2023 MSc proposal: Generalised weak justifications for rational defeasible entailment and the connection to strong explanations

Jane Imrie
IMRJAN001@myuct.ac.za
Department of Computer Science
University of Cape Town
South Africa

1 INTRODUCTION

Within the broader field of Artificial Intelligence, there exists a sub-field called Knowledge Representation and Reasoning (KRR). KRR utilises formal logical systems, such as propositional logic, description logic, first-order logic and so forth, in order to both represent and reason about domain knowledge [14]. This domain knowledge can be explicitly represented in a structure known as a *knowledge base* (KB) [15] and these logical systems can be classified as either *monotonic*: i.e. once an inference is made, it cannot be rescinded, even if conflicting information were to be added to the knowledge base; or *nonmonotonic*, which does make allowance for retraction of statements. Given that human knowledge is often incomplete i.e. more information can be added as it is discovered, and imperfect, i.e. there can be exceptions to rules, it thus generally follows that logics with the nonmonotonic properties are preferable to use when modelling knowledge and drawing inferences. However, monotonic logics hold importance, as many nonmonotonic reasoning principles and characteristics draw upon those found in monotonic logics.

Equally as important as reasoning about knowledge, it is imperative that AI systems are able to "explain" their results. The ability to do so is one of the key requirements needed to foster trust between people and these systems. In the best case, the level of trust would be comparable to that between people and the results provided by a calculator [6]. Explainability, as it will be referred to from here-on throughout this document, also provides a means of debugging a knowledge base [11]. Domain-expert users of a knowledge base (that contains information within their area of expertise), can verify that the reasoner works correctly by verifying the results of the entailment.

Whilst much research has been done in the area of classical logic and classical explanations [6], there has been very little work in that of defeasible explanations [4, 6, 8, 16]. Given what has been said previously about the nature of human knowledge, it stands to reason that defeasible explanations should be further researched as part of the importance of AI explainability.

2 BACKGROUND AND RELATED WORK

2.1 Classical Logic

Tarski's notion of consequence provides the basis of the most prevailing form of logic, namely *classical reasoning*. Tarski defined a set of properties which need to be fulfilled in order to prove that an inference has the most support possible from the explicit knowledge. This defines the classical notion of consequence [7], which forms part of the groundwork for propositional logic. However, this paradigm of reasoning cannot handle explicit knowledge that contains exceptions. The archetypal example of this is birds - flight is typically one of a bird's most defining characteristics. However, many animals are classified as birds despite being flightless - penguins, chickens, pigeons and swans are all classified as birds, but only some of these can fly. We formalise this as a knowledge base as follows:

$$\begin{array}{|l|}\hline \texttt{bird} \rightarrow \texttt{flies} \\ \texttt{penguin} \rightarrow \texttt{bird} \\ \texttt{penguin} \rightarrow \neg\ \texttt{flies} \\ \hline \end{array}$$

Using propositional logic (PL), we have to conclude that birds do not exist. We could modify the knowledge base to account for penguins, but then we would also have to do so for chickens, ostriches and so forth. Thus, this approach becomes unfeasible for many cases where new knowledge is being added to existing.

2.1.1 Propositional Logic. Despite this, there is a necessity to understand propositional logic, as this often represents the simplest case for concepts such as satisfiability and entailment checking. Using Ben-Ari et al. [2], propositional logic starts with atoms - the simplest statements that can be made within this system. Atoms can be assigned a value of either true (T) or false (F). Then there are Boolean operators/connectives, such as *and*, *or*, *not*, which have formal definitions and symbols within the system, and are used together with atoms to create *formulas*. Lower-case Greek letters will be used to represent formulas, such as α, β etc, and lower-case Latin letters, p, q, r,... will be used for atoms. These aforementioned elements constitute the syntax of propositional logic.

In addition to the syntax, the semantics of the language (i.e. PL) must be defined - that is, how we assign meaning to formulas. This enables us to perform inference and systemic analysis. A simple example is, given two atoms p and q and a formula such as $p \rightarrow q$, we can assign some truth values $\{T, F\}$ to our atoms. We refer to this as an interpretation or world, denoted with $\mathcal{I}$. Interpretations allow for the evaluating of formulas by considering the value of the base atoms and connectives that they are composed of, and will return either true or false. . The model(s) of a formula, $Mod(\alpha)$, are those where the values of the atoms contained within it will allow the formula to evaluate to true. A formula is said to be *satisfiable* if and only if it has a model:

Definition 2.1 (Satisfiability). For some $p \in \mathcal{P}$, where $\mathcal{P}$ represents the set of all propositional formulas, if $\mathcal{I}(p) = T$, then $\mathcal{I} \Vdash p$

We can use the concept of satisfaction and models to define *logical consequence*. Given two formulas, α and β, we can say $\alpha \vDash \beta$, read as "α is a logical consequence of β", if $Mod(\alpha) \subseteq Mod(\beta)$. Equivalently:

Definition 2.2 (Logical Consequence). Define $\mathcal{U}$ to be the set of all interpretations for our language $\mathcal{L}$. We then state that $\alpha \models \beta$ if and only if for every $\mathcal{I} \in \mathcal{U}$, if $\mathcal{I} \Vdash \alpha$ then $\mathcal{I} \Vdash \beta$.

The above can be applied to derive classical entailment: Given $\mathcal{K}$ is a finite set of propositional formulas and a formula α, we can say that $\mathcal{K} \models \alpha$ - read as "$\mathcal{K}$ entails α" - if and only if for every $\mathcal{I} \in \mathcal{U}$ we have that when $\mathcal{I} \Vdash \mathcal{K}$ it is also the case that $\mathcal{I} \Vdash \alpha$. Plainly, entailment states what formula/formulae can be derived based on other formulae. We can extend satisfiability and entailment to knowledge bases, the minutiae of which can be found in [2] .

2.2 Classical Explanation

Users of a reasoning system may struggle to understand how certain entailments hold, particularly for knowledge bases of a domain that they may be unfamiliar with or are simply complex. Satisfactory explanation tools need to be equipped with features that allow them to both choose the best explanation/justification (out of potentially many) for a given entailment AND ensure that the chosen one is more understandable than the others [6]. For the classical case, addressing these two issues is relatively simple.

There are a number of approaches which can be used to compute explanations. Monotonic logics, such as PL, description logics and so forth, may use Baader et al.'s axiom-pinpointing algorithm [1], which involves finding minimal subsets of axioms which allow for a conclusion to be drawn. These sets are known as justifications [10]:

Definition 2.3 (Justification). Given a formula α, and a knowledge base $\mathcal{K}$, a justification for α is $\mathcal{K}'$, a minimal subset of $\mathcal{K}$ such that $\mathcal{K}' \models \alpha$.

Consider a modified version of the knowledge base given in section 2.1:

```
bird → flies
bird → wings
robin → bird
penguin → bird
penguin → ¬ flies
```

From now and proceeding forward, we use the first letter of each concept as shorthand to maintain brevity - for example "bird" is "b". Following from that, the following query can be posed: $\mathcal{K} \models \mathsf{r} \rightarrow \mathsf{w}$? Using the minimal subsets method, the justification for the query is: $\mathcal{J} = \{\mathsf{r} \rightarrow \mathsf{b}, \mathsf{b} \rightarrow \mathsf{w}\}$.

Justifications can be computed algorithmically, and there are two main approaches in the classical case: computing an individual or all justifications, or black-box and glass-box methods. Additionally, it is important to note that the expression of explanations is not necessarily restricted to that of justifications, nonethless, they are the most popular due to the fact that they align better with how people perform reasoning. Explanations in the classical case are well-studied and understood [6].

2.3 KLM and Defeasible Logic

Kraus, Lehmann and Magidor (KLM) [12] and Lehmann and Magidor [13] defined an extension to propositional logic in order to add the nonmonotonic property to it. This was achieved with the concept of *preferential reasoning*, the core tenet of nonmonotonic reasoning. KLM asserts that the notion of *typicality* should be possible to explicitly state within the language. There are some properties within KLM which relations need to adhere to in order to achieve this advantageous feature - details of these postulates can be found in [12].

KLM semantics is based upon preferential semantics - the idea that an agent may prefer one interpretation over another. More specifically, that they may choose a more typical interpretation over another. For defeasible reasoning, this translates to preferred worlds being those that are more typical, referred to as *preferential interpretations*. Unlikely worlds are not considered unless explicit information necessitates this.

Ranked interpretations are a subset of preferential interpretations, which allow for a more finer-grained ordering of the preferred worlds. One of the weaknesses of preferential interpretations is that, although it is possible to distinguish between more typical, less typical and impossible worlds, there is no way of differentiating between worlds that are at the same level of typicality. However, ranked interpretations act as a function over the entire set of possible interpretations and impose a total ordering [11], which addresses the aforementioned issue. Ranked interpretations define rational consequence relations - these relations inherit all the properties of preferential consequence relations, along with one other, rational monotonicity [13].

2.4 Nonmonotonic Entailment

Both preferential and ranked entailments are monotonic. However, defeasible knowledge bases require some form of nonmonotonic entailment. This form of entailment is created using the notion of ranked interpretations and minimal sets. Highly detailed explanations of the varying entailment relations is beyond the means of this paper, though references for further reading will be provided.

Firstly, a semantics for nonmonotonic entailment needs to be defined. Models for a knowledge base can be ranked and can have a partial order defined over them, such that the more preferred ranked interpretations, denoted $\mathcal{R}^{\mathcal{K}}$, will be "pushed" further down the rankings. This minimal element (essentially a ranked interpretation), represents a specific pattern of reasoning, and will align precisely with that of the preferential entailment relation. This is referred to as "minimal ranked entailment" (MRE) and it's nonmonotonicity makes it highly desirable as a basis for entailment relations in defeasible reasoning [11].

2.4.1 Rational Closure. Rational closure (RC) is an entailment relation and provides a syntactic definition for MRE, and vice versa, MRE is the semantic definition of RC. At its core RC uses the process of *materialisation* and *exceptionality* in order to perform entailment. Materialisation refers to the process of converting defeasible implications into their classical counterparts e.g. $\alpha \mathrel{|\!\sim} \beta$ has the material counterpart $\alpha \rightarrow \beta$. A knowledge base that has undergone materialisation will be denoted as $\overrightarrow{\mathcal{K}}$. Exceptionality is the metric used to distinguish between specificity levels for any given $\mathcal{K}$ [13].

Definition 2.4 (Exceptionality). For a given $\mathcal{K}$ and α, α is said to be exceptional for $\mathcal{K}$ **iff** $\mathcal{K} \mid\!\approx \top \mathrel{|\!\sim} \neg\alpha$

Statements within a knowledge based are ranked from 0 to n according to how exceptional they are, with the final ranking resulting in more specific statements receiving a higher ranking. Statements which are not retractable are placed in the infinite rank. When entailment checking is performed, first the entire materialised knowledge base is checked to see if it entails the negated antecedent of the query. If it does, then this means the query is exceptional in relation to the knowledge base. The first rank, 0 is then discarded and the remaining ranks (including the infinite rank) are tested. In more mathematical terms, given some query, say $\alpha \mid\sim \beta$, we are initially testing if the union of all ranks entails $\neg\alpha$, i.e. $\overrightarrow{\mathcal{K}}_\infty \cup \overrightarrow{\mathcal{K}_0} \cup \overrightarrow{\mathcal{K}_1} \cup ... \overrightarrow{\mathcal{K}_n} \models \neg\alpha$. If it does, then this means that $\alpha \mid\sim \beta$ is exceptional with regards to $\mathcal{K}$ and one or more ranks must be eliminated.

This process continues until either the antecedent is no longer exceptional or the infinite rank is the only rank left. Classical entailment is used thereafter for performing reasoning with the knowledge base.

In addition to a semantic definition, there is an algorithm for Rational Closure, which consists of two parts: a *BaseRank* algorithm presented by [9], that provides a unique ranking of formulas within a knowledge base, as well as the Rational Closure algorithm (which utilises BaseRank). The pseudocode [5] is defined for the aforementioned algorithms below:

Algorithm 1 BaseRank

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

Algorithm 2 RationalClosure

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

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

Table 1: Base Ranking for knowledge base from section 2.4.1

Using the above statements, an example can now be explained. Given a knowledge base such as:

bird $\mid\sim$ flies
bird $\mid\sim$ wings
robin $\rightarrow$ bird
penguin $\rightarrow$ bird
penguin $\mid\sim \neg$ flies

Using the BaseRank algorithm, one obtains the ranking found in table 1. Following from that, we can then consider the query: p $\mid\sim \neg$f ? The Rational Closure process is as follows:

(1) Take the union of all ranks and test the entailment given the negated antecedent of the query: $\mathcal{K}_\infty \cup \overrightarrow{\mathcal{K}_0} \cup \overrightarrow{\mathcal{K}_1} \models \neg$p
(2) Given that the above holds, $\mathcal{K}_0$ is removed.
(3) We then again test the entailment $\mathcal{K}_\infty \cup \overrightarrow{\mathcal{K}_1} \models \neg$p
(4) Since the entailment no longer holds, the process is stopped. Classical entailment is used and p $\mid\sim \neg$f holds.

Practically, Rational Closure can be performed using the above algorithms with the use of existing SAT solvers [11].

2.5 Defeasible Explanation

The computing of minimal subsets is inadequate for use with non-monotonic logics and there are a number of reasons for this. Firstly, standard justifications cannot account for knowledge which is not part of the justification itself, but still a part of the knowledge base, which may prevent an inference being drawn from the justification. We show a simple example of this, using the KB from section 2.1, as follows: first, consider the query p $\rightarrow$ f. One possible justification for this query is:
$\mathcal{J} = \{$ b $\rightarrow$ f, p $\rightarrow$ b$\}$
However, there is another statement within the KB that makes this justification invalid, that is: p $\rightarrow \neg$f.

Secondly, it cannot take into consideration when the removal of one statement from a justification prevents an inference from being drawn, and the removal of more statements allows for the inference to be redrawn. We illustrate this predicament with the use of a somewhat complex example:

bird $\mid\sim$ animal
bird $\mid\sim$ wings $\wedge$ flies
penguin $\mid\sim$ bird
penguin $\mid\sim$ heavy
heavy $\wedge$ wings $\mid\sim \neg$ flies
insects $\mid\sim \neg$ animal
insects $\mid\sim$ wings $\wedge \neg$ flies
ladybug $\mid\sim$ insect
ladybug $\mid\sim$ small
small $\wedge$ wings $\mid\sim$ flies

With this knowledge base, we can posit the query $(p \vee l) \wedge \neg(p \wedge l) \mid\sim f$. For typographic compactness, we write this as $p \oplus l \mid\sim f$ (p XOR l $\mid\sim$ f).

One minimal justification for this query is:
$\mathcal{J}_0 = \{b \mid\sim w \wedge f, p \mid\sim b, p \mid\sim h, h \wedge w \mid\sim \neg f, i \mid\sim w \wedge \neg f, l \mid\sim i, l \mid\sim s, s \wedge w \mid\sim f\}$

For clarity, the statements relating to penguins (p) are shaded blue and red is for ladybugs (l). This justification allows for one to conclude $p \mid\sim \neg f$ and $l \mid\sim f$ and by extension, $\mathcal{J}_0 \mid\approx p \oplus l \mid\sim f$.

Removing $p \mid\sim h$ from $\mathcal{J}_0$ produces $\mathcal{J}_1$, with $\mathcal{J}_1 \not\mid\approx p \oplus l \mid\sim f$:

$\mathcal{J}_1 = \{b \mid\sim w \wedge f, p \mid\sim b, h \wedge w \mid\sim \neg f, i \mid\sim w \wedge \neg f, l \mid\sim i, l \mid\sim s, s \wedge w \mid\sim f\}$

With this new justification, we conclude both $p \mid\sim f$ and $l \mid\sim f$, which means $p \oplus l \not\mid\sim f$.

One could then remove $l \mid\sim s$, producing $\mathcal{J}_2$:

$\mathcal{J}_2 = \{b \mid\sim w \wedge f, p \mid\sim b, , h \wedge w \mid\sim \neg f, i \mid\sim w \wedge \neg f, l \mid\sim i, s \wedge w \mid\sim f\}$

This time we can assert that $p \mid\sim f$ and $l \mid\sim \neg f$ and thus, $p \oplus l \mid\sim f$. Consequently, $\mathcal{J}_2 \mid\approx p \oplus l \mid\sim f$ once again.

Given the above problems, it is evident that nonmonotonicity adds several layers of complexity to defeasible explanations. Some work has been done which aims to address the difficulties presented by the above: weak and strong justifications.

2.5.1 Weak Justifications. Weak justifications are used by Chama [6] and Wang [16]. Their work focused on procedural and algorithmic approaches to computing defeasible justifications for description logic (Chama) and propositional logic (Wang) respectively. Chama modified the Rational Closure algorithm to compute justifications which account for defeasibility, following the intuition that minimal subsets can still be used for defeasible explanations. This is due to the fact that once statements are ranked and the irrelevant (i.e. more general ones) discarded, they are turned into their classical counterparts through materialisation. In this case, unlike the classical one however, not all subsets will be valid and applicable to the query and consequently some are removed. Everett et al. then [8] adapted Chama's weak justification to Relevant and Lexicographic Closure (for PL) and were also able to develop a set of properties for weak justification for the case of RC, that are based on a strengthened version of the KLM postulates. This result shows the clear link between weak justifications for defeasible reasoning and justifications in the classical case.

Wang [16] developed a defeasible entailment justification algorithm that used the KLM framework for propositional logic. Horridge's justification finding algorithm for description logics, as well as Chama's defeasible justification algorithms were translated into a propositional context in order to produce the final algorithm. This was then transformed into a software tool which is publicly available[1].

2.5.2 Strong Justifications. Weak justifications are not logic agnostic, and thus there is a need for a different framework for explanations that can be generalised to both monotonic and nonmonotonic logics. Brewka et al. [4] developed such a notion for logic programming and answer set semantics, though their approach can be generalised to most other logics. Their approach is known as *strong explanations* and is based upon earlier work which was concerned with inconsistency, namely *strong inconsistency* [3]. This was achieved by strengthening the standard concept of inconsistency for nonmonotic logics by adding a constraint on the classical explanation: information from within the knowledge base, but outside of the justification(s), cannot render the justification false. Given that justifications are one type of expression of an explanation, we refer to justifications which have this criterion of strong inconsistency as *strong justifications*.

Consider again a modified, defeasible version of the KB from section 2.1:

```
bird |~ flies
penguin |~ bird
penguin |~ ¬ flies
special penguin |~ bird
special penguin |~ flies
```

In this context, there are two justifications for the query $s \mid\sim f$:

(1) $\mathcal{J}_1 = \{s \mid\sim f\}$
(2) $\mathcal{J}_2 = \{s \mid\sim p, p \mid\sim b, b \mid\sim f\}$

However, $\mathcal{J}_2$ does not qualify as a strong justification since adding $\{p \mid\sim \neg f\}$ causes the entailment to fail.

Strong justifications have been defined for Rational Closure as an extension of weak justifications [8] using the approach of Brewka et al. [4]. One important caveat of this approach is that it is not always possible to extend a weak justification in order to create a strong one, and vice-versa. That is, a strong justification is not always necessarily an extension of a weak one [8].

3 RESEARCH MOTIVATION AND OBJECTIVES

With this project, there are some broad research aims. The first is to create a thesis that provides an overview of classical and defeasible explanations. The second aim is to attempt to connect the notions of strong justifications, a language and formalism agnostic concept, with that of the more procedural defeasible reasoning concepts. Weak justifications, as they have been formalised, only apply to the case of Rational Closure. We seek to characterise weak justifications more generally to rational defeasible entailment and test if these properties can satisfy the properties of strong justifications.

Should weak justifications satisfy these properties, we would further investigate the connection between generalised strong and weak justifications. More specifically, is it the case that every strong justification can be represented by means of extending a generalised weak justification? What are the implications of the aforementioned being true or false in terms of how we define defeasible justifications? If they cannot satisfy all the properties, we would then research why. Namely, determining which do and do not and why - it may be, for example, due to the nature of the properties (being too strict, for instance) or because generalised weak justifications are incompatible with the nature of strong justifications.

[1] KLMDefeasibleJustificationTool

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Given the theoretical nature of this work, no ethical or legal problems have been anticipated. There will be no collection or storing of any personal data. Copyrighted works will not be required and relevant papers will be referenced. Evaluation of the work will be primarily done by the supervisor and exam committee.

5 PROJECT PLAN

The first year will involve developing a strong theoretical understanding of work. This will consist of:

(1) Auditing the Logics for AI course in order to develop a solid understanding of the fundamentals, namely propositional and description logic.
(2) Regular 1-1 meetings with the supervisor to discuss papers, namely: [4], [8], [6], [16].
(3) Weekly lecture-style meetings with supervisor to learn the fundamentals of defeasible reasoning, such as KLM and Rational Closure

Additionally, where possible, attempt will be made to complete the majority of the background chapters of the thesis.

The second year will consist of:

(1) Reviewing and performing corrections on the thesis chapters which have been written thus far.
(2) Researching generalised weak justifications
(3) Writing drafts of the thesis, performing corrections and submitting

5.1 Risks

Please see section A in the appendix.

5.2 Milestones, Deliverables and Timeline

Please see sections B and C in the appendix.

REFERENCES

[1] Franz Baader and Rafael Peñaloza. 2008. Axiom Pinpointing in General Tableaux. *Journal of Logic and Computation* 20, 1 (11 2008), 5–34. https://doi.org/10.1093/logcom/exn058 arXiv:https://academic.oup.com/logcom/article-pdf/20/1/5/2792224/exn058.pdf

[2] Mordechai Ben-Ari. 2012. *Mathematical logic for computer science.* Springer Science & Business Media.

[3] Gerhard Brewka, Matthias Thimm, and Markus Ulbricht. 2019. Strong inconsistency. *Artificial Intelligence* 267 (2019), 78–117. https://doi.org/10.1016/j.artint.2018.11.002

[4] Gerhard Brewka and Markus Ulbricht. 2019. Strong explanations for nonmonotonic reasoning. *Description Logic, Theory Combination, and All That: Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday* (2019), 135–146.

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

[6] Victoria Chama. 2020. *Explanation for defeasible entailment.* Master's thesis. Faculty of Science.

[7] John Etchemendy. 1988. Tarski on Truth and Logical Consequence. *The Journal of Symbolic Logic* 53, 1 (1988), 51–79. http://www.jstor.org/stable/2274427

[8] Lloyd Everett, Emily Morris, and Thomas Meyer. 2021. Explanation for KLM-Style Defeasible Reasoning. In *Southern African Conference for Artificial Intelligence Research*. Springer, 192–207.

[9] Michael Freund. 1998. Preferential reasoning in the perspective of Poole default logic. *Artificial Intelligence* 98, 1-2 (1998), 209–235.

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

[11] Adam Kaliski. 2020. An overview of KLM-style defeasible entailment. (2020).

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

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

[14] Hector J Levesque. 1986. Knowledge representation and reasoning. *Annual review of computer science* 1, 1 (1986), 255–287.

[15] Hector J Levesque and Gerhard Lakemeyer. 2001. *The logic of knowledge bases.* MIT Press.

[16] Steve Wang, Thomas Meyer, and Deshendran Moodley. 2022. Defeasible Justification Using the KLM Framework. In *Southern African Conference for Artificial Intelligence Research*. Springer, 187–201.

A RISKS AND RISK MANAGEMENT STRATEGIES

Scale is 1 to 5 where 1 represents low impact or probability and 5 represents high impact or probability.

Risk Description	Probability	Impact	Mitigate	Monitor	Manage
Work that solves one or more parts of the research problem is released before the thesis is completed	2	5	Publish small papers for conferences as quickly as possible	Regularly check relevant journals for new papers	Meet with supervisor and change scope of dissertation
Get stuck/not making enough progress	2	4	Meet weekly with supervisor to discuss progress and show results	Check how much time is being spent on certain problems to see if there is a bottleneck	Meet with supervisor and act on advice or shift to different part of the problem and then revisit original part
Lose too much time due to external internship	2	4	Liase with company to reduce scope of internship tasks where necessary and be consistent with working on MSc work during internship	Track progress of MSc tasks and ensure they are completed before deadlines	Request an early end to internship
Bad time-management and incorrect time-boxing of work	3	5	Speak with other students to develop a guideline of how long certain tasks will require and setup a calendar	Monitor how long tasks take and check against the guideline	Restructure calendar and meetup with supervisor to discuss task prioritisation

B DELIVERABLES

Note: where needed, some small papers will be written for various conferences.

Activity/Deliverable	Start	End	Comments
Learning	**13-03-23**	**31-07-23**	This is a rough estimate, learning is an ongoing process. The dates serve as rough guidelines by which one should have a decent understanding of the content.
Proposal	**01-06-23**	**08-08-23**	
Proposal Document	01-06-23	16-07-23	
Proposal Presentation	16-07-23	08-08-23	
Thesis			
First Draft	13-03-23	31-08-24	3 week gap is to allow for supervisor feedback.
Final Draft	21-09-24	21-12-24	
Project Artefacts	**01/01/25**	**31/01/25**	
Website	01/01/25	15/01/25	
Poster	16/01/25	31/01/25	

C GANTT CHART

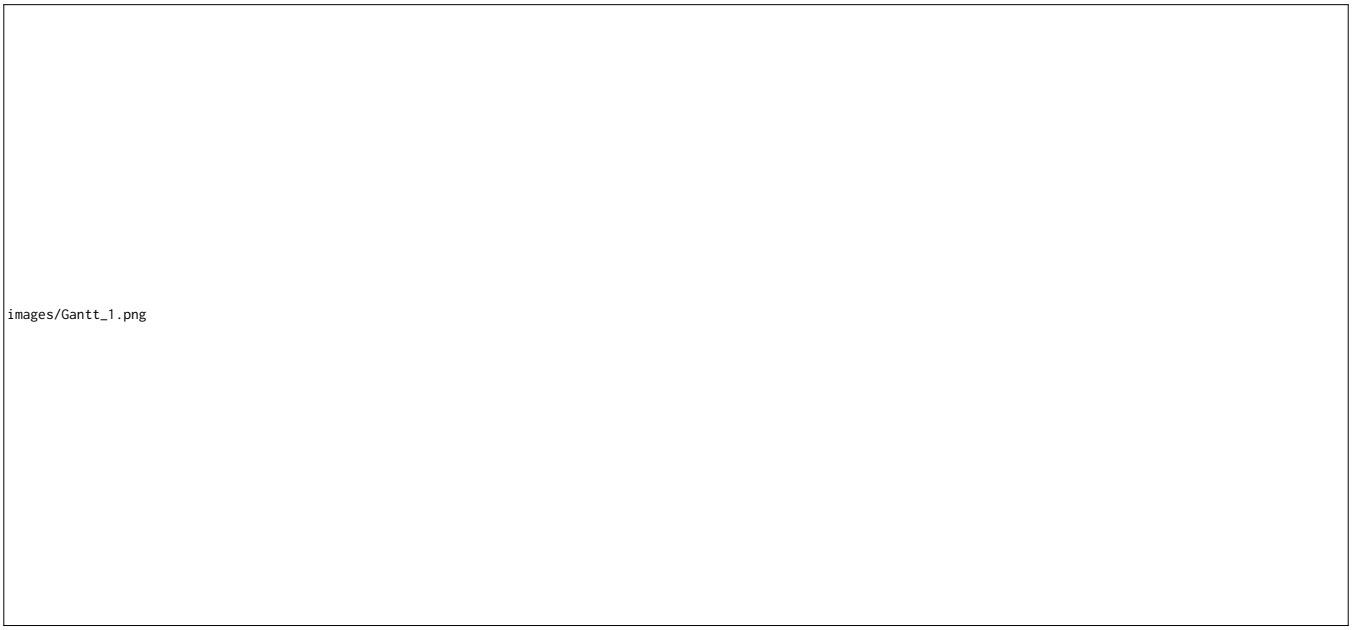

Figure 1: Gantt Chart