A Proposal on Extending Defeasible Reasoning Beyond Rational Closure

Maqhobosheane Mohlerepe
University of Cape Town
Cape Town, South Africa
mhlmaq001@myuct.ac.za

Thabo Vincent Moloi
University of Cape Town
Cape Town, South Africa
mlxtha036@myuct.ac.za

Mamodike Sadiki
University of Cape Town
Cape Town, South Africa
sdkmam0016@myuct.ac.za

ABSTRACT

Logic plays a pivotal role in artificial intelligence (AI), serving as a foundational framework for representing and reasoning about knowledge. Classical reasoning, grounded in classical logic, employs deductive reasoning to derive conclusions from a set of premises, adhering to the principle of bivalence, which posits that propositions are either true or false. In contrast, defeasible reasoning offers a means to navigate the complexities of real-world decision-making by accommodating exceptions and uncertainties. This paper explores the theoretical underpinnings and current practical implementations of defeasible reasoning within the context of artificial intelligence (AI). Theoretical discussions delve into the limitations of classical propositional logic and the emergence of defeasible reasoning as a solution to address these shortcomings. Furthermore, the implementation aspect reviews existing defeasible reasoning tools, focusing on algorithms, optimizations, and their implications for enhancing performance and efficacy. The project outlined herein aims to develop an intuitive defeasible reasoning tool, leveraging insights from both theoretical and implementation-focused research [21, 22, 26].

KEYWORDS

Artificial Intelligence, Knowledge Representation and Reasoning, Defeasible Reasoning, Propositional Logic, Rational Closure, KLM framework

1 INTRODUCTION

Knowledge Representation and Reasoning (KRR) forms a crucial domain within the field of AI, focusing on the symbolic representation of knowledge and its automated manipulation through reasoning programs [4]. The knowledge in question refers to information about the world which computers can't understand. The most basic form of reasoning is classical reasoning. Classical reasoning, characterized by deductive and inductive processes, provides a structured framework for drawing conclusions from premises. However, the strict adherence to deductive validity fails to account for exceptions and uncertainties encountered in real-world scenarios. Defeasible reasoning offers a solution by allowing for rational but not necessarily deductively valid inferences. This departure from classical logic is essential for modeling human-like cognition in AI systems, where decision making often involves navigating complex, uncertain environments. The most widely recognized framework for defeasible reasoning is the KLM framework developed by Kraus, Lehmann and Magidor [7]. In the KLM approach, defeasible reasoning is modeled using non-monotonic logics, which allow for conclusions to be drawn tentatively, subject to revision in light of new information. The key idea behind the KLM approach is the use of preference orderings on rules to determine the strength of conclusions. In this project, we aim to develop a defeasible reasoning tool that incorporates insights from theoretical research on nonmonotonic reasoning and practical implementations of defeasible reasoners [21, 22, 26].

2 BACKGROUND

2.1 Propositional Logic

Propositional logic is a language that formalizes knowledge representation and reasoning. It deals with indivisible statements, known as atomic propositions [14], which can either be true or false. This language utilizes atomic propositions and logical operators ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$) to construct propositional formulas[5]. A set of these formulas is formally denoted by $\mathcal{L}$. Valuations are all the logically possible states of the world based on a set of propositional atoms $\mathcal{P}$ in some language $\mathcal{L}$[15]. Formally, a valuation is a function $u : \mathcal{P} \rightarrow \{\mathrm{T}, \mathrm{F}\}$ where T and F are True and False, respectively. A valuation u is said to satisfy a formula α in the language $\mathcal{L}$, formally denoted as $u \Vdash \alpha$, if the truth value of α under the valuation u is true [21, 22, 26].

Any valuation u that satisfies α is a model of α, denoted as $\mathrm{Mod}(\alpha)$. A knowledge base (denoted by $\mathcal{K}$) is a finite set of propositional formulas from which conclusions can be drawn. The set of valuations that satisfy all the formulas in a knowledge base $\mathcal{K}$ are the models of $\mathcal{K}$, denoted as $\mathrm{Mod}(\mathcal{K})$ [21, 22, 26].

2.2 Entailment

Entailment is the notion that using the relationship between two or more statements one can make deductions or conclusions . This is a fundamental concept in reasoning, it formally shows how certain states of the world are logical consequence of others [29]. Suppose we are given propositional statements or formulas, $\alpha, \beta \in \mathcal{L}$, α is said to entail β or β referred to as a logical consequence of α, denoted $\alpha \models \beta$ if and only if the models of α are a subset of the models of β ($Mod(\alpha) \subseteq Mod(\beta)$). The set of models of α , denoted $Mod(\alpha)$, are the set of all valuations that satisfy α [15, 21, 22, 26].

2.3 KLM: Defeasible Entailment

Defeasible reasoning is a type of reasoning where conclusions are derived from information that is incomplete or uncertain. It differs from classical reasoning in that once a conclusion is derived from a set of premises, it is considered definitive and does not change with the addition of new information. This state of information is referred to as monotonic. In defeasible reasoning conclusions can change when new information is introduced and the state of information is non-monotonic. This property of defeasible reasoning allows us to model real-world scenarios more accurately

[1, 15, 21, 22, 26] . Suppose we have a knowledge base $\mathcal{K}$ with the following statements: $\{b \rightarrow f, b \rightarrow w, p \rightarrow b, p \rightarrow \neg f\}$, where the atoms, b, f, w and p represent birds, fly, wings and penguins, respectively. The complete formulas or statements meaning birds can fly, birds have wings, penguins are birds, penguins cannot fly, in that order. In classical reasoning we can conclude that since birds fly and penguins are birds, then penguins should fly as well. However, with the consideration of the statement that penguins cannot fly we come across an exception within the knowledge base which results in a conclusion that penguins cannot exist[21, 22, 26]. Which of course in reality this is not the case. In defeasible reasoning, this exception is acknowledged and managed through the KLM Framework, named after Kraus, Lehmann and Magidor which introduces the concept of typicality, denoted $\vdash$ [15, 29]. The typicality replaces the material implication of formulas or propositional statements in a knowledge base that give rise to exceptions. In the case of the knowledge base provided above, such a formula would be $b \rightarrow f$, and it would be changed to $b \vdash f$ because though birds fly not all birds fly (penguins). The resulting knowledge base would be a defeasible knowledge and to draw conclusions from it we use defeasible entailment as apposed to the classical entailment defined above. Defeasible entailment is denoted by $\mid\!\sim$, and there no specific methods to determine if defeasible entailment is satisfied . In response to that the KLM proposed properties that must be satisfied in order for the defeasible entailment to hold [6, 13, 15, 17, 20–22, 26]. These are referred to as the LM – Rational, and are as follows [15]:

$$Reflexivity : \mathcal{K} \mid\!\sim \alpha \vdash \alpha$$

$$Leftlogicalequivalence : \frac{\mathcal{K} \mid\!\sim \alpha \leftrightarrow \beta, \mathcal{K} \mid\!\sim \alpha \vdash \gamma}{\mathcal{K} \mid\!\sim \beta \vdash \gamma}$$

$$Rightweaking : \frac{\mathcal{K} \mid\!\sim \alpha \rightarrow \beta, \mathcal{K} \mid\!\sim \gamma \vdash \alpha}{\mathcal{K} \mid\!\sim \gamma \vdash \beta}$$

$$And : \frac{\mathcal{K} \mid\!\sim \alpha \vdash \beta, \mathcal{K} \mid\!\sim \alpha \vdash \gamma}{\mathcal{K} \mid\!\sim \alpha \vdash \beta \wedge \gamma}$$

$$Or : \frac{\mathcal{K} \mid\!\sim \alpha \vdash \gamma, \mathcal{K} \mid\!\sim \beta \vdash \gamma}{\mathcal{K} \mid\!\sim \alpha \vee \beta \vdash \gamma}$$

$$CautiousMonotonicity : \frac{\mathcal{K} \mid\!\sim \alpha \vdash \gamma, \mathcal{K} \mid\!\sim \alpha \vdash \beta}{\mathcal{K} \mid\!\sim \alpha \wedge \beta \vdash \gamma}$$

$$Rationalmonotonicity : \frac{\mathcal{K} \mid\!\sim \alpha \vdash \gamma, \mathcal{K} \not\mid\!\sim \alpha \not\vdash \beta}{\mathcal{K} \mid\!\sim \alpha \wedge \beta \vdash \gamma}$$

2.3.1 Preferential Interpretations. Preferential interpretations, as articulated by KLM [17] and Lehmann and Magidor [20], are based on the preferential semantics initially formulated by Shoham [15, 17] . This semantic framework introduces a structured ordering of valuations, where a preferred valuation is deemed more typical than others. Central to this concept is the notion of "states," each associated with a classical valuation. It is important to distinguish states from valuations, as multiple states may correspond to the same valuation. Consequently, a single interpretation may encompass an infinite number of states. Within this framework, a specific subset of preferential interpretations known as ranked interpretations is of particular interest [21, 22, 26].

Semantically, defeasible reasoning is supported by ranked interpretations. These interpretations establish a hierarchical ordering of interpretations based on their degree of 'typicality'. Each interpretation is assigned a rank within this hierarchy, with ranks ranging from 0 (most typical) to n (least typical), and a reserved rank of ∞ denoting impossibility.

A ranked interpretation is formally defined as a function, denoted as $R : \mathcal{U} \rightarrow \mathbb{N} \cup \infty$, where $\mathcal{U}$ represents the set of interpretations. For an element u in the set $\mathcal{U}$, if $R(u) = 0$, it indicates that u is assigned the highest degree of typicality. Moreover, this interpretation must satisfy the convexity property, which states that for every $i \in \mathbb{N}$, if $u \in \mathcal{U}$ such that $R(u) = i$, then there must be a v such that $R(v) = j$ with $0 \leq j < i$ [7, 21, 22, 26]. The convexity property ensures a structured ordering of interpretations, where each rank builds upon the previous one without gaps, reflecting the relative strength of logical entailment[19, 21, 22, 26]. $R^{\mathcal{K}}$ is sometimes used to denote a ranking interpretation R with respect to knowledge base $\mathcal{K}$[7].

Within this semantic framework, a ranked interpretation, R is said to satisfy a defeasible implication $\alpha \vdash \beta$ if, in the lowest rank where α is satisfied, β holds in all corresponding models within that rank. This is formally denoted as $R \Vdash \alpha \vdash \beta$[7]. In this scenario, R is considered a model of $\alpha \vdash \beta$ [28]. Essentially, R entails $\alpha \vdash \beta$ if and only if the implication $\alpha \rightarrow \beta$ holds true in all the 'best' or most typical interpretations of α [15, 21, 22, 26]. Furthermore, interpretations satisfying standard propositional formulas are defined as those interpretations where the formula holds true across all possible interpretations of a given rank, reflecting the absence of typicality considerations in classical propositional logic [21, 22, 26].

2.4 Rational Closure

Rational closure in defeasible reasoning is a crucial process for deriving conclusions from both explicit and implicit statements within a knowledge base. It allows us to conduct rational inference on defeasible knowledge bases [9], ensuring that they are logically sound and consistent. One key aspect of rational closure is to assign ranking to each in the knowledge base, thereby extending preferential reasoning [21, 22, 26, 29].

Formally, rational closure involves determining whether defeasible implication is entailed by a knowledge base using a method that adheres to the rationality properties proposed by Lehmann and Magidor. These properties ensure that conclusions drawn from defeasible implications remain consistent and coherent, even when exceptions are present. Rational closure is the most conservative form of defeasible entailment [21, 22, 26]. This means that anything entailed by rational closure will also be entailed by the other common forms of defeasible entailment [23]. The process of rational closure can be defined either semantically or algorithmically, as outlined in [7, 21, 22, 26].

2.4.1 Minimal Ranked Entailment. Involves establishing a partial ordering $\preceq_{\mathcal{K}}$ of all ranked models of knowledge base $\mathcal{K}$. The ordering is defined such that for any two ranked interpretations $\mathcal{R}_1$ and $\mathcal{R}_2$, $\mathcal{R}_1 \preceq_{\mathcal{K}} \mathcal{R}_2$ if for every valuation $v \in \mathcal{U}$, $\mathcal{R}_1(v) \leq \mathcal{R}_2(v)$ [7]. The most typical models are ranked lower which leads to the identification of a minimal element [11]. This minimal element is denoted as $\mathcal{R}^{\mathcal{K}}_{RC}$. Minimal ranked entailment, denoted by $\mid\!\sim$, is then formulated based on a defeasible knowledge base $\mathcal{K}$ and its corresponding ranked interpretation $\mathcal{R}^{\mathcal{K}}_{RC}$. For any defeasible implication

$\alpha \vdash \beta, \mathcal{K} \approx \alpha \vdash \beta$ holds if and only if $\mathcal{R}^{\mathcal{K}}_{RC}$ entails $\alpha \vdash \beta$. This approach to entailment, based on ranked interpretations, is shown to have similarities to preferential entailment [21, 22, 26].

2.4.2 Materialisation. To ascertain the exceptional nature of a statement, it is imperative to materialize defeasible statements [9]. This involves defining the counterpart of each defeasible implication $\alpha \vdash \beta$ as the propositional formula $\alpha \rightarrow \beta$. The collection of all such counterparts for $\mathcal{K}$ is represented as $\overrightarrow{\mathcal{K}}$. This is formally denoted as $\overrightarrow{\mathcal{K}} = \{\alpha \rightarrow \beta : \alpha \vdash \beta \in \mathcal{K}\}$. A propositional statement α is deemed exceptional in relation to a knowledge base $\mathcal{K}$ if and only if $\mathcal{K} \models \neg\alpha$, meaning α is false in all the most typical interpretations across every ranked model in $\mathcal{K}$ [21, 22, 26].

2.4.3 Base Rank Algorithm. The base rank algorithm separates formulas in the knowledge base into ranks, assigning ranks based on the exceptional subsets derived from the defeasible statements and classical propositional logic statements. The base rank is the initial step to algorithmically determining whether a defeasible implication is encompassed within the rational closure of a knowledge base. As detailed in [7], the base rank is the first of the two sub-algorithms used for rational closure.

The process starts with the materialization of the knowledge base $\mathcal{K}$ itself, denoted as $\mathcal{E}_0$. Subsequent materializations, $\mathcal{E}_i$, are derived by filtering out statements of the form $\alpha \rightarrow \beta$ from the previous materialization $\mathcal{E}_{i-1}$, where α can be demonstrated as false based on the existing statements. This process continues until no new statements can be derived, with $\mathcal{E}_\infty$ representing the final materialization

A ranking is constructed based on the difference between consecutive materializations. Statements from the original knowledge base $\mathcal{K}$ occupy the bottom rank (infinite rank), while statements become progressively more general in higher ranks. A statement $\alpha \rightarrow \beta$ is assigned a base rank i if it resides in rank i, signifying that in any ranked interpretation derived from the corresponding materialization $\mathcal{E}_i$, $\alpha \rightarrow \beta$ will be valid in at least one interpretation belonging to the most typical rank.

The second sub-algorithm is the process of checking defeasible entailment and is detailed below. Given a defeasible implication $\alpha \vdash \beta$, check if $\neg\alpha$ is entailed by the statements in the infinite rank. Then, systematically eliminate sets of classical implications from higher ranks until identifying a rank where $\neg\alpha$ is not entailed. Finally, assess whether the remaining statements entail $\alpha \rightarrow \beta$. If only the bottom rank remains and the statements therein entail $\neg\alpha$, conclude that $\mathcal{K}$ defeasibly entails $\alpha \vdash \alpha$.

This algorithm ensures that a statement is deemed defeasibly entailed by a knowledge base only if it holds true in accordance with the minimal ranked interpretation of the knowledge base [10, 21, 22, 26].

2.5 Lexicographical Closure

Lexicographic closure is a form of defeasible entailment was first introduced by Lehmann [19]. Lexicographical closure is a non-monotonic reasoning procedure that builds on the concepts of rational closure [8] and refines it [19]. It is used to derive conclusions from a combination of explicit knowledge and default information. Lexicographic closure can be defined in terms of ranked interpretation and ranking formulas. Both of these definitions use a *"seriousness"* ordering, in which we prefer ordering "lighter" set of defaults over a more serious one [19]. Two criteria must be considered when determining the seriousness of two sets: the cardinality and specificity[21, 22, 26].

2.6 Bayesian Inference

Bayesian inference is a method of statistical inference wherein Bayes' theorem is employed to update the probability estimate for a hypothesis as additional evidence or information becomes available. This method offers a means to integrate prior knowledge or beliefs about a hypothesis (prior probability) with new data (likelihood), thereby yielding a revised probability estimate (posterior probability)[27].

Bayesian networks are graphical models that represent probabilistic relationships among a set of variables. These networks consist of nodes, representing variables, and directed edges, indicating probabilistic dependencies between variables. The nodes in a Bayesian network can represent various types of variables, including observable or hidden variables, while the edges encode probabilistic dependencies[3].

The relationship between Bayesian networks and defeasible reasoning is rooted in their complementary characteristics. Bayesian networks furnish a structured framework for probabilistic reasoning, facilitating the integration of prior knowledge and the revision of beliefs in light of new evidence. Consequently, Bayesian networks are well suited for modeling uncertain and intricate domains where reasoning may not adhere strictly to deterministic principles. This project aims to explore the extension of defeasible reasoning through Bayesian inference. One of the optimization techniques to be investigated in this project entails employing Bayesian networks within a defeasible reasoner.

3 PROJECT DESCRIPTION

3.1 Problem Overview

Defeasible reasoning is crucial in the modeling of knowledge with incomplete or uncertain information. However, practical implementations, particularly in the propositional context, remain limited. The field is dominated by theoretical research, leaving gaps in practical, scalable, and efficient methods.

Existing implementations from previous projects—SCADR, SCADR2, and MBDR—show progress. However, there's potential to enhance their scalability and efficiency. Classical reasoners struggle with contradicting information due to their monotonic nature. Description logics, although powerful, pose challenges in direct translation to propositional logic. Given the advancements from previous projects[25] [2], which focused on Rational and Lexicographic Closure, this project aims to not only optimise the existing implementations but also extend defeasible reasoning beyond the limitations of rational closure.

3.2 Motivation

Firstly, there is limited implementation of automated defeasible reasoning. This research project, building on the foundation laid by previous year's projects[2, 25, 29], aims to analyze the efficiency and scalability of these implementations and identify avenues for

further optimization. Additionally, while rational closure has been a commonly employed approach in defeasible reasoning, its conservative nature has spurred exploration into alternative extensions, such as lexicographic closure. However, the current rank-based entailment approach of lexicographic closure [7] presents intractability issues, underscoring the need for exploring alternative computational methods and their implementation. Therefore, this project will make a significant contribution to the research on defeasible reasoning within the field of logic-based artificial intelligence.

4 PROBLEM STATEMENT

In this project, we seek to expand previous attempts at developing defeasible reasoners by extending defeasible reasoning beyond rational closure. We will implement reasoners for algorithms that attempt to extend rational closure. Subsequently, we will conduct an empirical assessment of its algorithmic complexity and compare it with the optimized standard implementation, SCADRV2 by Pillay and Vallabh [25], using a carefully selected set of problem knowledge bases. Additionally, we will explore the integration of rational closure with dynamic Bayesian networks and evaluate its performance against the Naïve Bayesian method proposed by Lang et al.[29] in terms of algorithmic complexity.

4.1 Research Questions

4.1.1 Knowledge Base Generation. In what ways can one create a parameterised knowledge base generator that generates defeasible knowledge bases of varying sizes and distributions (i.e., random, normal) of defeasible statements for stress-testing the rational closure algorithm and its extended algorithms, within a short period relative to existing defeasible knowledge generators?

4.1.2 Debugger. How will the debugger contribute to a better understanding of defeasible reasoning?

4.1.3 Optimisation. How does the algorithmic complexity and execution time of the optimised extension of rational closure compare to previous algorithms[29],[25]?

4.2 Aims

- Creating a knowledge base generator that will be used to create a defeasible knowledge bases with varying sizes and distributions for stress-testing the rational closure algorithm, lexicographic closure algorithm, and their extension within a short period relative to existing defeasible knowledge base generator.
- To explore the improvement of the thread management system of the optimized knowledge base generator introduced by Alec Lang to yield better execution times to account for extremely large numbers of defeasible implications.
- To explore methods for creating complex propositional and defeasible formulas, and parameters that influence ranking.
- Developing a debugger that assists users in understanding defeasible reasoning by displaying the detailed process.

5 METHODS AND PROCEDURES

5.1 Knowledge Base Generator

5.1.1 Methodology. As a starting point, in-depth research on the properties of classic or propositional knowledge bases and how transformations to defeasible knowledge bases occur will be done to determine useful parameters for knowledge base generation. And studying the KLM-rational properties to ensure the construction of a suitable defeasible knowledge base where appropriate inferences can be made. The base ranking system algorithm will be analysed to understand and clearly define how ranks are created and what properties of materialised defeasible statements invoke the addition of ranks. A Clear definition of the structure and rules of the defeasible knowledge base generator will be formed and compared with the existing optimized knowledge base generator to ensure the appropriate principles are followed in the creation of the knowledge base. The optimized knowledge base generator will be studied to determine whether proper and efficient algorithms are employed in knowledge base generation, this including evaluating the thread management system, i.e., the degree of parallelism and concurrency within the system. Some research will be conducted on how to use the Monte Carlo method to introduce some randomness in the knowledge base generator. Further research on how to use dynamic programming in knowledge base generation will conducted to minimize execution time by reusing existing knowledge bases to create larger knowledge bases. Thus a clear final structure of the new knowledge base generator is to be created, considering the properties and parameters of the defeasible knowledge base, the Monte Carlo method for randomization, any thread system optimization detected, and dynamic programming application in the context of the knowledge base generator. The approval will be acquired from the supervisor concerning the designed knowledge base generator and the building of the knowledge base generator in the Java programming language using the logic libraries from the TweetyProject will commence. Lastly, the completed knowledge base generator will be tested by comparing it to Alec Lang's proposed knowledge base generator in terms of execution time (time complexity), degree of randomness, and conformance with the properties of a defeasible knowledge base generator under specific controlled parameters.

5.1.2 Potential Challenges. Identifying all necessary parameters that affect defeasible knowledge base generation and controlling them to see their effect on the generated knowledge base. The use of the Monte Carlo method to provide some level of randomness in knowledge base generation will increase execution time and use more computing resources (memory). The use of dynamic programming to create a knowledge base may negatively impact randomness. However, it is an appropriate way to generate a knowledge base since in the real world information or propositional statements are continuously being added.

5.2 Optimization

5.2.1 Methodology. In the project's initial phase, the focus will be on establishing the theoretical basis for extending rational closure and creating adaptable defeasible knowledge bases. Following this, we will outline formal algorithms. These algorithms will be checked for accuracy and potential areas for optimisation. Practical

implementations will then be carried out to gauge the algorithms' efficiency in terms of time and space. A detailed examination of using dynamic Bayesian networks to perform probabilistic preferential reasoning for rational closure will be undertaken. After gaining insights into the problem's complexities, we will design a theoretical framework for probabilistic preferential reasoning. Upon confirming the feasibility with the supervisor, we'll draft initial specifications for the advanced ranking and entailment algorithms, which will undergo a review before moving on to the implementation and optimization phase. Once the foundational algorithms are developed, we'll build defeasible reasoners using libraries and tools similar to those used in SCADR and SCADRV2 [25],[2]. Other possible optimisation techniques will also be explored. The revised implementation that uses an extended version of rational closure will be evaluated against SCADRV2 reasoners and Lang et al.'s algorithms [25], [29], with a focus on space and time complexity. This evaluation may involve creating specific knowledge bases manually to showcase how Bayesian inference via dynamic networks influences rational closure.

5.2.2 Possible Challenges. Implementing reduced minimal ranked entailment and dynamic Bayesian networks could lead to algorithms with high computational complexity. Optimizing these algorithms to be efficient in terms of time and space may be non-trivial. Conducting a comprehensive empirical assessment involves designing experiments that can effectively compare the performance of the new approach with existing methods like SCADRV2 and Lang et al.'s algorithms. Ensuring the fairness and validity of these comparisons can be challenging.

5.3 Debugger

5.3.1 Methodology. The debugger will be designed to interact seamlessly with different defeasible reasoning algorithms, such as rational closure and lexicographic closure, and will be integrated with existing defeasible reasoners. The debugger will have a console-based and a graphical user interface versions. The code of one or two existing defeasible reasoners will be extended; this is the only code that will be extended. Users will be able to define and refine knowledge bases, submit defeasible queries, and view detailed explanations of query results, including rankings. The debugger will also provide insights into the time taken for different algorithms and each step of the reasoning process, which will differentiate it from previous implementations. The implementation will be in Java, utilizing the Java Swing GUI framework, with each component built as a separate module. Integration testing will ensure smooth interaction between the debugger and the defeasible reasoners. The implementation will undergo evaluation to validate its accuracy and usefulness, involving experts in the field, such as Master's students or our supervisor, who will assess its functionality and effectiveness.

5.3.2 Possible Challenges. Integrating the debugger with various existing defeasible reasoners presents a significant challenge due to the potential differences in their implementation approaches. It may also be difficult to understand the existing implementations of defeasible reasoners, especially if the code lacks sufficient documentation.

6 RELATED WORK

This project builds on work by [25],[2], [29] which aimed to extend Casini et al.'s work in [7].These are based on the KLM Framework[17], [20], [19].In the context of knowledge base generation this paper [12] used the concept of three types of atoms to in order to build a defeasible knowledge base with multiple ranks. The anchor atoms to control the number of defeasible ranks, extension atoms to populate defeasible ranks and a single contradiction atom to form inconsistencies [12]. The EXTRC 2023 project involved creation of complex formulas and optimization of the knowledge base generator using multi-threading[18].

7 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Our research does not require any testing, or collection of data of any kind on human subjects and thus requires no ethical clearance, nor does it infringe on the POPI Act [16]. It however builds up on work that has already been done in defeasible reasoning, by optimizing proposed algorithms, implementing proposed designs, and using previous work as the benchmarks for testing purposes. Any algorithm or design used is publicly released academic literature, properly referenced, and acknowledged. Thus we are not infringing any intellectual property rights. We are using the Java Development Kit (JDK) to develop our algorithms, it uses the GNU General Public License (GPL) version 2 license with the Classpath Exception [24] which allows us to develop and distribute code written using Java libraries without being constrained by GPL requirements.

8 ANTICIPATED OUTCOMES

- Extended Rational Closure Algorithm: We aim to develop a robust and efficient algorithm for extended rational closure that integrates reduced minimal ranked entailment and dynamic Bayesian networks.
- Probabilistic Preferential Reasoning Framework: We plan to create a theoretical and practical framework for probabilistic preferential reasoning within each level of typicality using dynamic Bayesian networks
- Defeasible Reasoner: A software tool implementing the developed algorithms and frameworks, providing a user-friendly interface for creating, managing, and querying defeasible knowledge bases.
- Knowledge Base Generator: We plan to develop a defeasible knowledge base generator which uses dynamic programming.
- The development of a debugger that displays the detailed defeasible reasoning process for a given query..

8.1 Key Success Factors

- Algorithmic Efficiency: Successfully developing algorithms that are both theoretically sound and computationally efficient will be crucial. Optimizing the algorithms to handle large-scale defeasible knowledge bases while maintaining high performance will be a key success factor.
- Empirical Validation: Conducting rigorous empirical evaluations that demonstrate the practical benefits and limitations of the proposed methods will be essential. Validating the

algorithms' performance against existing benchmarks will provide credibility and confidence in the research outcomes.
- User-Friendly Software: Developing a user-friendly and intuitive software tool that enables researchers and practitioners to easily create, manage, and query defeasible knowledge bases will be essential for the adoption and impact of the research.
- Knowledge Base Generation: Creating a generator for defeasible knowledge bases that enhances the generation of complex formulas and incorporates an efficient thread management system which will be observed by measuring lower execution time, and produces random knowledge bases that emulate real-world scenarios which will be measured by deviation on formulas in ranks relative to some applied distribution.
- Creating a debugger that assists users in enhancing their understanding of defeasible reasoning.

9 PROJECT PLAN

9.1 Risks

As a team we carefully considered the risks that would impede the success of the project, analysed their impact (ranging from very low to very high), probability of occurrence (ranging from very unlikely to very likely), and came up with plans to mitigate, monitor, and manage the risks to maximize project success. Detailed entries of the identified risks can be found in Appendix A.

9.2 Required Resources

- Papers on previous projects (SCADR,SCADR2, and the EXTRC 2023 version), along with their algorithmic implementations [2, 18].
- Hardware: Laptops with sufficient processing power and memory.
- Software: A development environment such as Visual studio or Intellij IDEA, and the Java open source software and Logics libraries from the Tweetyproject [2, 18].

9.3 Deliverables

Deliverable	Due Date
Literature Review Scaffolds	7 March 2024
Literature Reviews	25 March 2024
Research Plan	19 April 2024
Project Proposal Draft	22 April 2024
Proposal Presentation Slides	23 April 2024
Project Proposal Final	30 April 2024
Project Final Paper Draft	23 August 2024
Project Final Paper	30 August 2024
Project Final Code	9 September 2024
Project Poster	27 September 2024
Project Website	4 October 2024

Table 1: Deliverables

9.4 Timeline

The timeline or sequence of tasks that will be completed in this project is shown using a Gantt chart found in Appendix B.2.

9.5 Work Allocation

The first phase of our work will be implementing the base algorithms for the knowledge base generator, rational closure, and lexicographic closure. These will act as a benchmark for testing improvements and optimization success and the implementation task will be done collectively.

Student	Work Allocation
Mamodike Sadiki	Developing a defeasible knowledge base generator with improved complex formula generation, thread management system and generates random knowledge bases that mimic the real world.
Maqhobosheane Mohlerepe	The development of prototypes for both naïve and optimized defeasible reasoners using Bayesian inference, coupled with extensive research on potential optimization strategies and performance evaluation of the reasoners against each other
Thabo Vincent Moloi	Create a debugger for defining and refining knowledge bases, submitting defeasible queries, and viewing results, including the steps involved in the defeasible reasoning process.

Table 2: Work distribution

REFERENCES

[1] Grigoris Antoniou and Mary-Anne Williams. 1997. *Nonmonotonic Reasoning*. MIT Press. https://doi.org/10.7551/mitpress/5040.001.0001

[2] Aidan Bailey. 2021. Scalable Defeasible Reasoning. Project Paper.

[3] Irad Ben-Gal. 2008. Bayesian Networks. In *Encyclopedia of Statistics in Quality and Reliability*, Fabrizio Ruggeri, Ron S. Kenett, and F. W. Faltin (Eds.). John Wiley & Sons. https://doi.org/10.1002/9780470061572.eqr089

[4] Ronald J Brachman and Hector J Levesque. 2004. *Knowledge Representation and Reasoning*. Elsevier. https://doi.org/10.1016/B978-155860932-7/50086-8

[5] H. K. Büning and T. Lettmann. 1999. *Propositional Logic: Deduction and Algorithms*. Cambridge University Press.

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

[7] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2019. Taking defeasible entailment beyond rational closure. In *European Conference on Logics in Artificial Intelligence*. 182–197.

[8] Giovanni Casini and Umberto Straccia. 2012. Lexichographic Closure for Defeasible Description Logics. *CEUR Workshop Proceedings* 969, xx–xx.

[9] Victoria Chama. 2019. *Explanation For Defeasible Entailment*. Master's Thesis. Faculty of Science, University of Cape Town.

[10] Michael Freund. 1998. Preferential reasoning in the perspective of Poole default logic. *Artificial Intelligence* 98, 1 (1998), 209–235. https://doi.org/10.1016/S0004-3702(97)00053-2

[11] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence* 226 (2015), 1–33.

[12] Joel Hamilton, Daniel Park, Aidan Bailey, and Thomas Meyer. 2021. An Investigation into Optimisations of Defeasible Reasoning Algorithms. In *SACAIR'21 Proceedings Volume II*. University of Cape Town and Centre for Artificial Intelligence Research (CAIR), Cape Town, South Africa.

[13] Michael Harrison and Thomas Meyer. 2020. DDLV: A system for rational preferential reasoning for datalog. *South African Computer Journal* 32, 2 (2020). https://doi.org/10.18489/sacj.v32i2.850

[14] Internet Encyclopedia of Philosophy. [n. d.]. *Propositional Logic*. https://iep.utm.edu/propositional-logic-sentential-logic/ Accessed: Mar. 26, 2024.

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

[16] Agbor Kandeh, Reinhardt Botha, and Lynn Futcher. 2018. Enforcement of the Protection of Personal Information (POPI) Act: Perspective of data management professionals. *South African Journal of Information Management* 20, 1 (2018), 9. https://doi.org/10.4102/sajim.v20i1.917

[17] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44, 1-2 (1990), 167–209. https://doi.org/10.1016/0004-3702(90)90101-5

[18] Alec Lang. 2023. Extending Defeasible Reasoning Beyond Rational Closure. Project Paper.

[19] Daniel Lehmann. 1995. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence* 15, 1 (01 03 1995), 61–82. https://doi.org/10.1007/BF01535841

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

[21] Maqhobosheane Mohlerepe. 2024. Extending Defeasible Reasoning Beyond Rational Closure. Literature review.

[22] Thabo Vincent Moloi. 2024. Extending Defeasible Reasoning beyond Rational Closure. Literature Review.

[23] Matthew Morris, Tala Rossa, and Thomas Meyer. 2020. Algorithmic definitions for KLM-style defeasible disjunctive Datalog. *SACJ* 32, 2 (December 2020). Research Article.

[24] Oracle Corporation. 1991. Java Development Kit (JDK). GNU General Public License (GPL) version 2 with Classpath Exception. Available at: https://openjdk.org/legal/gplv2+ce.html. Accessed on 21 April 2024..

[25] Evashna Pillay. 2022. *An Investigation into the Scalability of Rational Closure V2*. Ph. D. Dissertation. University of Cape Town, Cape Town.

[26] Mamodike Sadiki. 2024. A Literature Review on Extending Defeasible Reasoning Beyond Rational Closure. Literature review.

[27] Anil Seth. 2021. *Being You: A New Science of Consciousness*. Penguin Publishing Group. 352 pages.

[28] Yoav Shoham. 1987. A semantical approach to nonmonotonic logics. In *Logic in Computer Science*. https://api.semanticscholar.org/CorpusID:117644704

[29] Luke Slater and Thomas Meyer. 2023. Extending Defeasible Reasoning Beyond Rational Closure. In *Artificial Intelligence Research*, Anban Pillay, Edgar Jembere, and Aurona J. Gerber (Eds.). Springer Nature Switzerland, Cham, 151–171.

A RISKS

A.1 Risk Identification

Risk Identification			
ID	Risk description	Probability	Impact
1	Unavailable supervisor.	Low	Moderate
2	Lack of understanding of project requirements.	Low	Very high
3	Unfinished project due to lack of proper time management.	Moderate	Very high
4	Hardware and Software failures	Low	High
5	Communication issues within the team.	Moderate	Very high
6	Team member unavailable to perform their assigned tasks, i.e., due to illness, deregistering or any other reason. Thus affecting team progress.	Unlikely	Very low
7	No further optimization of algorithms possible.	Likely	High
8	Incorrect estimated completion time and complexity of required tasks in the project.	Moderate	High
9	Lack of access to resources, e.g., papers on related work.	Low	Moderate
10	Using non-peer reviewed papers as a base for research may lead to some misinformation.	Low	Very high

Table 3: Risk identification.

A.2 Risk Management

Risk Mitigation, Monitoring and Management Strategies			
ID	Mitigation	Monitoring	Management
1	Scheduling meeting times ahead of time, or deciding on a recurring meeting time period throughout the project duration.	Keeping track of how often meetings are missed, or rescheduled by the supervisor.	Discussing alternative meeting channels (e.g., online), or involving a supervisor selected co-supervisor to stand in for the supervisor when they are not available.
2	Consulting the supervisor and reading up on related topics	Reading supervisor comments or feedback on project or background understanding from submitted documents.	Reading up on more papers related to the subject and consulting the supervisor to get a clearer understanding of the project requirements.
3	Following the project plan to ensure tasks are performed on time.	Checking if tasks are being completed on time, or keeping track of how far behind or ahead the team is in terms of time.	Clearly communicating this to the supervisor, deciding on the main tasks that students should focus on as, and clearly explaining this on the final project report.
4	Using hardware equipment as laptops (including accessories like chargers) in safe environments, carrying them in protective cases, and using compatible, updated software.	Keeping track of the health of hardware equipment (including accessories), and any warnings about software.	In terms of hardware failure repair of equipment is necessary and in the case of software failure using compatible versions of the software or virtual machines in order to work with some software would be the best way to manage the issue.
5	Creating a team communication platform, and providing rules for communication. During meetings talking points should be clarified.	Ensuring communication rules or talking points are followed and relevant topics are discussed	Having the supervisor facilitate communication in order to resolve communication issues.
6	A clear separation of tasks and having no dependency between tasks assigned to different team members.	Checking if there are tasks that evolve over that are dependent on tasks performed by other team members	Coming up with a new task assignment strategy that minimizes overlapping of tasks.
7	By doing thorough research on optimization techniques earlier in the project.	By continuously checking how proposed optimization techniques measure relative to the base algorithms.	Noting optimization issues in the final project report and possibly changing project focus.
8	Having granular description of tasks to allow appropriate time estimation, and using that to build the Gantt chart	Checking if tasks are completed on time using the Gantt chart.	Assessing why tasks are not completed in time, e.g. lack of knowledge or availability of required resources(e.g., no laptop). Then coming up with a suitable solutions as researching in how to do tasks or consulting supervisor and for unavailability of laptop using computers in UCT computer labs.
9	Ensuring publicly available papers related to the research are used.	Checking whether publicly available papers address the research topic to the full extent and whether subscription fees are required to access papers.	Ask supervisor for papers that you cannot access that they have access to or relatively equivalent papers.
10	Ensuring publicly available peer-reviewed papers related to the research are used.	Continuously checking if papers that are used are peer-reviewed.	The team must get approval to use the paper from the supervisors and clearly note that such a paper has been used.

Table 4: Risk management.

B TIMELINE

B.1 Milestones and Tasks

Tasks	Start Date	End Date
Literature Review	22/02	25/03
Project Proposal	**01/04**	**30/04**
Project Proposal Draft	01/04	22/04
Project Presentation	01/04	22/04
Final Project Proposal	24/04	30/04
Implementing Foundational Defeasible Reasoner	**01/05**	**22/05**
Research	01/05	08/05
Implement and test the algorithm	08/05	18/05
Evaluate the implementation	18/05	22/05
Implementing Knowledge Base Generator	**23/05**	**30/07**
Researching on Knowledge base generation	23/05	26/05
Researching on optimization of Knowledge base generator	27/05	30/05
Researching on randomization techniques	31/05	02/06
Designing Knowledge base generator to be implemented	03/06	09/06
Implement the design	10/06	13/07
Evaluating the Knowledge base generator	14/07	30/07
Optimizing Algorithms	**23/05**	**30/07**
Theoretical research on potential optimisations	23/05	06/06
Formalise optimisations with algorithms	07/06	20/06
Algorithm Review	07/06	20/06
Implement and test optimised algorithms	20/06	20/07
Evaluate optimised reasoner	21/07	30/07
Implement a debugger	**23/05**	**26/07**
Set up existing reasoners	23/05	27/05
Design and develop CLI version	28/05	19/06
Design and develop GUI version	20/06	11/07
Integrate the debugger with reasoners	12/07	19/07
Evaluate the debugger	22/07	26/07
Final Paper	**01/08**	**30/08**
Scaffold	01/08	01/08
Write a draft	01/08	15/08
Incorporate Supervisor Feedback into Final Paper	15/08	30/08
Project Demonstration	**16/09**	**20/09**
Project Poster	**20/09**	**27/09**
Project Website	**27/09**	**04/10**

Table 5: Milestones and tasks.

A Proposal on Extending Defeasible Reasoning Beyond Rational Closure

B.2 Gantt Chart