathcal{E}_i' = \mathcal{E}_i \backslash \{C \subsetneq D\}$, so $C \subsetneq D$ satisfies local redundancy at i.

($\Longleftarrow$) Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a conditional knowledge base and $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ a dense ranking of the statements in $\mathcal{D}$; let $i \in [0, n]$ be the highest such that $C \subsetneq D \in \mathcal{E}_i$, and suppose that $C \subsetneq D$ is locally redundant at i. Let $\mathcal{E}' = (\mathcal{E}_0 \backslash \{C \subsetneq D\}, ..., \mathcal{E}_n \backslash \{C \subsetneq D\})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{D} \backslash \{C \subsetneq D\})$. For all $j > i$, $C \subsetneq D \notin \mathcal{E}_j$, so $\mathcal{E}_j' = \mathcal{E}_j$ hence $\bigsqcap \overline{\mathcal{E}_j'} \equiv \bigsqcap \overline{\mathcal{E}_j}$. Since $C \subsetneq D$ is locally redundant at i, by theorem 6.2 it is locally redundant at j for all $0 \leq j \leq i$, so $\bigsqcap \overline{\mathcal{E}_j} \equiv \bigsqcap \overline{\mathcal{E}_j'}$.

$\mathcal{E}$ is dense, so $\mathcal{E}'$ must be dense. Suppose not. Then there is some k such that $\bigsqcap \overline{\mathcal{E}_k'} \equiv \bigsqcap \overline{\mathcal{E}_{k+1}'}$. But we have shown that for all integers $j \in [0, n]$ $\bigsqcap \overline{\mathcal{E}_j'} \equiv \bigsqcap \overline{\mathcal{E}_j}$. So then $\bigsqcap \overline{\mathcal{E}_k} \equiv \bigsqcap \overline{\mathcal{E}_{k+1}}$, contradicting the density of $\mathcal{E}$.

Now $\mathcal{E}$ and $\mathcal{E}'$ are both dense, $|\mathcal{E}| = |\mathcal{E}'|$, and for all $j \in [0, n]$, $\bigsqcap \overline{\mathcal{E}_j'} \equiv \bigsqcap \overline{\mathcal{E}_j}$. So by theorem 5.12, $\mathcal{E} \equiv \mathcal{E}'$. Hence $C \subsetneq D$ is totally redundant. □

Theorem 7.1: Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to conditional knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $\mathcal{D}' = \{\top \subsetneq \bigsqcap \overline{\mathcal{E}_0}\} \cup \{\neg \bigsqcap \overline{\mathcal{E}_i} \subsetneq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \leq i < n\}$. Let $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ and $\mathcal{E}' = \texttt{ComputeRanking}(\mathcal{K}')$. Then $\mathcal{E} \equiv \mathcal{E}'$.

Proof. We analyse the action of ComputeRanking on input $(\mathcal{T}, \mathcal{D}')$.

Since $\mathcal{D}'$ is non-empty and $\mathcal{D}_\infty$ is set equal to $\mathcal{D}'$, the while loop of line 5 will run at least once.

We start with $\mathcal{E}_0'$ set equal to $\mathcal{D}'$. By lemma 7.2, $\bigsqcap \overline{\mathcal{E}_0'} \equiv \bigsqcap \overline{\mathcal{E}_0}$.

The algorithm then sets $\mathcal{E}_1' = \texttt{Exceptional}(\mathcal{T}, \mathcal{E}_0')$. By lemmas 7.3 and 7.4, we now have $\bigsqcap \overline{\mathcal{E}_1'} \equiv \bigsqcap \overline{\mathcal{E}_1}$.

The while loop of line 9 will proceed until $\mathcal{E}_{i+1}' = \mathcal{E}_i'$. By lemma 7.4, we see that $\mathcal{E}_j' = \{\neg \bigsqcap \overline{\mathcal{E}_j} \subsetneq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \leq j < n\}$; it is clear that since $\mathcal{E}$ is dense, for every $j \in [0, n]$, no $j, j+1$ pair will have $\mathcal{E}_j' = \mathcal{E}_{j+1}'$. Hence the algorithm will iterate until $i = n + 1$, in which case $\mathcal{E}_{n+1}' = \mathcal{E}_{n+2}' = \emptyset$.

The body of the while loop will therefore set

$$\mathcal{E}_j' = \texttt{Exceptional}(\mathcal{T}, \mathcal{E}_{j-1}')$$

for every integer $j \in [2, n]$. By lemmas 7.3 and 7.4, this ensures that $\bigsqcap \overline{\mathcal{E}'_j} \equiv \bigsqcap \overline{\mathcal{E}_j}$.

Now $\mathcal{D}_\infty$ is set to $\mathcal{E}_i$, but as already noted, $\mathcal{E}_i = \emptyset$, so the while loop of line 5 will not run again. Furthermore, no statements are added to $\mathcal{T}$ or removed from $\mathcal{D}$ in lines 14 and 15. The algorithm then returns $\mathcal{E}'$ as its ranking.

In our analysis of the algorithm, we saw that for every $i \in [0, n]$, $\bigsqcap \overline{\mathcal{E}'_i} \equiv \bigsqcap \overline{\mathcal{E}_i}$, and that there were two extra ranks, $\mathcal{E}_{n+1}$ and $\mathcal{E}_{n+2}$, which were equal to $\emptyset$. It should be clear that $\mathcal{E}_{n+1}$ and $\mathcal{E}_{n+2}$ do not change the set of entailed statement, so we can set $\mathcal{E}'' = \mathcal{E}'[0:n]$ and have $\mathcal{E}'' \equiv \mathcal{E}'$. By lemma 5.10, we can conclude that that $\mathcal{E}'' \equiv \mathcal{E}$, hence $\mathcal{E}' \equiv \mathcal{E}$. □

Lemma 7.2: $\bigsqcap \overline{\mathcal{E}'_0} \equiv \bigsqcap \overline{\mathcal{D}'} \equiv \bigsqcap \overline{\mathcal{E}_0}$.

Proof.

$$\begin{aligned}\bigsqcap \overline{\mathcal{D}'} &= \overline{\{\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}\}} \sqcap \bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\}} \\ &= (\neg\top \sqcup \bigsqcap \overline{\mathcal{E}_0}) \sqcap \bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\}} \\ &\equiv \bigsqcap \overline{\mathcal{E}_0} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_i} \sqcup \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\}\end{aligned}$$

By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \mathcal{E}_{i+1}$, so $\bigsqcap \overline{\mathcal{E}_i} \sqcup \bigsqcap \overline{\mathcal{E}_{i+1}} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$, so this reduces to

$$\begin{aligned}\bigsqcap \overline{\mathcal{D}'} &\equiv \bigsqcap \overline{\mathcal{E}_0} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\} \\ &\equiv \bigsqcap \overline{\mathcal{E}_0} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_i} \mid i \in \mathbb{Z}, 0 < i \le n\}\end{aligned}$$

By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_0} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$ for all $0 < i \le n$, so $\bigsqcap \overline{\mathcal{E}_0} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_i} \mid i \in \mathbb{Z}, 0 < i \le n\} \equiv \bigsqcap \overline{\mathcal{E}_0}$. Hence

$$\bigsqcap \overline{\mathcal{E}'_0} \equiv \bigsqcap \overline{\mathcal{D}'} \equiv \bigsqcap \overline{\mathcal{E}_0}$$

□

Lemma 7.3: For all $i \in [1, n]$,

$$\bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$$

Proof.

$$\begin{aligned}&\bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}} \\ \equiv &\bigsqcap \{\bigsqcap \overline{\mathcal{E}_j} \sqcup \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}\end{aligned}$$

By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_j}$ for all $i \le j \le n$, so $\bigsqcap \overline{\mathcal{E}_j} \sqcup \bigsqcap \overline{\mathcal{E}_{j+1}} \equiv \bigsqcap \overline{\mathcal{E}_{j+1}}$. The above then reduces to

$$\begin{aligned}&\bigsqcap \{\bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\} \\ \equiv &\bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i+1 \le j < n\}\end{aligned}$$

By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_j}$ for all $i \le j \le n$, so $\bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap \bigsqcap \{\bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i+1 \le j < n\} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$ as required. □

Lemma 7.4: For all integers $i \in [0, n]$,

$$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i) = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}$$

Proof. **Base case:** $i = 0$. Then $\mathcal{E}'_0 = \mathcal{D} = \{\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}\} \cup \{\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\}$. $\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_0)$ returns every $C \sqsubseteq D$ in $\mathcal{E}'_0$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_0} \sqsubseteq \neg C$. But $\bigsqcap \overline{\mathcal{E}'_0} \equiv \bigsqcap \overline{\mathcal{E}_0}$. With the exception of $\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}$, every $C \sqsubseteq D \in \mathcal{E}'_0$ is of the form $\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}}$ for some $i \ge 1$, so $\neg C \equiv \bigsqcap \overline{\mathcal{E}_i}$. By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_0} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$ for all $0 < i \le n$, so with the exception of $\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}$, every $C \sqsubseteq D \in \mathcal{E}'_0$ is in $\mathcal{E}'_1 = \texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_0)$. In other words, $\mathcal{E}'_1 = \mathcal{E}'_0 \backslash \{\top \sqsubseteq \bigsqcap \overline{\mathcal{E}_0}\} = \{\neg \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \le i < n\}$.

Inductive step: Suppose it is true for $j < i$. We consider two cases:

Case 1: $i = n$. Then by the inductive hypothesis,

$$\mathcal{E}'_i = \texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_{i-1}) = \{\neg \bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}\}$$

$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i)$ returns every $C \sqsubseteq D$ in $\mathcal{E}'_i$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_i} \sqsubseteq \neg C$. By lemma 7.3, $\bigsqcap \overline{\mathcal{E}'_i} \equiv \bigsqcap \overline{\mathcal{E}_i}$, so this is equivalent to every $C \sqsubseteq D$ in $\mathcal{E}'_i$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \neg C$. There is only one statement in $\mathcal{E}'_i$, for which $\neg C = \bigsqcap \overline{\mathcal{E}_{i-1}}$. $\mathcal{E}$ is dense and by lemma 4.4, $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, so $\bigsqcap \overline{\mathcal{E}_i} \not\sqsubseteq \bigsqcap \overline{\mathcal{E}_{i-1}}$, hence this is not in the set returned by $\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i)$. There are no other statements, so

$$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i) = \emptyset = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}$$

as required.

Case 2: $i < n$. By the inductive hypothesis,

$$\mathcal{E}'_i = \texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_{i-1}) = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i-1 \le j < n\}$$

$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i)$ returns every $C \sqsubseteq D$ in $\mathcal{E}'_i$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_i} \sqsubseteq \neg C$. By lemma 7.3, $\bigsqcap \overline{\mathcal{E}'_i} \equiv \bigsqcap \overline{\mathcal{E}_i}$, so this is equivalent to every $C \sqsubseteq D$ in $\mathcal{E}'_i$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \neg C$.

$\mathcal{E}'_i$ can be written equivalently as

$$\mathcal{E}'_i = \{\neg \bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}\} \cup \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}$$

For the first set in the union, the only statement is $\neg \bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, which has $\neg C = \bigsqcap \overline{\mathcal{E}_{i-1}}$. $\mathcal{E}$ is dense and by lemma 4.4, $\bigsqcap \overline{\mathcal{E}_{i-1}} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$, so $\bigsqcap \overline{\mathcal{E}_i} \not\sqsubseteq \bigsqcap \overline{\mathcal{E}_{i-1}}$, hence this is not in the set returned by $\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i)$.

For the second set in the union, every statement is of the form $\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}}$ for some $j \ge i$, each of which has $\neg C = \bigsqcap \overline{\mathcal{E}_j}$ for some $j \ge i$. By lemma 4.4, $\bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_j}$ for all $j \ge i$, so they will be returned by $\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i)$. Hence

$$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}'_i) = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \le j < n\}$$

as required. □