Propositional Defeasible Explanation

Lloyd Everett
Department of Computer Science
University of Cape Town
lloydeverett@gmail.com

Emily Morris
Department of Computer Science
University of Cape Town
mrremi007@myuct.ac.za

ABSTRACT

Explanations provide a way of showing why an entailment holds in classical logics and are a crucial aspect of reasoning systems. However, they have not yet been explored in detail for forms of defeasible reasoning such as KLM. In our project, we aim to propose algorithms for justifications for KLM-style formalisms and also to characterise justifications for defeasible knowledge bases generally in a declarative manner. Such contributions would enhance our understanding of explanation for KLM and may also serve as the basis for practical implementations of explanation services.

CCS CONCEPTS

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

KEYWORDS

Artificial Intelligence, Knowledge Representation and Reasoning, Propositional Logic, Justifications, Explanations, Defeasible Reasoning, KLM Framework

1 INTRODUCTION

Knowledge Representation and Reasoning is a field within Artificial Intelligence in which information is modeled using formal logic, allowing one to apply a set of rules and manipulations related to a form of reasoning [3]. There are many different forms of logic, with different levels of expressiveness. Our work will focus on propositional logic, one of the most basic forms of classical logic.

An important aspect of reasoning is being able to infer new knowledge from existing information. In propositional logic, this is achieved using classical entailment. In addition to being able to infer new knowledge from existing knowledge, it is often useful to know why certain information is being inferred. Explanations provide reasons as to why an entailment holds. The form of explanation that we will focus on is justifications, which are conceptually simple and well-understood for classical logics [9].

There are certain forms of reasoning that classical logics cannot model. In particular, it is difficult to represent information that *typically* holds, but for which there might be exceptions, which is much closer to how humans reason. To do this, one must use a different form of reasoning known as *defeasible reasoning*. We will focus on defeasible reasoning looking in particular at the approach proposed by Kraus, Lehmann and Magidor (KLM) [11]. Unlike the classical case, little work has been done on defeasible explanation. In this project, we intend to explore and define a concept of defeasible explanation with a focus on KLM formalisms.

2 BACKGROUND

2.1 Propositional Logic

Classical propositional logic has a simple semantic and is the foundation for more complex logics. The following is a short description of classical propositional logic [1]. We begin with a finite set $\mathcal{P} = \{p, q, \cdots\}$ of *propositional atoms* which represent basic statements that can be assigned values of true or false. The binary connectives $\land, \lor, \rightarrow, \leftrightarrow$ and the unary negation operator $\neg$ are used recursively to form propositional formulas such as $\neg(p \lor q) \rightarrow p$. The set of all such formulas over $\mathcal{P}$ is called the *propositional language* $\mathcal{L}$.

An *interpretation* or *valuation* is a function $\mathcal{P} \rightarrow \{T, F\}$ that assigns a truth value to each atom in $\mathcal{P}$. We say a formula $A \in \mathcal{L}$ is *satisfied* by an interpretation $\mathcal{I}$, written as $\mathcal{I} \vDash A$, if A evaluates to true according to the truth values of the atoms in A and the semantics of operators in A which should be familiar from Boolean algebra. For example, if $\mathcal{I}(p) = T$ and $\mathcal{I}(q) = F$, then $\mathcal{I} \vDash p \lor q$ but $\mathcal{I} \nvDash p \land q$. The interpretations that satisfy a formula A are referred to as *models* of A, and the set of models of A is denoted $\text{Mod}(A)$.

A finite set of propositional formulas is called a *knowledge base* $\mathcal{K}$. An interpretation is a model of a knowledge base $\mathcal{K}$ if it is a model of all the formulas in $\mathcal{K}$. We say that $\mathcal{K}$ entails a statement A, denoted $\mathcal{K} \vDash A$, if $\text{Mod}(\mathcal{K}) \subseteq \text{Mod}(A)$.

This gives us the basis for a basic reasoning system. If knowledge is expressed as a set of knowledge base statements, we are able to test whether that knowledge entails other propositional statements. As an illustration of how this would work, consider the following example.

Example 2.1. Suppose one has a knowledge base containing the following statements:

(1) Tweety is a bird ($t \rightarrow b$)
(2) Birds fly ($b \rightarrow f$)

Following the logic of the natural language sentences, one can see how from this knowledge base one could conclude the that 'Tweety flies' since Tweety is a bird and we know birds fly. Formally, we have $\mathcal{K} \vDash t \rightarrow f$ since $\text{Mod}(\mathcal{K}) \subseteq \text{Mod}(f)$.

2.2 Defeasible Reasoning

2.2.1 Overview. A major limitation of classical logics is an inability to describe *typicality*. This makes it very difficult to then represent additional exceptional knowledge in a succinct way. We will demonstrate this by means of an example.

Example 2.2. Suppose one has a knowledge base $\mathcal{K}$ that contains the following information:

(1) Birds fly ($b \rightarrow f$)
(2) Penguins are birds ($p \rightarrow b$)

Using similar reasoning to the previous example one can conclude that $\mathcal{K}$ entails 'penguins fly'. Now suppose one adds a new statement 'penguins do not fly'($p \rightarrow \neg f$) i.e. that penguins are an exceptional type of bird that do not fly. One can now infer that penguins both fly and do not fly which removes one's ability to reason about penguins as they can no longer exist. More formally the atomic proposition of penguin must be false in every interpretation.

For this example to work, one should instead phrase the first fact as 'birds typically fly'. This captures an idea of uncertainty that allows one to later retract inferred statements if one learns information that contradicts them. This form of reasoning is referred to as *defeasible reasoning*.

Although there are many approaches to defeasible reasoning, one approach that has been studied extensively in the literature is that proposed by Kraus, Lehmann and Magidor (KLM) [11]. KLM proposes restrictions on the definitions for defeasible entailment by giving a set of properties, known as the KLM properties, they should adhere to. Thus, KLM does not define a single notion of defeasible entailment but rather defines a class of defeasible entailment relations that have some interesting theoretical and computational properties [10, 13]. Unlike the classical case, it is generally understood that it is desirable to have a number of formalisms for defeasible entailment that correspond to different reasoning styles [10]. Multiple rational [6] formalisms for defeasible entailment have been described in the literature including *Rational Closure* [13] and *Lexicographic Closure* [12]. Rational Closure corresponds to a more conservative form of reasoning compared to Lexicographic Closure which is much more permissive. Another formalism proposed by Casini et al. [5] that is not quite rational but is still closely related to KLM is *Relevant Closure*, which lies between Rational and Lexicographic Closure in terms of permissiveness. Before describing these formalisms, we first consider KLM as a reasoning framework.

2.2.2 Reasoning Framework for KLM. We extend classical propositional logic with a defeasible connective $\mid\sim$ which can be seen as the defeasible analogue of $\rightarrow$. Statements of the form $p \mid\sim q$ are read as *p typically implies q*. So for our previous example, we would express 'birds typically fly'as $b \mid\sim f$. We then define a notion of defeasible entailment, denoted $\approx$, which can be seen as the defeasible analogue of $\models$. Consider the following example of the desired behaviour of defeasible entailment.

Example 2.3. Suppose one has the following defeasible knowledge base:

(1) Birds typically fly ($b \mid\sim f$)
(2) Penguins are birds ($p \rightarrow b$)
(3) Penguins do not fly ($p \rightarrow \neg f$)
(4) Tweety is a bird ($t \rightarrow b$)
(5) Rico is a penguin ($r \rightarrow p$)

One would expect that any reasonable form of defeasible entailment would allow one to conclude that 'Tweety flies'and 'Rico does not fly '

The idea in the above example is that we want defeasible entailment to favour the most specific rules in the knowledge base that are applicable [10]. In the case of Rico, $r \rightarrow p, p \rightarrow \neg f$ is more specific than $r \rightarrow p, p \rightarrow b, b \mid\sim f$, so the former rule is favoured.

2.3 Rational Closure

Rational Closure is the form of defeasible reasoning that Lehmann and Magidor [13] proposed that satisfies the KLM properties. Rational Closure can be defined semantically, using what is known as ranked interpretations, as well as algorithmically. We will present the algorithmic definition. First we need to introduce some preliminary ideas. We require that all statements are defeasible implications of the form $\alpha \mid\sim \beta$ where $\alpha, \beta \in \mathcal{L}$. Note all classical formulas α have an equivalent defeasible form of $\neg\alpha \mid\sim \bot$.

We define the *materialisation* $\overline{\mathcal{K}}$ of a knowledge base $\mathcal{K}$ as the set of classical implications $\{\alpha \rightarrow \beta \mid \alpha \mid\sim \beta \in \mathcal{K}\}$. We say that a classical formula $\alpha \in \mathcal{L}$ is *exceptional* for $\mathcal{K}$ if $\overline{\mathcal{K}} \models \neg\alpha$ The intuition here is that exceptional formulas are false in the most typical valuations for $\mathcal{K}$ but may be true for more specific sets of valuations. The following example illustrates this idea.

Example 2.4. Consider $\mathcal{K} = \{b \mid\sim f, p \rightarrow b, p \mid\sim \neg f\}$. This should really be expressed as $\mathcal{K} = \{b \mid\sim f, \neg(p \rightarrow b) \mid\sim \bot, p \mid\sim \neg f\}$. Then $\overline{\mathcal{K}} = \{b \rightarrow f, \neg(p \rightarrow b) \rightarrow \bot, p \rightarrow \neg f\}$. Since $\overline{\mathcal{K}} \models \neg p$ we know that p is exceptional for $\mathcal{K}$ but on the other hand $\overline{\mathcal{K}} \not\models \neg b$ so b is not exceptional for $\mathcal{K}$.

We also define $\varepsilon(\mathcal{K})$ to give us the set of statements in $\mathcal{K}$ whose antecedents are exceptional for $\mathcal{K}$. This concept allows us to obtain for any knowledge base $\mathcal{K}$ a sequence of knowledge bases $\mathcal{E}_0^{\mathcal{K}}, \mathcal{E}_1^{\mathcal{K}}, \cdots, \mathcal{E}_n^{\mathcal{K}}$ such that knowledge bases earlier in the sequence contain, in addition to the statements in later knowledge bases, statements that are more defeasible or retractable than those in later knowledge bases. We simply let $\mathcal{E}_0^{\mathcal{K}} = \mathcal{K}$ and $\mathcal{E}_{i+1}^{\mathcal{K}} = \varepsilon(\mathcal{E}_i^{\mathcal{K}})$. The last knowledge base $\mathcal{E}_n^{\mathcal{K}}$ is the first $\mathcal{E}_i^{\mathcal{K}}$ where $\varepsilon(\mathcal{E}_i^{\mathcal{K}}) = \mathcal{E}_i^{\mathcal{K}}$ and is usually denoted with the infinity sign instead of n, i.e., $\mathcal{E}_\infty^{\mathcal{K}}$, as it is unique in that it contains statements that cannot be retracted (provided it is not empty).

We use this sequence of $\mathcal{E}$ knowledge bases to create a ranking $\mathcal{K}_0, \cdots, \mathcal{K}_\infty$ of the statements in $\mathcal{K}$ which we obtain by setting $\mathcal{K}_i = \mathcal{E}_i^{\mathcal{K}} \setminus \mathcal{E}_{i+1}^{\mathcal{K}}$ for $0 \leq i \leq n-1$ and $\mathcal{K}_\infty = \mathcal{E}_\infty^{\mathcal{K}}$. This ranking is such that the ranks are disjoint and earlier ranks contain statements that are more defeasible than later ranks.

Example 2.5. Suppose we have the following knowledge base

$$\mathcal{K} = \{b \mid\sim f, b \mid\sim w, p \rightarrow b, p \mid\sim \neg f, r \rightarrow p\}.$$

The associated ranking of formulas is given in Figure 1.

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

Figure 1: Ranking $\mathcal{K}_0, \cdots, \mathcal{K}_\infty$ for Example 2.5

This result aligns with our expectations: $p \mid\sim \neg f$ ('Penguins typically do not fly') is identified as less defeasible, or more typical, than $b \mid\sim f$ ('Birds typically fly') and $b \mid\sim w$ ('Birds typically have wings'). Note also that classical information always appears in the infinite rank $\mathcal{K}_\infty$.

Now that we have this ranking of statements, to compute $\mathcal{K} \approx \alpha \mid\sim \beta$ we start by checking whether α is exceptional with respect to $\mathcal{K}$ i.e. whether $\overline{\mathcal{K}} \vDash \neg\alpha$. If α is not exceptional, we compute $\overline{\mathcal{K}} \vDash \alpha \rightarrow \beta$ as our defeasible entailment result. Otherwise we remove the lowest level from our ranking and repeat the process.

Example 2.6. Consider the query $r \mid\!\sim \neg f$ and the same knowledge base as in Example 2.5. The antecedent r is exceptional for $\mathcal{K}$, so we remove the statements in the first rank $\mathcal{K}_0$ from $\mathcal{K}$. Now r is not exceptional for $\mathcal{K}$, so we check whether $\overline{\mathcal{K}} \vDash p \rightarrow \neg f$. This classical entailment holds, so we conclude that the defeasible entailment holds for Rational Closure.

2.4 Relevant Closure

Rational Closure, though a viable form of reasoning, is sometimes overly conservative. The reason for this, intuitively, is that we always retract entire ranks of more typical statements even though only a handful of statements in a rank may disagree with statements in higher, less typical ranks. In doing so, we significantly restrict the entailments we can derive as soon as the antecedent of the query is even slightly atypical. Relevant Closure, initially described by Casini et al. [5] tries to address this problem by adapting Rational Closure so that we only retract the statements *relevant* to the exceptionality of the antecedent. Casini et al. in fact describe two forms of Relevant Closure, *Basic Relevant Closure* and *Minimal Relevant Closure*, where the former is more conservative than the latter. Neither form of Relevant Closure is rational, which can arguably be seen as a limitation; however, Relevant Closure represents a less conservative style of reasoning while remaining computationally tractable [5].

2.5 Lexicographic Closure

Lehmann [12] presents a form of defeasible entailment known as Lexicographic Closure which satisfies the KLM properties. Like Rational Closure, Lexicographic Closure has both a semantic and algorithmic definition. The algorithm for Lexicographic Closure can in fact be defined as an extension of the Rational Closure algorithm, as shown by Casini et al. [6]. This is done by ranking statements according to the Rational Closure algorithm, and then imposing an ordering on statements within each level. Intuitively, the idea in Lexicographic Closure is that instead of removing an entire level as in Rational Closure, we instead aim to retain as much information as possible by only removing the statements within that level that cause the inconsistency.

3 PROJECT DESCRIPTION

As humans, when reasoning we are generally able to give evidence as to why we think a specific fact holds. For example, if one knows that 'birds fly'and 'penguins are birds', one could reasonably conclude that 'penguins fly'. To substantiate ones conclusion, one would give the two known facts as evidence. In reasoning systems, *explanations* tell us which statements in a knowledge base are relevant to the entailment between a knowledge base and an entailed statement [7]. *Justifications* are a simple form of explanation frequently used for classical logics based on the idea of minimal subsets of a knowledge base that entail a propositional formula. While much research has been done concerning classical justifications and their computation, little work has been done in extending this to defeasible reasoning. In particular, no definition of defeasible explanation has been proposed specifically for KLM-style entailment.

3.1 Project Work

We plan to extend the algorithms for Relevant and Lexicographic Closure to provide a way of computing justifications for these forms of defeasible entailment. We suspect that because these algorithms reduce to a series of classical entailment computations, we will be able to employ methods for computing classical justifications to obtain a version of defeasible justification as Chama did for Rational Closure [7]. We also aim to investigate and provide a general definition for defeasible explanation in the context of the KLM framework and possibly in the more general context of defeasible reasoning. Given this definition, we will then go and prove that our extended algorithms, along with a previously extended algorithm for Rational Closure [7], agree with our proposed definition.

3.2 Motivation

Explanation services are a crucial aspect of using reasoning systems practically. They are particularly helpful for large and complicated knowledge bases where it is not always obvious why an entailment holds. Explanations give one insight into how a particular knowledge base works and can aid in the debugging of knowledge bases [9]. They can also be presented in a way that enables people unfamiliar with the knowledge base or reasoning system to gain greater understanding [2, 14] and can be used to build versions of formal proofs.

As we have seen, classical logics can be restrictive in terms of their ability to express information that is not necessarily categorical. Explanation has not yet been explored in detail for defeasible reasoning apart from some foundational work [4, 8]. Our work would suggest algorithms for evaluating defeasible justifications and may establish a better understanding of explanation in the context of KLM and perhaps more generally for defeasible formalisms. These contributions would be valuable seeing as explanation services are a crucial aspect of reasoning systems and these ideas are not currently well-understood. Our results may also serve as the basis for a practical implementation of a KLM-style reasoning system that offers defeasible explanation as a reasoning service, e.g., perhaps oriented around knowledge base debugging.

4 PROBLEM STATEMENT

4.1 Aims

This project aims to:

- Extend algorithms for Lexicographic and Relevant Closure to compute justifications.
- Provide a definition for defeasible explanation.
- Prove that the extended algorithms (along with the extended Rational Closure algorithm) are compliant with the proposed definition.

4.2 Research Questions

The work in this project aims to present answers to the following research questions:

(1) Can the Lexicographic Closure algorithm be extended to allow the computation of defeasible justifications?
(2) Can the Relevant Closure algorithm be extended to allow the computation of defeasible justifications?
(3) Can one provide a reasonable definition for defeasible explanation?
(4) Are the extended algorithms compliant in terms of the proposed definition for defeasible explanation?

5 RELATED WORK

Horridge [9] provides and investigates a variety of algorithms for computing classical justifications. Horridge presents algorithms that can be classified as either glass-box or black-box algorithms. In glass-box algorithms, justification computation is built into the reasoning algorithm. This means justifications are computed during the reasoning process. Black-box algorithms are independent of the underlying reasoning process and are computed separately. The two types of algorithms have trade offs in terms of efficiency and ease of use which must be taken into consideration when it comes to choosing which one to use.

Chama [7] presents an adaption of the Rational Closure algorithm for the computation of justifications for Rational Closure defeasible entailment. Chama uses algorithms presented by Horridge as a basis for computing justifications. The approach here resembles the reasoning process for Rational Closure: after eliminating more typical ranks, we rely on classical tools to reason about the knowledge base, only in this case we use classical justification instead of classical entailment.

Brewka et al. [4] take a different approach for defining defeasible justifications. Brewka et al. present an abstract idea of a defeasible justification that is claimed to work for all forms of defeasible reasoning, called a strong explanation. One of the reasons why justifications work well in classical reasoning is that anything entailed by a justification is entailed by the knowledge base. However, in the defeasible case, the rest of the knowledge base might contain information contradictory to the entailment. Brewka et al. address this by extending the definition of a justification to include an extra property that ensures there is no extra information in the knowledge base that contradicts the justification.

6 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

There are no real ethical issues to be taken into account given that our project is theoretical and does not involve any human subjects. Likewise, we do not foresee any legal issues. The Protection of Personal Information Act is not applicable as we will not be using or storing any personal information. Licensing and copyright are similarly not a concern; we are not making any use of copyrighted works other than referencing the literature on the subject according to academic norms.

7 PROJECT PLAN

7.1 Methods

The work for this project will consist of iterations of working to understand a concept, producing a theory and then testing the validity of this theory. Considerable work has already been done in terms of understanding the relevant literature that forms a basis for this project, but this knowledge will need to be consolidated and expanded upon as the project progresses. This project has three main stages. These are tasks (5), (6) and (7) in Table 1.

7.1.1 Algorithm Extension. Before we can adapt the relevant algorithms, we must first gain an in-depth knowledge of how they function. Thus we will start by reading the papers that originally defined these algorithms and then read any other literature that presents a version of these algorithms, paying particular attention to any examples that may be given as these help build an intuitive understanding. We will also focus on how these algorithms relate to Rational Closure. Next we will look at how the Rational Closure algorithm was extended for computing justifications. Finally, we will combine all this information to provide extended algorithms for Relevant and Lexicographic Closure which compute justifications

To evaluate whether this section has been satisfactorily completed, we will present rough drafts of these extended algorithms to our supervisor. We will also provide an abstract motivation as to why they provide a meaningful way of computing defeasible explanations. If the algorithms are not deemed satisfactory, we will review any issues and work to produce new extended algorithms.

7.1.2 Definition for Defeasible Explanation. The process of producing a definition for defeasible explanation is more roughly defined. We will start by reviewing existing literature for explanations and defeasible explanations. A particular focus is on a definition that has been given for what is known as strong justifications [4]. As a starting point, we will explore this definition in terms of KLM, both from a semantic perspective and in terms of the rankings of statements we discussed earlier for Rational Closure. Using this knowledge as a basis, we will then define a general definition for defeasible explanation by adapting and strengthening the conditions we had for classical justifications.

We will evaluate this section more intuitively, verifying whether it corresponds to our expectations. We will provide a high level motivation as to why our approach is suitable. For this, it will be helpful to read other papers in the field of logics where definitions are proposed and substantiated to get a better sense of how this is formally done.

7.1.3 Relate Definition and Algorithms. Once we have presented a definition for defeasible explanation, we will go back and formally prove our extended algorithms, as well as proving the extended algorithm that has already been defined for Rational Closure. This will include proving the algorithms correct by showing they are sound and complete according to our definition. If the defined algorithms are not correct, we will review either the algorithms or our definition depending on where the issue arises. We will also provide a high level description of the relation between the algorithms and the definition. Proofs can be verified by our supervisor.

7.2 Anticipated Outcomes

We anticipate that we will produce justification algorithms for Relevant Closure and Lexicographic Closure similar to the algorithm given by Chama [7] for Rational Closure. We will also produce some form of a definition for defeasible explanation. Finally, we will produce soundness and completeness proofs that relate the reasoning algorithms for Rational, Relevant and Lexicographic Closure to this declarative description of defeasible explanation. We also intend to give more informal descriptions to help the reader understand these results intuitively. These results would give a theoretical framework that may serve as the basis for implementing the explanation services for defeasible reasoning as discussed in Section 3.2.

We will judge the success of this project based on whether these outcomes have been achieved or not. Producing a definition for defeasible explanation may prove to be a difficult task. For this section, we will consider it successful if we produce a thorough investigation of how defeasible explanation should behave, even if we are not able to provide a formal definition.

7.3 Risks

Along with the more generic risks associated with projects, this project is contingent on one's ability to understand relatively complex literature. This introduces risks that arise from not being able to understand the work, or not being able to understand or address the work in the required amount of time. In particular, spending too much time trying to grasp a specific concept can result in delays that can jeopardise the entire project. A table of risks, along with their corresponding probability and impact are provided in Appendix A. A risk mitigation, monitoring and management plan is also presented.

7.4 Resources

The only resource required for this project is the literature needed to understand KLM-style defeasible reasoning, the various defeasible entailment algorithms and classical explanations. Since there will be no physical implementation of the algorithms, no special software is required.

7.5 Deliverables

The main deliverables for this project will be the results presented and proven in the final papers. We will provide adapted versions of the Relevant Closure and Lexicographic Closure algorithms which allow for the computation of justifications. We will present and motivate a reasonable definition for defeasible explanation. We will use this definition to prove the soundness and completeness of the justification computation algorithms, starting with Rational Closure and then proceeding to Relevant and Lexicographic Closure. Other formal deliverables include the literature reviews, project proposal, project proposal presentation, the final paper scaffold, the final project presentation, the project poster and the website for the project.

7.6 Milestones and Timeline

Our overall approach to the project was discussed in Section 7.1. With that in mind, we present a set of milestones for our project in Table 1. It is difficult to predict precisely when these stages will take place, so the dates here should therefore be seen as helpful estimates that ensure that we are on track rather than exact deadlines. A Gantt chart is given in Appendix B which illustrates this timeline.

Table 1: Project Milestones and Targets

Task	Dates
(1) Topic introduction	22/01–17/03
(a) Personal introductions	22/01–27/01
(b) Introductory lectures	28/01–17/03
(2) Literature review	14/05–04/06
(3) Project proposal	07/06–24/06
(a) Report	07/06–21/06
(b) Project presentation	22/06–24/06
(4) In-depth topic exploration	06/07–14/07
(a) Focused review with supervisor	06/07–07/07
(b) Discussions on approach	08/07–09/07
(c) Consolidating understanding	12/07–14/07
(5) Extending the algorithms: *Relevant Closure (Lloyd) & Lexicographic Closure (Emily)*	15/07–27/07
(i) Review algorithm	15/07–19/07
(ii) Write up algorithm	19/07–20/07
(iii) Extend to include explanations	20/07–26/07
(iv) Evaluate results intuitively	23/07–27/07
(6) Explore a declarative description of defeasible justification	28/07–05/08
(a) Explore from an axiomatic perspective	28/07–02/08
(b) Explore from a semantic perspective	30/07–04/08
(c) Evaluate results intuitively	02/08–05/08
(7) Relate algorithms to declarative description	06/08–16/08
(a) Soundness and completeness proofs	06/08–12/08
(b) Intuitive discussion	12/08–16/08
(8) Final paper	16/08–17/09
(a) Scaffold	16/08–20/08
(b) Complete draft	23/08–06/09
(c) Final submission	06/09–17/09
(9) Other deliverables	20/09–18/10
(a) Demonstration	20/09–08/10
(b) Project poster	24/09–11/10
(c) Web page	05/10–18/10

7.7 Work Allocation

Task (5) has been allocated so that Emily Morris will be focusing on the algorithm for Lexicographic Closure whereas Lloyd Everett will be looking at the algorithm for Relevant Closure. This is indicated in Table 1. Many of the tasks associated with learning the topic in task (4) may be shared or collaborative to some extent but our key focuses will be different as we will be learning about our respective algorithms in preparation for task (5). Once task (5) is complete, we will work together on (6) and divide work between us as we go; this task would be difficult to split up in advance. Task (7) is again split up according to algorithms: Emily Morris will complete proofs for Lexicographic Closure and Lloyd Everett for Relevant Closure. Our initial focus here will be to start with proofs for Rational Closure which will serve as a starting point for proofs of our respective algorithms.

REFERENCES

[1] Mordechai Ben-Ari. 2012. *Propositional Logic: Formulas, Models, Tableaux.* Springer London, London, 7–47. https://doi.org/10.1007/978-1-4471-4129-7_2

[2] Stefan Borgwardt, Anke Hirsch, Alisa Kovtunova, and Frederik Wiehr. 2020. In the Eye of the Beholder: Which Proofs are Best?. In *Proc. of the 33th Int. Workshop on Description Logics (DL 2020).*

[3] Ronald J. Brachman and Hector J. Levesque. 2004. *Knowledge representation and reasoning.* Elsevier.

[4] Gerhard Brewka and Markus Ulbricht. 2019. *Strong Explanations for Nonmonotonic Reasoning.* Springer International Publishing, Cham, 135–146. https://doi.org/10.1007/978-3-030-22102-7_6

[5] Giovanni Casini, Thomas Meyer, Kodylan Moodley, and Riku Nortjé. 2014. Relevant Closure: A New Form of Defeasible Reasoning for Description Logics. In *Logics in Artificial Intelligence*, Eduardo Fermé and João Leite (Eds.). Springer International Publishing, Cham, 92–106.

[6] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2019. Taking Defeasible Entailment Beyond Rational Closure. In *Logics in Artificial Intelligence*, Francesco Calimeri, Nicola Leone, and Marco Manna (Eds.). Springer International Publishing, Cham, 182–197.

[7] Victoria Chama. 2020. *Explanation for defeasible entailment.* Master's thesis. University of Cape Town.

[8] V. Chama. 2020. *Explanation for defeasible entailment.* Master's thesis. University of Cape Town, Cape Town, South Africa. http://hdl.handle.net/11427/32206

[9] Matthew Horridge. 2011. *Justification based explanation in ontologies.* Ph.D. Dissertation. University of Manchester.

[10] Adam Kaliski. 2020. *An overview of KLM-style defeasible entailment.* Master's thesis. University of Cape Town.

[11] 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

[12] Daniel Lehmann. 1995. Another perspective on default reasoning. *Annals of mathematics and artificial intelligence* 15, 1 (1995), 61–82.

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

[14] Tu Anh Thi Nguyen. 2013. *Generating natural language explanations for entailments in ontologies.* Ph.D. Dissertation. The Open University.

APPENDIX A RISKS

ID	Risk	Probability (1-5)	Impact (1-5)
1	Failure to understand the necessary content	3	5
2	Cannot access required papers	1	4
3	Supervisor is not available	2	3
4	Poor time management and time-boxing of work	3	3
5	Cannot find a reasonable definition for defeasible explanation	3	4
6	Algorithms are not sound/complete according to the definition of defeasible explanation	2	4
7	Conflict within the group	1	2
8	Project partner leaves	1	2

ID	Mitigation	Monitoring	Management
1	Engage with content early and frequently to find and deal with areas of confusion early on	Discuss content with supervisor regularly to ensure understanding is correct	Meet with supervisor to address areas of confusion
2	Use university credentials to access papers from journals that require an account	N/A	Ask supervisor for relevant papers or email the authors
3	Set up regular meetings with supervisor to ensure there is a scheduled time for interaction	Keep in contact with supervisor so that one is aware in advance when they are unavailable	If possible, continue with other work until supervisor becomes available
4	Create a schedule and deadlines to ensure that work is being done at a steady pace and that parts of the project do not overwhelm the timeline	Meet regularly with partner and supervisor to ensure project is on track	Ensure base project work is completed before attempting more time consuming extensions
5	Read literature on how people come up with and motivate properties for logical concepts	Review possible definitions and approaches with supervisor	Ensure there is enough project work that is not dependent on the definition so that a thesis can still be completed
6	No possible mitigation, but since the algorithms seem to present a good way of computing justifications, it is unlikely that they will not work with any reasonable definition	We are struggling to complete this section beyond its allotted time	Adapt algorithms so that they are sound/complete according to the proposed definition
7	Ensure open and frequent communication between group members and address issues as they arise	Maintain a constant dialogue to ensure both members are happy with the project and its direction	Engage supervisor to help advise on disputes
8	N/A	N/A	Ensure that work is separate enough that a one's project can continue without one's partner

APPENDIX B TIMELINE

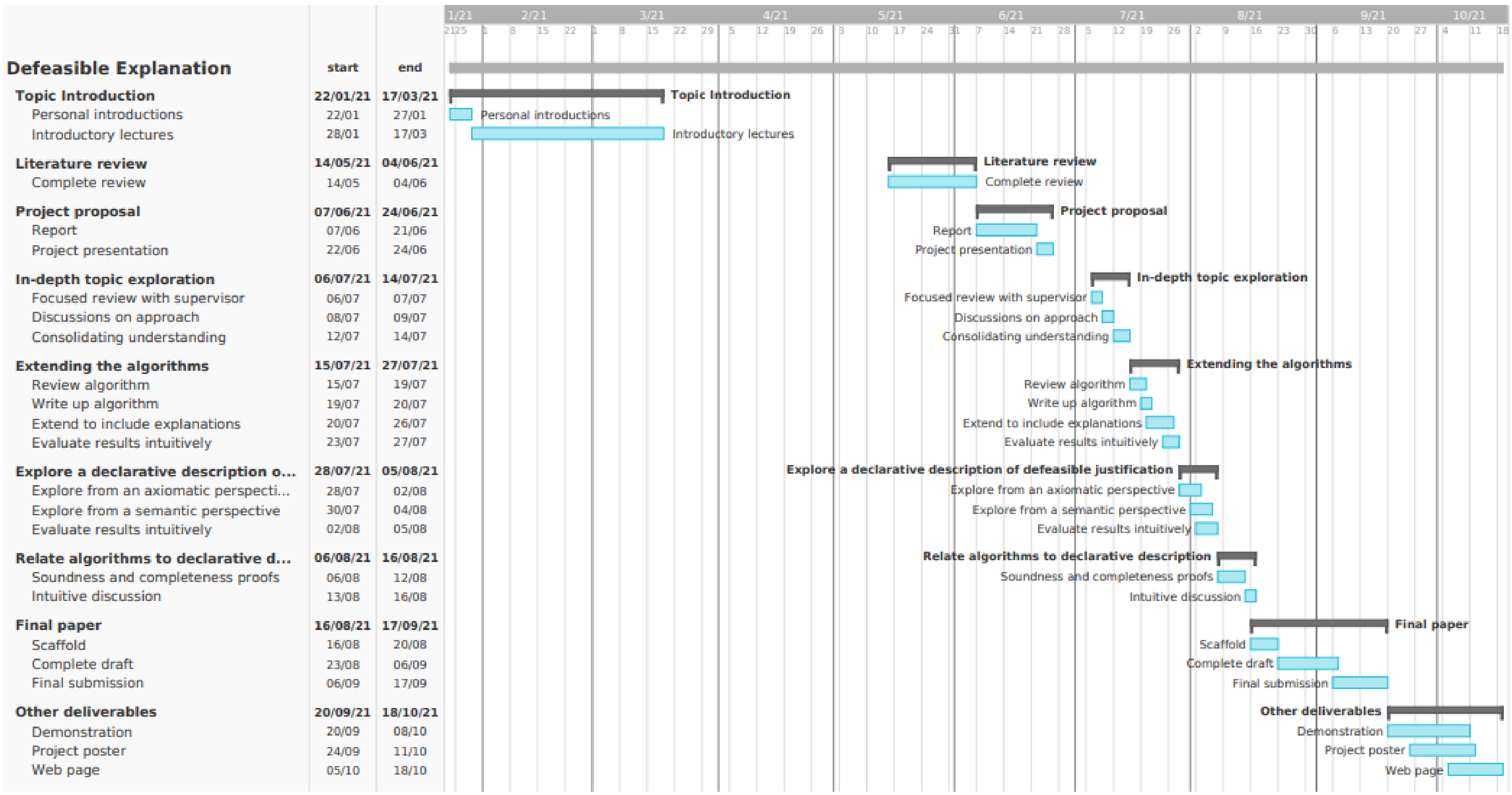

Figure 2: Project Timeline Gantt Chart