Extending Defeasible Reasoning Beyond Rational Closure

Project Proposal

Alec Lang
University of Cape Town
Cape Town, South Africa
lngale007@myuct.ac.za

Luke Slater
University of Cape Town
Cape Town, South Africa
sltluk001@myuct.ac.za

David Pullinger
University of Cape Town
Cape Town, South Africa
plldav013@myuct.ac.za

ABSTRACT

Defeasible reasoning is a vital component of many intelligent systems, as it provides a way to reason about uncertain or exceptional information. Rational closure has been at the forefront of research in defeasible reasoning, serving as a basis for a class of rational defeasible entailment relations. This project aims to provide insight into previously unexplored methods of generating these rational defeasible entailment relations, via Bayesian Inference and user-defined rankings. Finally, a parameterised tool for creating large, defeasible knowledge bases will be created to facilitate testing and validation of such defeasible entailment relations. The findings of this project can provide insights into the characteristics, advantages, and limitations of these forms of entailment, and how they compare to rational closure and lexicographic closure.

CCS CONCEPTS

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

KEYWORDS

artificial intelligence, knowledge representation and reasoning, defeasible reasoning, Rational Closure, Lexicographic Closure, Bayesian inference, defeasible knowledge base generation

1 INTRODUCTION

Knowledge representation and reasoning is an essential component of artificial intelligence, and allows intelligent systems to store, manipulate, and produce new information from existing knowledge. Classical reasoning, based on propositional logic, is an effective way to reason with complete and consistent information. However, real-world scenarios often involve uncertain, incomplete, or inconsistent data, requiring more flexible reasoning methods.

Defeasible reasoning allows for conclusions to be withdrawn or revised in light of new evidence, making it more suitable for handling exceptions and contradictions. Nonmonotonic logics are designed to accommodate such defeasible reasoning. The Kraus-Lehmann-Magidor (KLM) framework [8] is a prominent approach to defeasible reasoning, which incorporates preferential reasoning to model the typicality of different situations or objects. By establishing an order of preference among possible worlds, the KLM framework allows for the expression of defeasible implications, enabling the inference of conclusions that hold "typically" rather than absolutely.

In this project, we aim to build upon the KLM framework, and related techniques such as rational and lexicographic closure, by developing novel extensions of rational closure using Bayesian inference and user-defined rankings. Moreover, we seek to address the need for a comprehensive and parameterized defeasible knowledge base generator, which will facilitate the analysis and evaluation of new entailment relations in the field of defeasible reasoning.

2 BACKGROUND

2.1 Propositional Logic

2.1.1 Syntax. Propositional logic focuses on fundamental units of knowledge called *atoms*. The set of all possible atoms is denoted by $\mathcal{P}$. By connecting atoms with zero or more logical operators ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), we create *formulas*. The set of all such formulas is denoted by $\mathcal{L}$ [2].

2.1.2 Semantics. Propositional logic relies on assignments, also known as valuations, interpretations, or "worlds", that map atoms to truth values. When an atom, or formula, α is true in some valuation u, it is said to be satisfied. This is expressed as $u \Vdash \alpha$ [2].

Consider $\mathcal{U}$ as the set of all valuations. If for every valuation $u \in \mathcal{U}$ where $u \Vdash \alpha$ also satisfies $u \Vdash \beta$, then β is considered a logical consequence of α. This relationship is represented as $\alpha \models \beta$. A valuation that satisfies every statement in a knowledge base $\mathcal{K}$ is called a *model* of $\mathcal{K}$.

2.1.3 Entailment. A statement α is entailed by a knowledge base $\mathcal{K}$ if all models of $\mathcal{K}$ are simultaneously models of α. In the context of classical propositional logic, to verify whether a statement α is entailed by a knowledge base $\mathcal{K}$, we perform the following steps:

- Conjunctively combine all formulas $\beta \in \mathcal{K}$.
- Append the negation of α to the combined formula.
- If the resulting formula is unsatisfiable, then $\mathcal{K} \models \alpha$.

2.2 Defeasible Reasoning

Classical reasoning presupposes absolute truth, yet the real world is full of contradictions and exceptions. Defeasible reasoning addresses this by permitting the retraction of statements under specific conditions. It essentially involves reasoning with incomplete or uncertain information, acknowledging multiple possible "parallel" worlds without identifying the precise one.

Kraus, Lehmann, and Magidor devised the KLM framework for such situations, based on *preferential reasoning* [8]. This approach assigns preferences to each possible world, determining statement validity by evaluating its truth in the most preferred "minimal" worlds.

2.3 Preferential Reasoning

Classical monotonic logics are inadequate for representing exceptions and conducting defeasible reasoning. Preferential reasoning, first introduced by Shoham [13, 14], and an important foundation for rational closure, addresses this by ordering valuations based on preference or typicality. Introducing defeasible implication, $\alpha \mid\!\sim \beta$, to signify "typically, if α, then β" [8].

2.3.1 KLM Postulates. In examining various forms of defeasible entailment, Lehmann and Magidor advocate a number of properties [11]. Each rule comprises multiple premises (indicated above) and a singular conclusion (indicated below). The central tenet of each postulate posits that if the premises are deemed valid, the conclusion must necessarily hold true as well. For an entailment relation to be deemed preferential, it must satisfy each of these properties:

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

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

$$\text{(And)}\ \frac{\mathcal{K} \models\!\approx \alpha \mid\!\sim \beta, \mathcal{K} \models\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \models\!\approx \alpha \mid\!\sim \beta \wedge \gamma} \qquad \text{(Ref)}\ \mathcal{K} \models\!\approx \alpha \mid\!\sim \alpha$$

2.3.2 Preferential Interpretations. Preferential interpretations, denoted by $\mathcal{P}$, consist of a triple $\langle S, l, \prec \rangle$. S represents possible states, with l mapping them to valuations, hence, multiple states can correspond to the same valuation. The strict partial order $\prec$ ranks states by preference [7]. In a preferential interpretation, $\alpha \mid\!\sim \beta$ holds if all minimal states satisfying α also satisfy β.

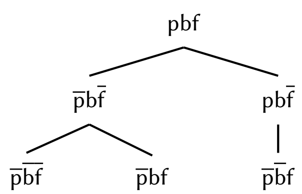

Figure 1: Preferential interpretation for $\mathcal{K}$ [7].

Example 2.1. Let $\mathcal{K} = \{\mathsf{p} \rightarrow \mathsf{b}, \mathsf{b} \mid\!\sim \mathsf{f}, \mathsf{p} \mid\!\sim \neg\mathsf{f}\}$, signifying the knowledge that penguins are birds, most birds can fly, and penguins typically cannot fly. Figure 1 illustrates a preferential interpretation for $\mathcal{K}$. Assessing satisfiability, all $\mathcal{K}$ statements hold, rendering the preferential interpretation a *model* of $\mathcal{K}$. Further examination of $\mathcal{P}$ can reveal additional satisfiable defeasible implications, such as `flies` $\mid\!\sim_{\mathcal{P}}$ $\neg$`penguin`.

2.4 Rational Closure

Although preferential reasoning offers a robust framework for reasoning with defeasible knowledge bases, it has been shown to still be monotonic. Rational closure, the first truly nonmonotonic approach, serves as the nonmonotonic core of defeasible reasoning. This method employs ranked interpretations, which organize valuations into modular tiers instead of a graph structure [7]. They are a subset of preferential interpretations where any two worlds are either equal in rank, or one is preferred over the other (in contrast, preferential interpretations can have incomparable elements [7], such as $\overline{p}\overline{bf}$ and $\overline{p}bf$ in Figure 1).

2.4.1 Rational Monotonicity. Rational monotonicity is a property stipulating that new information should not invalidate a conclusion, unless it explicitly contradicts existing information. This enables more speculative inferences by allowing the addition of new information without invalidating established conclusions, provided no direct contradictions arise. An entailment relation that complies with all KLM postulates, including rational monotonicity, is considered LM-rational [11]. The foundation for these entailment relations is rational closure [3].

$$\text{(RM)}\ \frac{\mathcal{K} \models\!\approx \alpha \mid\!\sim \gamma, \mathcal{K} \models\!\approx \alpha \mid\!\sim \neg\beta}{\mathcal{K} \models\!\approx \alpha \wedge \beta \mid\!\sim \gamma}$$

2.4.2 Minimal Ranked Entailment. Minimal ranked entailment provides a semantic characterization of rational closure by establishing a preferential ordering over all ranked interpretations $\mathcal{R}$ of a knowledge base $\mathcal{K}$: $\mathcal{R}_1 \preceq_{\mathcal{K}} \mathcal{R}_2$ iff $\mathcal{R}_1(u) \leq \mathcal{R}_2(u)$ for every valuation u [8]. Giordano et al., in [5], demonstrated the existence of a minimal element in this partially ordered set, $\mathcal{R}^{\mathcal{K}}_{RC}$, allowing a formal definition of minimal ranked entailment: $\mathcal{K} \models\!\approx_{RC} \alpha \mid\!\sim \beta$ iff $\mathcal{R}^{\mathcal{K}}_{RC} \Vdash \alpha \mid\!\sim \beta$.

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

Figure 2: Minimal ranked model, $\mathcal{R}^{\mathcal{K}}_{RC}$, for $\mathcal{K}$ [3].

Example 2.2. Suppose we incorporate the statement `bird` $\mid\!\sim$ `wings` into knowledge base $\mathcal{K}$ from Example 2.1. Figure 2 displays the minimal ranked model for the updated $\mathcal{K}$. To determine whether `penguins` $\mid\!\sim$ `wings` is entailed by $\mathcal{R}^{\mathcal{K}}_{RC}$, we examine the minimal interpretations satisfying p: $pb\overline{f}\overline{w}$ and $pb\overline{f}w$. Consequently, $\mathcal{K}$ does not entail `penguins` $\mid\!\sim$ `wings`. This exemplifies rational closure's prototypical nature, which states that while typical members of a class inherit all of its properties, atypical members may not [3].

2.5 Lexicographic Closure

Lexicographic closure is a refinement of rational closure that organizes exception sets hierarchically based on specificity and cardinality [7]. Compared to rational closure alone, lexicographic closure considers not only the minimal sets of exceptions but also their relative size and importance, resulting in more "adventurous" inferences.

Semantically, lexicographic closure can be characterized with respect to minimal ranked entailment: $v \preceq^{\mathcal{K}}_{LC} u$ if $\mathcal{R}^{\mathcal{K}}_{RC}(u) = \infty$, or $\mathcal{R}^{\mathcal{K}}_{RC}(v) < \mathcal{R}^{\mathcal{K}}_{RC}(u)$, or $\mathcal{R}^{\mathcal{K}}_{RC}(v) = \mathcal{R}^{\mathcal{K}}_{RC}(u)$ and v satisfies more formulas than u [3]. The resulting lexicographic ranked model is denoted as $\mathcal{R}^{\mathcal{K}}_{LC}$. The lexicographic closure of $\mathcal{K}$ is then formally defined as follows: $\mathcal{K} \models\!\approx_{LC} \alpha \mid\!\sim \beta$ iff $\mathcal{R}^{\mathcal{K}}_{LC} \Vdash \alpha \mid\!\sim \beta$.

Example 2.3. Figure 3 demonstrates the lexicographic ranked model of knowledge base $\mathcal{K}$ from Example 2.2. To ascertain if `penguins` $\mid\!\sim$ `wings` is entailed by $\mathcal{R}^{\mathcal{K}}_{LC}$, we consider the minimal

5	$pbf\overline{w}$
4	$pbfw$
3	$pb\overline{f}\overline{w}$ $\overline{p}b\overline{f}\overline{w}$
2	$pb\overline{f}w$ $\overline{p}b\overline{f}w$
1	$\overline{p}bf\overline{w}$
0	$\overline{p}\overline{b}\overline{f}\overline{w}$ $\overline{p}\overline{b}\overline{f}w$ $\overline{p}\overline{b}f\overline{w}$ $\overline{p}\overline{b}fw$ $\overline{p}bfw$

Figure 3: Lexicographic ranked model, $\mathcal{R}^{\mathcal{K}}_{LC}$, for $\mathcal{K}$ [3].

interpretations satisfying p: $pb\overline{f}w$. As w holds true in this valuation, the statement is entailed by $\mathcal{R}^{\mathcal{K}}_{LC}$. This exemplifies lexicographic closure's presumptive nature, which presumes that the properties of a class apply to all members unless contradictory evidence exists [3].

2.6 Bayesian Inference

Kraus, Lehmann, and Magidor present a valuable framework for reasoning under conditions of uncertainty. Bayesian inference, or abductive reasoning, aims to identify the most plausible explanation for a collection of observations, particularly when these observations are incomplete, uncertain, or ambiguous [12]. Like rational and lexicographic closure, Bayesian inference can also get things wrong. It relies on probability to determine the "best explanation", and can be viewed as reasoning in reverse, from observed effects to their most probable causes.

2.6.1 Bayes' Rule. Bayes' rule serves as a mathematical approach to transition from our current knowledge (the *prior*) to our subsequent beliefs (the *posterior*) by incorporating newly acquired information (the *likelihood*), thus updating the probabilities for each hypothesis [12]:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{1}$$

In this equation, $P(H)$ denotes our initial belief about the probability of an event occurring, prior to considering any new evidence. The likelihood, represented by $P(E|H)$, is the probability of observing the new evidence, assuming the hypothesis is accurate [4]. The posterior is the revised probability of the hypothesis being true after accounting for the new evidence. Lastly, $P(E)$ represents the probability of the evidence, independent of the hypothesis, also known as the marginal probability of the evidence.

2.6.2 Probabilistic Entailment. One way to check if a statement $\alpha \mid\!\sim \beta$ is probabilistically entailed using Bayesian inference, is to consider α as evidence, and the different possible valuations of the remaining atoms as hypotheses. For instance, using the knowledge base from Example 2.2, to evaluate if `penguin` $\mid\!\sim$ `wings` is entailed by $\mathcal{K}$, treat p as evidence and $bf\overline{w}$, bfw, $b\overline{f}\overline{w}$, and $b\overline{f}w$ as hypotheses. The statement is probabilistically entailed by $\mathcal{K}$ if the most probable hypothesis satisfies `wings`.

Probabilistic entailment is highly likely to breach rational monotonicity, consequently precluding the formation of a rational entailment relation. Nevertheless, it is expected to comply with the other postulates, implying that the implementation of any probabilistic defeasible entailment relation necessitates the use of preferential reasoning.

3 PROJECT DESCRIPTION

3.1 Overview

The representation theorem states that a defeasible entailment relation is LM-rational if it can be defined by a ranked interpretation [11]. Casini et al. in [3] question the sufficiency of the KLM postulates in isolating useful ranked models. The proposed solution, based on rational closure as the nonmonotonic core of defeasible entailment, considers entailment relations entailing at least as much as rational closure as being reasonable.

This project aims to build on these insights by developing two novel extensions of rational closure, via Bayesian inference and user-defined rankings. Furthermore, we aim to address a considerable need within this research field for a comprehensive and parameterised defeasible knowledge base generator, which will enable the analysis of novel entailment relations.

3.2 Motivation

Although rational closure has been commonly utilized in defeasible reasoning, its conservative nature has led to the investigation of alternative extensions, such as lexicographic closure. However, these extensions have not been extensively studied, resulting in a gap in the literature. To address this gap, we aim to explore extensions of rational closure using Bayesian inference and user-defined rankings. Furthermore, we aim to develop a parameterized defeasible knowledge base; by doing so, we hope to provide a comprehensive evaluation and testing tool for defeasible entailment relations.

4 PROBLEM STATEMENT

In this research project, we aim to extend rational closure's utility by building on Casini et al.'s insights in [3]. We will develop two novel methods for generating parameterized refinements to the ranking of interpretations within each typicality level of rational closure. One that utilizes user defined rankings, and one that utilizes Bayesian inference. Furthermore, we will address challenges in testing and evaluating new forms of defeasible entailment due to the limited availability of large, complex knowledge bases.

4.1 Aims

Our aims are to:

Alec Lang

- Create a generator for defeasible knowledge bases that is capable of efficiently producing knowledge bases with different configurations for the purpose of testing defeasible entailment relations.
- Evaluate the impact of different knowledge base configurations on the efficiency of generation.

David Pullinger

- Develop a tool that will allow users to rank the defeasible implications of a knowledge base.
- Develop a tool that computes rational closure and the user-defined closure for a knowledge base, identifying and highlighting areas of disagreement between them.

- Design an algorithm which interactively resolves disagreements between rational closure and the user-defined closure.

Luke Slater

- Leverage Bayesian insights to develop a theoretical framework for performing probabilistic preferential reasoning within each typicality level of a minimal ranked model.
- Implement algorithms to calculate the probabilistically refined minimal ranked model, and compute entailment.
- Assess the defeasible reasoning capabilities of the integrated approach and examine its differences and synergies with the prototypical characteristics of traditional rational closure.

4.2 Research Questions

Our work will aim to address the following research questions:

Alec Lang

- What key factors and relationships should be considered in the development of a defeasible knowledge base generator to effectively produce ranked statements that conform to user-specified preferences, such as the desired number of ranks and the distribution of statements within each rank?
- How can BaseRank and Lexicographic closures ranking function be leveraged to allow for the development of a more versatile knowledge base ranking method?
- How do different number of ranks, number of statements and distributions of statements over the ranks impact the efficiency of generating knowledge bases of various sizes?

David Pullinger

- How do user-defined rankings of defeasible implications affect the ranks of valuations in the minimal ranked model of rational closure?
- How can we effectively integrate user-defined rankings of defeasible implications to resolve disagreements between rational closure and lexicographic closure in a defeasible knowledge base?
- How does the inclusion of user-defined rankings of defeasible implications impact the consistency, coherence, and overall performance of the resulting rational defeasible entailment relation?

Luke Slater

- How can Bayesian insights contribute to the refinement of rational closure by facilitating probabilistic preferential reasoning within each level of typicality in a minimal ranked model?
- How does the integration of Bayesian inference with rational closure affect reasoning, efficacy, and performance compared to employing rational closure alone?

5 METHODS AND PROCEDURES

5.1 Overview

In the initial stages, the project will primarily focus on developing the theoretical foundations for the proposed extensions to rational closure, and generating customizable defeasible knowledge bases. Subsequently, formal specifications of the required algorithms for each project will be devised. These algorithms will then undergo evaluation to ensure correctness and identify potential optimization opportunities. Finally, practical implementations will be conducted to assess and compare the algorithms' time and space complexity.

5.2 Theoretical Foundation and Algorithm Design

Alec Lang The first step will be to gain an understanding of classical knowledge bases, how they are formed and structured. We will then take a look at KLM-style defeasible reasoning and how its properties are applied to classical knowledge bases to create a defeasible knowledge base. Research will be conducted on knowledge base ranking algorithms such as rational closures' BaseRank, as this will be vital in understanding how defeasible statements affect the structure of a defeasible knowledge base. As well as this we will investigate randomness in data generation, with the intent to add some degree of pseudo randomness to the final generator. The focus will be on creating an algorithm that allows users to indicate preferences regarding the number of rankings and statements, distribution of statements within each rank, and the complexity of defeasible statements. Using this as input we will be able to generate knowledge bases that are unique in both structure and data. Draft versions of the generator will be designed and presented to the supervisor for review.

David Pullinger A review of the existing algorithms for rational closure and lexicographic closure will be conducted. The focus will be on creating a defeasible entailment relation by allowing the user to indicate how statements within the knowledge base should be ranked. When designing the algorithm, a vital step will be determining when the user should be prompted to rank defeasible implications and which defeasible implications are relevant enough to be ranked. Thereafter, the resulting entailment relation will be tested to determine if it falls in the class of rational defeasible entailment relations. Finally, draft versions of the algorithm will be presented to the supervisor for review, and any critiques will be subsequently incorporated.

Luke Slater A comprehensive analysis of rational closure and Bayesian inference will be undertaken to determine the most optimal form of Bayesian inference for the refinement process. Upon gaining a solid understanding of the problem and its intricacies, a theoretical framework for performing probabilistic preferential reasoning will be devised. Following consultation and with the supervisor to confirm feasibility, initial specifications of the refined ranking and entailment algorithms will be developed and submitted for supervisor review prior to advancing to the implementation and optimisation stage.

5.3 Implementation and Optimization

Alec Lang The implementation of the knowledge base generator will be developed in Java, allowing for the generation of fine tuned data sets. Libraries from the TweetyProject [6] will be made use of, in particular its propositional logic library and the Sat4j SAT solver. Once a working prototype is developed, it will be further refined to enhance its efficiency when generating large amounts of data.

David Pullinger To validate the behavior and performance of the foundational algorithm, a preliminary Java implementation will be created. This implementation will require integration with a suitable logic library, such as the TweetyProject [6], and will involve constructing a wrapper around the propositional logic package to ensure better compatibility with defeasible logic.

Once the wrapper has been constructed, a front end for user input (Java Swing) will be built, allowing early iterations of the algorithm to be developed. Later, an optimized algorithm with improved time and space complexity will be developed, and unit tests will be conducted to verify its correctness. Finally, benchmarking testbeds such as the Java Microbench Harness (JMH) tool will be used to analyze the optimized algorithm.

Luke Slater Upon completing the development of the core algorithms, functional prototypes will be created to verify their performance and effectiveness. To compute entailment, it is anticipated that a logic library, like [6], will be employed. After the successful validation of these algorithms, a performance analysis will be carried out to pinpoint and incorporate any potential optimizations. This process will entail the use of a knowledge base generator in conjunction with a benchmarking testbed, such as JMH.

5.4 Evaluation and Comparison

Alec Lang The correctness of the generator will be tested by checking if the knowledge bases conform to the user defined specifications in which it was generated according to. Various configurations of knowledge bases will be generated and the time complexity of the algorithms shall be evaluated.

David Pullinger The defeasible entailment relations produced by the user-defined rankings will be compared to rational closure and lexicographic closure based on the similarity of the ranked models underlying them. Furthermore, the algorithm for computing the entailment relations will be compared to the algorithm by Casini et al. [3] in terms of time and space complexity (the time and space complexity of both algorithms will be calculated using a thorough analysis of the algorithms' data structures, operations, loops, and recursive calls). This comparison could help to determine if the use of custom, user-defined rankings in creating defeasible entailment relations results in any trade-offs in terms of computational efficiency.

Luke Slater The refined version of rational closure will be analyzed and compared to traditional rational closure, focusing on defeasible reasoning as well as space and time complexity. To identify any deviations or complementary aspects in relation to the prototypical nature of traditional rational closure, the new algorithms will be examined. This process will likely involve manually designing specific knowledge bases to illustrate how Bayesian inference can affect rational closure. Another possibility is generating knowledge bases and testing if the new algorithm adheres to each of the KLM postulates (excluding rational monotonicity).

6 RELATED WORK

The KLM framework was first introduced in [8]. Rational closure was then introduced by Lehmann and Magidor in [11], after which lexicographic closure was introduced by Lehmann in [10]. The primary aim of this project is to investigate and extend the work by Casini et al. in [3]. Bayesian inference can trace its origins to the foundational work of Reverend Thomas Bayes in the 18th century [1], and was later formalized by Pierre-Simon Laplace in the early 19th century [9]. A more modern formalization of Bayesian inference can be found in "Bayesian Data Analysis" by Gelman et al. [4].

7 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Throughout the project, the team will ensure proper attribution of all referenced work and maintain transparency in the research process.

8 ANTICIPATED OUTCOMES

8.1 Expected Impact

This project's multifaceted impact aims to advance understanding of using rational closure as the nonmonotonic core of defeasible reasoning, and comprehensive knowledge base generation for testing, contributes to practicality, efficiency, and accuracy improvements in defeasible entailment.

8.1.1 Rational Closure Extensions. Developing two novel extentions to rational closure, through integration of Bayesian inference and user-defined rankings, could enhance their applicability, thus refining current decision-making and reasoning methods.

8.1.2 Defeasible Knowledge Base Generation. Creating a customizable knowledge base generator offers researchers a valuable tool for generating defeasible knowledge bases, expediting the evaluation and comparison of various algorithms and entailment relations.

8.2 Key Success Factors

8.2.1 Rational Closure Extensions. The extensions to rational closure will be considered a success if:

- Algorithms for their computation are designed and implemented
- These algorithms are proven/shown to be partially correct and show to result in a rational defeasible entailment relation
- These algorithms are optimized to be equal in time and space complexity to other algorithms for computing defeasible entailment relations

8.2.2 Knowledge Base Generator. The knowledge base generator will be considered a success if:

- An algorithm for generating knowledge bases, given various parameters such as number of defeasible implications, is designed and implemented.
- The algorithm is demonstrated to conform to its specified input parameters.

8.3 System

The three components of the project will result in Java-based implementations of the designed algorithms. These will likely make use of the TweetyProject Java library, and thus a robust and thorough integration/interface must be developed. While the Bayesian inference and knowledge base generator components will likely be implemented as command line tools, the user ranking component may need to make use of a GUI library, such as Java Swing, to make ranking defeasible implications more intuitive.

Two major design challenges have been identified. Firstly, integration with the TweetyProject library will require a thorough investigation of the library and its API. This is crucial to all three components of the project. Secondly, the lack of an established knowledge base generator will cause issues in testing for the refinements of rational closure.

9 PROJECT PLAN

9.1 Risks and Contingencies

Table 2 in Appendix A lists several risks and their corresponding probabilities of occurrence and impact on the project, while Table 3 in the same appendix details the mitigation, monitoring, and management strategies for each risk. Based on this risk assessment, it can be concluded that the project has a low overall risk of failure.

9.2 Resources Required

Since the various components of the project will be developed in Java, the following resources will be required:

- **Development Environment:** We will need to have the Java Development Kit (JDK) installed to write and compile Java code
- **Libraries:** We will need to use various third-party libraries such as the TweetyProject [6]
- **Testing tools:** To ensure that the applications are reliable and perform as expected, we will need to use testing tools such as JUnit
- **Benchmarking tools:** To accurately and reliably benchmark the various algorithms, we will need to use benchmarking tools such as the JMH
- **Text editors:** To write any code, we will need access to text editors such as VSCode and Neovim
- **Hardware:** Access to computers with sufficient processing power and memory to run the aforementioned software is necessary

9.3 Milestones and Tasks

Table 4 in Appendix B lists the various milestones and tasks of the project, split amongst the three project members.

9.4 Timeline

A Gantt chart illustrating the projects milestones and tasks has been included in Appendix C.

9.5 Work Allocation

All three components will be completed individually from the Preliminary Work phase to the Implementation and Optimisation phase. In the Algorithm Analysis phase, however, the components focusing on refining rational closure will be able to make use of the knowledge base generator, and vice versa (the knowledge base generator will need a tool which can compute rational closure, to verify its correctness). Thereafter, the Final Paper will be completed individually while the Final Demonstration, Poster and Web page will be completed collaboratively.

9.6 Deliverables

Deliverable	Due Date
Literature Reviews	20 March
Proposal Draft	21 April
Proposal Presentation	25 April
Proposal Final	2 May
Progress Demonstration	17 July
Final Paper Draft	28 August
Final Project Papers	11 September
Final Project Code	15 September
Final Project Presentation	26 September
Project Poster	9 October
Project Webpage	16 October
School of IT Showcase	24 October

Table 1: Deliverables

REFERENCES

[1] Thomas Bayes. 1763. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London* 53 (1763), 370–418.

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

[3] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2019. Taking defeasible entailment beyond rational closure. In *Logics in Artificial Intelligence: 16th European Conference, JELIA 2019, Rende, Italy, May 7–11, 2019, Proceedings 16.* Springer, 182–197.

[4] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. 2013. *Bayesian Data Analysis* (3 ed.). CRC Press.

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

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

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

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

[9] Pierre-Simon Laplace. 1812. *Théorie Analytique des Probabilités.* Courcier.

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

[11] 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

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

[13] Yoav Shoham. 1987. Nonmonotonic logics: Meaning and utility. In *IJCAI*, Vol. 10. Citeseer, 388–393.

[14] Yoav Shoham. 1987. A Semantical Approach to Nonmonotonic Logics. In *Logic Programming and Nonmonotonic Reasoning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 227–250.

Appendix A RISKS AND CONTINGENCIES

ID	Risk	Probability	Impact
1	Inadequate understanding of the theoretical components behind each project	Medium	High
2	Lack of time due to poor time management.	Medium	High
3	Limitations in the technologies that have been chosen	Low	Medium
4	Scope creep during the design and implementation of algorithms	High	Medium
5	Integration challenges with third-party libraries	Medium	Medium
6	Failure of physical equipment during development	Low	High
7	Failure of software during development	Low	High
8	Unavailability of a project member	Low	Low
9	Inability to generate knowledge bases that vary in structure	Low	High
10	Inability to generate and mix complex statements with more basic statements	Medium	Medium
11	Inability to prove partial correctness of user-defined ranking algorithms	Medium	Low
12	Difficulty in meaningfully ranking defeasible implications	High	Medium
13	The integration of Bayesian inference into rational closure may lead to increased computational complexity and scalability issues	Medium	High
14	Introducing probabilistic preferential reasoning within minimal ranked entailment may result in incompleteness and inconsistencies in reasoning	Medium	Medium

Table 2: Risk Identification

ID	Mitigation	Monitoring	Management
1	Spend a suitable amount of time studying and compiling background information	Monitor feedback from the supervisor and reviewers to identify any potential issues with the understanding of theoretical components	Meet with the supervisor to address any gaps in the group's theoretical knowledge of the project
2	Punctually follow the agreed timeline for the project	Frequently check in on each group members progress	Adjust the deadlines of tasks
3	Conduct thorough research on the available technologies and assess their limitations before making a final decision	Conduct regular testing to identify any limitations or issues that may arise	Develop a plan for upgrading or replacing the technologies if necessary
4	Develop a comprehensive project plan that includes a detailed scope of work and a clear set of requirements for the algorithms	Regularly review the project progress and assess any changes to the scope or requirements of the algorithms	Meet with the supervisor to decide which parts of the project must be discarded to better align with the new scope
5	Consider using open-source libraries that have a large user base and a good track record of stability and reliability	Monitor updates and changes to the libraries, and assess their potential impact on the project	Upgrade or replace the libraries as needed
6	Ensure that project members have access to a backup workstation if their main one becomes unusable	Check that equipment is in working condition	Transfer development environment to backup workstation and ensure developed software still executes as expected
7	Frequently backup software and keep track of changes using Git	Use monitoring tools to identify any potential issues or limitations with the software	Roll back to a previous version of the software if the current version becomes unworkable
8	Keep in contact with project members through group chat	Communicate with team members to confirm their availability for meetings	Carry on with your part of the project
9	Study probability distributions and how to generate statements to adhere to a certain distribution	Check whether generated knowledge bases follow the correct distribution by graphing the data	Develop alternate strategies to distribute statements
10	Ensure that there is a thorough understanding of how complex statements are formed before implementing them in the final generator	Check that knowledge base contains statements that make sense	Develop a separate algorithm for generating knowledge bases with only complex statements and ensure that it works before implementing it
11	Implement a testing and validation framework to evaluate the accuracy of the user-defined ranking algorithms while they are being implemented	Monitor the performance of the ranking algorithms regularly to ensure they are producing accurate and relevant results	Provide a non-mathematical demonstration of the algorithm's correctness
12	Make ranking defeasible implications more intuitive (letting user input be ranking two statements rather than the entire knowledge base at once)	Evaluate if the resulting rational defeasible entailment relation is vastly different from the original input of ranked statements	Investigate more intuitive or understandable ways of ranking defeasible implications, and implement them
13	Investigate efficient algorithms, parallelization techniques, or approximations to reduce the computational burden of Bayesian inference	Continuously assess the computational requirements and performance of the proposed approach, identifying potential bottlenecks and areas for improvement	Adjust the scope or design of the project if computational limitations become prohibitive, considering alternative or complementary methods that can offer similar benefits with reduced complexity
14	Acknowledge that the primary aim of the project is to investigate the reasoning capabilities of the combined approach, and therefore, comprehensive mitigation may not be necessary	Regularly evaluate the coherence and reliability of the inferred conclusions, identifying cases where inconsistencies or incompleteness may impact the overall quality of the results	If inconsistencies or incompleteness become problematic, consider refining the proposed approach or incorporating additional techniques to address these issues

Table 3: Risk Mitigation, Monitoring, and Management

Appendix B TIMELINE

Tasks	Start Date	End Date
Literature Review	23/02	20/3
Project Proposal	12/4	2/5
- Draft	12/4	21/4
- Presentation	21/4	25/4
- Final	25/4	2/5
Theoretical Foundation and Algorithm Design	10/6	26/6
Luke Slater		
- Preliminary research	3/5	17/5
- Approach selection and approval	18/5	22/5
- Algorithm design	23/5	25/7
- Algorithm review	26/7	2/8
David Pullinger		
- Background research	3/5	10/5
- Algorithm design	10/5	31/5
- Rationality analysis	31/5	7/6
- Supervisor review of algorithms	7/6	14/6
Alec Lang		
- Preparatory research	3/5	14/5
- Algorithm design	14/5	30/6
- Approval of algorithms	30/6	7/7
Implementation and Optimisation	27/6	27/7
Luke Slater		
- Implementation in java	3/8	10/8
- Testing and optimization	10/8	15/8
David Pullinger		
- Defeasible wrapper construction	14/6	28/6
- Implementation of algorithms	28/6	14/7
Alec Lang		
- Algorithm implementation	7/7	28/7
- Generator development	28/7	14/8
Evaluation and Comparison	27/7	10/8
Luke Slater		
- Evaluation of defeasible reasoning	10/8	15/8
- Comparison to traditional rational closure and Bayesian inference	12/8	18/8
David Pullinger		
- Comparison to Rational Closure and Lexicographic Closure	31/7	5/8
- Analysis of presumptive/prototypical nature	5/8	10/8
- Evaluation of time and space complexity	10/8	15/8
- Proof of correctness	15/8	31/8
Alec Lang		
- Structure and feature analysis	14/8	20/8
- Time complexity assessment	20/6	25/8
Final Paper	26/7	2/9
- Scaffold	26/7	29/7
- Draft	29/7	28/8
- Final	28/8	11/9
Final Demonstration	11/9	26/9
Poster	26/9	9/10
Webpage	9/10	16/10
Showcase	16/10	24/10

Table 4: Milestones and Tasks

Appendix C GANTT CHART

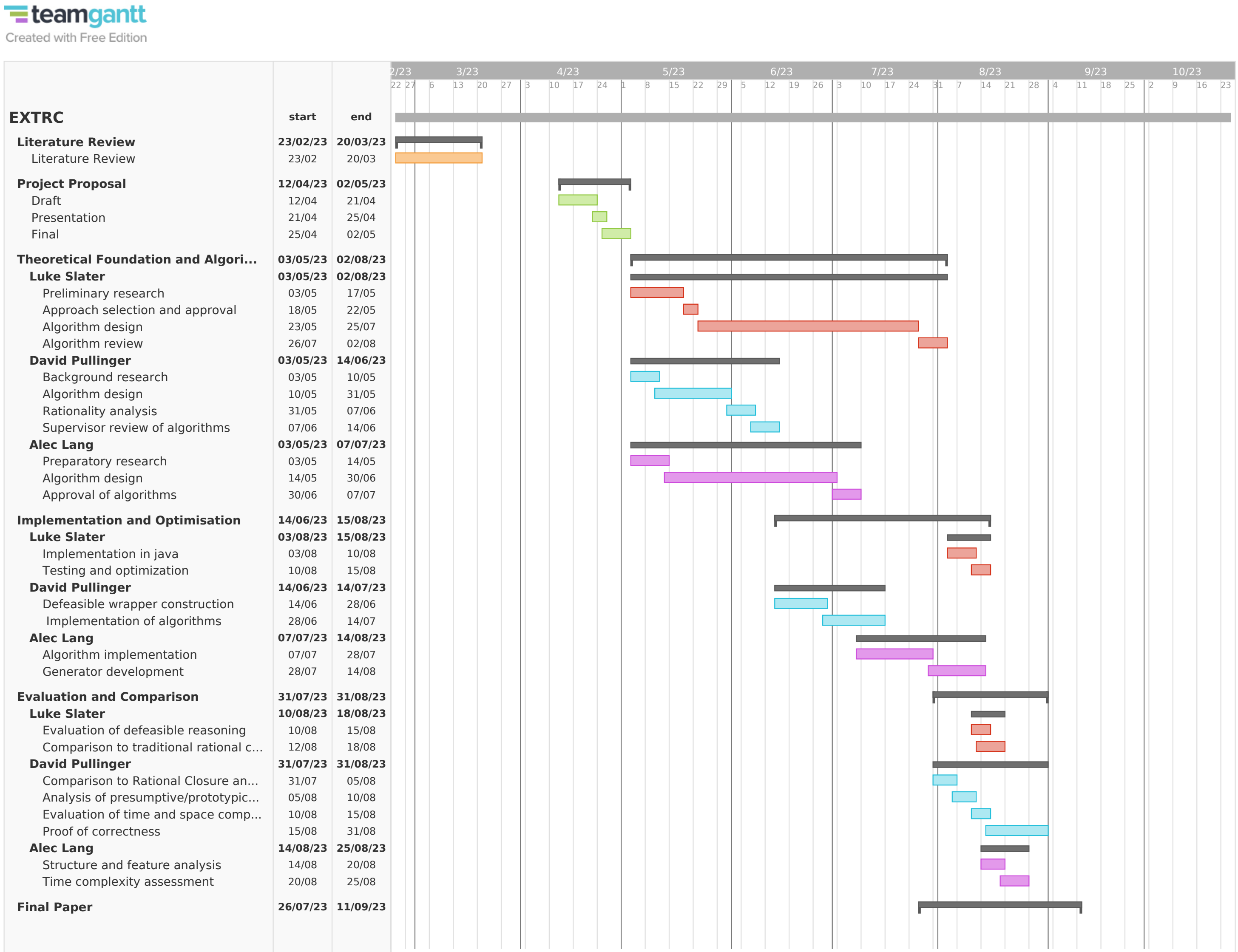

Figure 4: Gantt Chart (1)

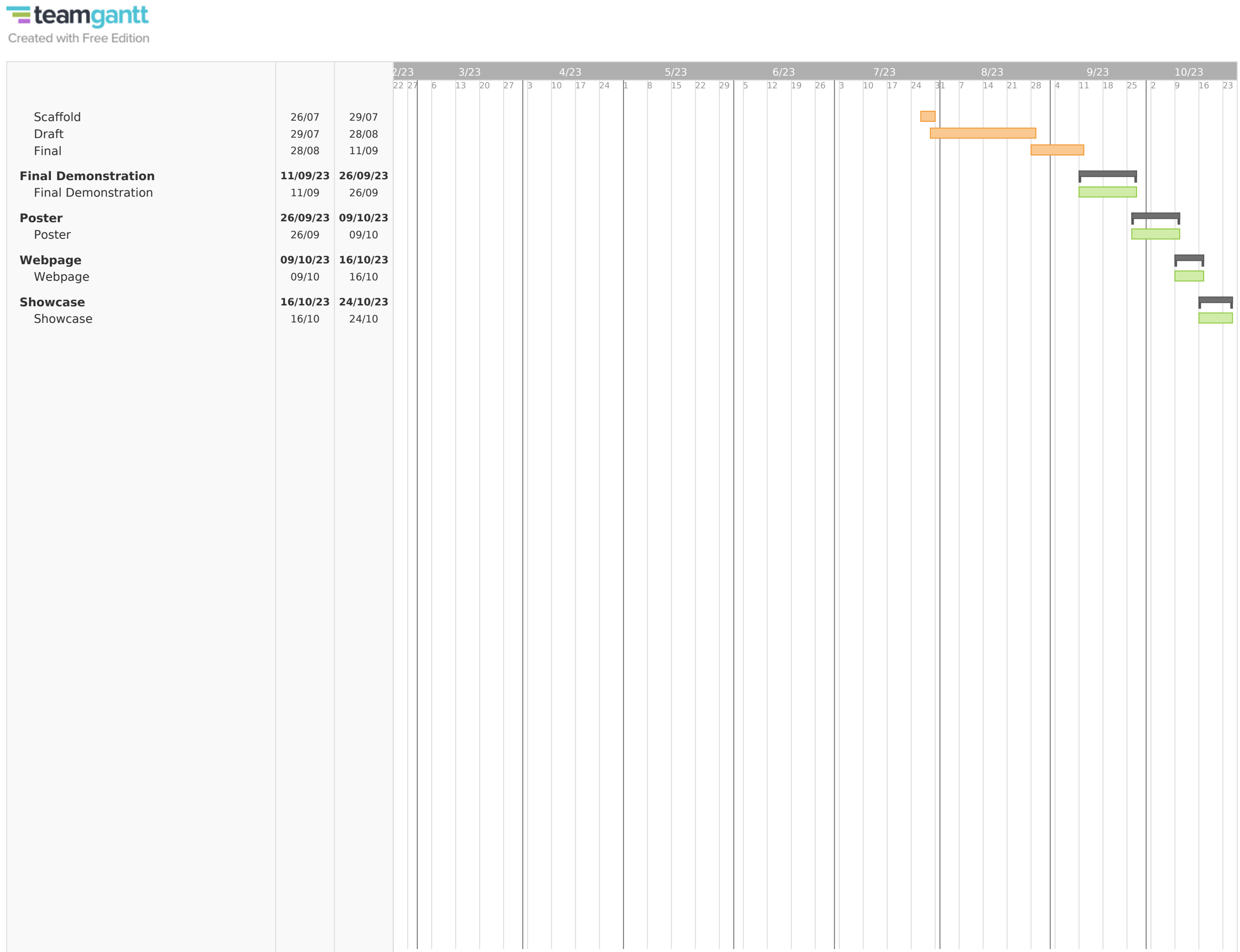

Figure 5: Gantt Chart (2)