Extending Defeasible Reasoning beyond Rational Closure

Jethro Dunn
dnnjet002
University of Cape Town
Cape Town, South Africa

ABSTRACT

Classical reasoning is only able to make conclusions about a knowledge base given formulas with definite outcomes. In the real world there are many uncertain factors and many inconsistencies (such as grammar rules in the English language). We will use defeasible reasoning which allows for these exceptions and this paper will discuss the use of methods of creating hierarchical structures for a knowledge base. The literature on this topic will guide us in creating a knowledge base generator, comparing two algorithms for rational closure and using XAI to create a user interface.

KEYWORDS

Propositional logic, defeasible reasoning, non-monotonic reasoning, rational closure, lexicographic closure, artificial intelligence

1 INTRODUCTION

Propositional logic and defeasible reasoning have been researched for several decades, with new research building off foundational ideas. With significant improvements in artificial intelligence (and specifically large language models) over the last decade, AI has seen a large increase in research and attention. This, combined with improved computational capacity, have made accessing computationally expensive algorithms much easier. This allows for greater access to defeasible reasoning and its associated algorithms. Defeasible reasoning will be the focus of this literature review with an analysis of two of the most common forms of closure associated with it (namely rational and lexicographic closure).

Propositional logic is the overarching field that will be used in this literature review. It is a branch of mathematics which uses "propositions" or statements, which are either true or false[1]. These rules and the associated syntax provide the basis for working on defeasible reasoning. This paper will show how atoms are connected by operators to form formulas, which can be evaluated. It will be shown that classical propositional logic (using classical entailment) has rigid rules, while defeasible reasoning builds on this by also allowing statements to typically be true or false[16].

To formally define defeasible reasoning we will use the KLM model (Kraus, Lehmann, Magidor model)[13]. This framework is used to rank statements by their relative priority. This priority and which statements are selected change depending on the form of closure used, but they all use KLM as the fundamental model. From ranking statement, conclusions can be made which are said to entail from the knowledge base. This is where defeasible entailment is used, to represent information that is part of a knowledge base.

The last three sections will focus on how my project will use the ideas of propositional logic to create a knowledge base generator, two algorithms for rational closure and a user interface. The knowledge base generator will be used to provide logically sound knowledge bases to the algorithms. These two algorithms will use this data so that given a question (in the form of a logical entailment) they can state if it is true or false. The final part of the project will be creating a user interface which uses explanatory artificial intelligence to provide a step by step account of how our algorithms reached their conclusions.

2 PROPOSITIONAL LOGIC

Propositional logic is a form of reasoning, adhering to specific rules, which is used to evaluate knowledge and information [1][8]. It is used as a foundation on which the reasoning models evaluated in this literature review are built [18]. Propositions are statements or pieces of information, single pieces of information, *atoms*, can be assigned either true (T) or false (F) values. Connectives can then be used to join atomic statements and these *formulas* can be evaluated as being true or false. By doing this, natural languages can be encapsulated by these much more compact statements, which can be evaluated to determine if they are true or false.

2.1 Syntax

Propositional logic has a formalised language meaning that it uses specific notation. The following subsections describe the three main components that are used to represent propositional logic.

2.1.1 *Atoms.*

Atoms are statements or representations of statements which can be assigned true or false values. They are pieces of information which can be used in combination with boolean operators to form formulas which can be evaluated to true or false.

An example of an atom is w which represents the statement "wings". More complex statements can also be encapsulated by an atom, such as "The tide is strong tonight" can be represented by t or it could be represented as $tides$. In the literature and in this paper, we will tend to use one character to represent an atom.

The set of all atoms is finite and are referred to as $\mathcal{P}$ [12]

2.1.2 *Boolean Operators.*

Boolean operators are used to join two formulas together and compare them. They are used to form truth tables which result in a true or false outcome. Specific symbols are used to represent different connections between formulas. With the exception of not, which is unary (meaning it only acts on one atoms), the connectives used are all boolean [1]. The following are the connectives that are used in the literature:

$\neg$ (negation)

$\wedge$ (and)

$\vee$ (or, inclusive)

$\rightarrow$ (if then)

$\leftrightarrow$ (if and only if)

2.1.3 Formulas. The combination of atoms and operators can create formulas, which provide more complex combinations of statements. These are referred to as propositional formulas (or fml). Lower case Greek symbols are used to represent formulas that can be derived from a knowledge base, where $\mathcal{L}$ is the set of all these formulas [1]. The notation to represent various formulas in $\mathcal{L}$ is [1]:

$$\begin{aligned} fml &::= p && \text{for some } p \in \mathcal{P} \\ fml &::= \neg fml \\ fml &::= fml\ op\ fml \\ op &::= \vee \mid \wedge \mid \rightarrow \mid \leftrightarrow \end{aligned}$$

Using this notation, an example of a formula could be $\alpha \rightarrow \beta$, where $\alpha, \beta \in \mathcal{L}$.

2.2 Object-Level Semantics

Within propositional logic, the object-level refers to all parts of a language used to model knowledge[12]. Object level semantics are comprised of formulas which use boolean operators and atoms to compare propositions within a knowledge base.

2.2.1 Valuations. Valuations, also referred to in the literature as $interpretations$ or $worlds$4, are a key concept to object-level semantics. They determine how truth values are assigned to formulas. This is important, as it then allows for satisfaction and entailment. The notation for this will be used throughout this paper and is as follows[5]:

- Given set $\mathcal{P}$= p,q,r, ...
- Valuation u maps each atom p $\in \mathcal{P} \in$ T,F

Where u(p) = "T" mean p is true and u(q) ="F" mean q is false. In this paper, this will be written as u = p$\overline{q}$.

2.2.2 Truth Tables. With atoms, boolean operator, formulas and valuations discussed, it is now possible to include a truth table to demonstrate how boolean operators affect propositions, using the correct notation. Truth tables are used to evaluate atoms or formulas. The following is a simple truth table comparing only α and β, but more complex tables can be used where the truth of a valuation is carried forward and used in the next valuation.

α	β	$\alpha \vee \beta$	$\alpha \wedge \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$
T	T	T	T	T	T
T	F	T	F	F	F
F	T	T	F	T	F
F	F	F	F	T	T

Figure 1: Truth Table for Boolean Operators

The first two column represent two distinct statement. The third column shows α and β. The forth shows α or (inclusive) β. The fifth shows if α then β. The intuition behind this is that the outcome is reliant on the antecedence (α) and consequent (β) making promises. For example, if the antecedent promises to bake a cake, if the consequent wins a game. It can be represented as either the consequent wins the game (T) or they lose the game (F) and the antecedent either bakes the cake (T) or does not bake the cake (F). It is then easier to understand, for example, when the antecedent bakes a cake (T) but the consequent does not win (F), then it results in being false. The last column reads as: if and only if α, then β. To be true as a statement, the consequent must be true if the antecedent is true, and vice versa.

2.3 Meta-Level Semantics

Meta-level semantics are used to evaluate knowledge about some knowledge [12]. The meta-level is where concepts like entailment are used, with their functionality used to evaluate the validity of knowledge within a knowledge base.

2.3.1 Entailment. Entailment is a key concept in Propositional logic. It is a meta-level semantic used to provided knowledge about a knowledge base. The notation is as follows, given some knowledge base $\mathcal{K}$ and a statement about the knowledge base $b \rightarrow f$, we can say $\mathcal{K} \models$ b → f. This is classical entailment, assuming this statement is true, it is read as: in $\mathcal{K}$ it follows that if its a bird then it can fly.

In the next chapter, defeasible reasoning will be discussed. In this form of reasoning a different form of entailment will be used. It is used in the same way but instead uses the notion of typicality to show that that something is usually true. For example $\mathcal{K} \mathrel{|\!\approx} b \rightarrow f$, read as: given knowledge base $\mathcal{K}$ it typically follows that if it is a bird then it can fly.

2.3.2 Satisfiability. Another key meta-level semantic for propositional logic is satisfiability. A statement or formula can be said to be satisfiable if and only if there exists some interpretation where it is true [1].Conversely, if there is no situation where a formula is true (in other words it is always false), then it is said to be unsatisfied. The notation for this is as follows, where an atom q is assigned some truth value by valuation u.

- satisfied: $u \Vdash q$ where $q = True$
- unsatisfied: $u \nVdash q$ where $q = False$

2.4 Object vs Meta Levels

It is important to note the differences between Object and Meta levels, as confusing them may result in obtaining or abstracting too much or too little information. Object-level describes interactions within some knowledge, whereas the meta-level describes information about the knowledge. The key differentiating factor is what level they act on. The meta-level is a higher level of abstraction compared to the object-level. This is best shown by example, where the first is a comparison of propositions within a knowledge base and the second is checking if some knowledge is entailed from a knowledge base:

(1) $b \rightarrow w$ where $b, w \in K$
(2) $K \models A$

A visual way to view the difference between object and meta levels is the following, where the meta-level is knowledge about the entire knowledge base, and the object level is between the knowledge in the knowledge base:

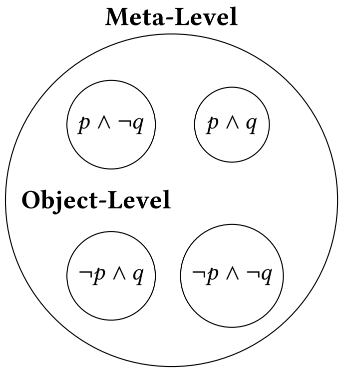

Figure 2: Illustration of meta-level vs. object-level

3 DEFEASIBLE REASONING

Classical reasoning is limited by its inability to model information that does not act consistently, where statements are not true under all circumstances. This is limiting in the real world, where there are often exceptions to typical behaviour. For example, the statement "birds **always can** fly" has been shown to be incorrect (since penguins exist), so this statement needs to change to be "birds **typically can** fly"[5]. The notation is as follows:

- $b \vdash f$ classical implication
- $b \mid\!\sim f$ defeasible implication

Defeasible reasoning is a way to make information represent typical scenarios rather than definite scenarios. This defeasible information can be represented by conditional assertions [16] such as in the example above.

The notation for defeasible reasoning is also different to classical propositional logic. The following symbols will be used through the rest of this literature review to represent defeasible concepts:

- $\mid\!\sim$ "usually is the case"
- $\mid\!\approx$ "usually follows on"

3.1 KLM Framework

The Kraus, Lehmann and Magidor framework (or KLM) is a framework used to determine entailment in a defeasible context, using the concept of typicality[13]. This typicality is shown above, but formally it can be denoted by $\mid\!\sim$ and for $\alpha, \beta \in \mathcal{L}$, the defeasible implication $\alpha \mid\!\sim \beta$ is read as "typically, if α then β [5]. This framework proposed six relations for preferential consequence relations (a formal framework for non-monotonic reasoning) to help formalise defeasible reasoning [12].The KLM framework sets out structures to create ranked interpretations[16].

3.1.1 Ranked interpretation. A ranked interpretation is a function $\mathcal{R}$ that maps statements from 0 to ∞(infinity)[5] Given some knowledge base $\mathcal{K}$, where P = b,f,p representing birds, fly and penguin, we can create a ranked interpretation $\mathcal{R}$. To set up the ranking, the lower levels represents the most typical valuations, with level 0 being the lowest. Conversely the highest level, denoted by ∞, represents a case so atypical that it is impossible [5]. The levels therefore go from 0 to $\mathbb{N}$, where $\mathbb{N}$ represent arbitrary levels of decreasing typicality, with the top level as ∞.

The following ranked interpretation is one possible ranking that could be constructed:

∞	$a\overline{b}c\ a\overline{b}\overline{c}$
1	$\overline{a}bc\ ab\overline{C}\ abc$
0	$ab\overline{c}\ \overline{a}\overline{b}c\ \overline{a}bc$

Figure 3: Ranked Interpretation of $\mathcal{P}$

This is an arbitrary ranking, with no formulas dictating typicality. An example of what can be deduced from this ranking is that $a\overline{b}c$ is impossible to occur and $ab\overline{c}$ is one of the most typical cases. Later in this chapter, rational closure will show how rankings can be created using formulas and how to interpret these rankings.

3.1.2 Entailment. Within the KLM framework, [13] defined six relations for classical entailment which meta-level relations should conform to. Lehmann and Magidor later introduced a seventh , rational monotonicity (RM), which suggests that if something new is learnt and it does not contradict current beliefs, then no information or conclusions should have to be changed [16]. This was formulated due to concerns about CV, which are not pertinent to this literature review. This paper is concerned only with the first six postulations for defeasible entailment. When a defeasible entailment relation follows these postulations it can be said to be LM-rational. These are the postulations by [13][16]:

(1) $\dfrac{\mathcal{K} \models \alpha \leftrightarrow \beta, \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \beta \mid\!\sim \gamma}$ **(LLE)**

(2) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma, \mathcal{K} \mid\!\approx \beta \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \alpha \vee \beta \mid\!\sim \gamma}$ **(Or)**

(3) $\dfrac{\mathcal{K} \mid\!\approx \alpha \rightarrow \beta, \mathcal{K} \mid\!\approx \gamma \mid\!\sim \alpha}{\mathcal{K} \mid\!\approx \gamma \mid\!\sim \beta}$ **(RW)**

(4) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma, \mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta}{\mathcal{K} \mid\!\approx \alpha \wedge \beta \mid\!\sim \gamma}$ **(CM)**

(5) $\mathcal{K} \mid\!\approx \alpha \mid\!\sim \alpha$ **(Ref)**

(6) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma, \mathcal{K} \not\mid\!\approx \alpha \not\mid\!\sim \beta}{\mathcal{K} \mid\!\approx \alpha \wedge \beta \mid\!\sim \gamma}$ **(RM)**

(7) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta, \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta \wedge \gamma}$ **(And)**

3.2 Extended KLM Framework

The core of the KLM framework for non-monotonic reasoning shows how defeasible entailment and ranked interpretations can be formed as a base case. The following sections show how this model can extend entailment in various forms.

3.2.1 ranked entailment. Ranked entailment is all conclusions that every ranked interpretation satisfies[12]. Symbolically ranked entailment can be shown as $\alpha \mid\!\sim \beta$, $\mathcal{K} \mid\!\approx_R \alpha \mid\!\sim \beta$ [16]. This is still a monotonic relation, to make it non-monotonic the concept of minimal ranked entailment will be used[12].

3.2.2 Minimal Ranked Entailment. Minimal Ranked entailment establishes a preference between consequence operators [16], for example $Cn_1(\mathcal{K}) \preceq CN_2(\mathcal{K})$. Minimal ranked entailment forms part of the minimal ranked model, discussed in rational closure, as it forms an extended form of rational closure.

3.3 Rational Closure

Rational closure is a non-monotonic closure and is first of two forms of defeasible entailment that satisfy LM-rationality[16] (the second will be lexicographic closure). Being non-monotonic means that new information can change the conclusion, allowing for additional information to be added to change (or improve) a ranked $\mathcal{K}$. Rational closure is seen as the most conservative form of non-monotonic closure, where it attempts to apply as few assumptions as possible to the knowledge base[5].

Rational closure uses ranking with two formal terms, the first is *between* ranked models where lower ranked models are more typical and, conversely, higher ranked models are less typical[9]. Using the same intuition, the term *within* ranked models of $\mathcal{K}$, are where valuations further up are less typical and ones lower down are more typical[5].

The following notation is used in the literature:

$\mathcal{R}$	ranked interpretation
$\mathcal{R}^{RC}_{\mathcal{K}}$	rational closure over a knowledge base
$\preceq_{\mathcal{K}}$	partial order

3.3.1 Base Ranks. The base rank is a propositional statement α (for some $\mathcal{K}$) which is a measure of how exceptional (or atypical) α is, in $\mathcal{K}$ [5][9]. Base ranks are used to determine the level where α becomes consistent with the typicality of $\mathcal{K}$. It quantifies the minimum level of typicality at which α becomes consistent with the defeasible rules in K. That is, the $br_{\mathcal{K}}(\alpha)$ is defined as the smallest r where α is not exceptional with respect to $\mathcal{E}^{\mathcal{K}}_r$[5].

3.3.2 Minimal Ranked model. The minimal ranked model is a ranking of a knowledge base using minimal ranked entailment. Unlike ranked entailment, minimal ranked entailment is non-monotonic, so it follows that the minimal ranked model is also non-monotonic [12]. This is an important point as this is the first non-monotonic model explored in this paper. This introduces a better way to model the real world, where new or changing information necessitates changes to conclusions.

The following is an example of a minimal ranked model[5]:
Given $\mathcal{P} = b, f, p$, with statements:

(1) p $\rightarrow$ b
(2) b $\mid\sim$ f
(3) p $\mid\sim$ $\neg$f

The minimal ranked model can be construed as follows:

∞	$p\bar{b}f$ $p\bar{b}\bar{f}$
2	pbf
1	$\bar{p}b\bar{f}$ $pb\bar{f}$
0	$\bar{p}\bar{b}\bar{f}$ $\bar{p}\bar{b}f$ $\bar{p}bf$

Figure 4: Ranked Interpretation of $\mathcal{P}$

Intuitively it is assumed in the minimal ranked model that statements are typical until proven otherwise, so all propositions begin on level 0. The following describes how propositions are moved to created the ranked model in figure 4:

(1) p $\rightarrow$b: p$\bar{\text{b}}$f, p$\overline{\text{bf}}$ are impossible as penguins have to be birds, so they are place in the ∞ level.
(2) b $\mid\sim$ f: $\bar{p}b\bar{f}$, pbf is push up to level 1.
(3) p $\mid\sim$ $\neg$f: pbf is pushed up another level, since $\bar{p}b\bar{f}$, $pb\bar{f}$, pbf are all on the same level, but p $\mid\sim$ $\neg$f says that in the best bird worlds, penguins typically do not fly.

It is worth noting that in the literature the ∞ level is often not included.

3.3.3 Materialisation. This is a short section explaining materialisation, as it is used in rational closure. Given a defeasible implication $\alpha \mid\sim \beta$, the material counter part is the propositional formula $\alpha \rightarrow \beta$ [16]. For a knowledge base $\mathcal{K}$, the material counter parts are the set of $\alpha \rightarrow \beta$ for all $\alpha \mid\sim \beta \in \mathcal{K}$. This set of propositions is denoted by $\overrightarrow{\mathcal{K}}$.

4 LEXICOGRAPHIC CLOSURE

Lexicographic closure is another form of non-monotonic defeasible entailment which uses a ranked model to represent a given knowledge base [4]. Lexicographic closure is a "bolder" form of closure than rational closure as statements inherit more from general statements than in rational closure. This creates a ranked model with more levels.

Lexicographic closure was proposed by Lehmann [15] and in defining it he provided four informal properties to guide this form of entailment [12]:

- Presumption of Typicality
- Presumption of Independence
- Priority of Typicality
- Respect for Specificity

The presumption of typicality mean that given two statements you will prefer the one with stronger justification than the other. For example, given $\mathcal{K}$:= *bird* $\mid\sim$ *fly*. The Lehmann and Magidor (LM) rational entailment could be $\mathcal{K} \approx$ *bird* $\wedge$ *penguin* $\mid\sim$ *fly*, or it could be $\mathcal{K} \approx$ *bird* $\mid\sim$ $\neg$ *penguin*. Both of these are feasible, but according to the first condition, the former is preferred [12].

The presumption of Independence means that you assume typicality for a statement, unless there is a conflict that suggests it is not typical.

The priority of typicality is used where there is a conflict between the first two guidelines. It resolves this conflict by choosing the presumption of typicality as being more significant.

The fourth guideline is used when any 2 inferences clash, here the implication with the most specific antecedent is preferred. This does not have a perfect formalisation and is specific to the knowledge base. An example of this is give two antecedents, *bird* and *penguin*, *bird* is a more general antecedent compared to penguin. Like wise, with antecedents of *penguin* and *African Penguin*, the *African Penguin* is a more specific antecedent than penguin. From this, the inference with *African Penguin* as the antecedent is the most preferred.

4.1 Ranked Lexicographic Closure

As with rational closure, it is important to rank the interpretations (R_{LC}) [12]. To rank the implications a level of "seriousness" is given to each implication and preference will be given to those that contradict less serious implications. To understand "seriousness", Lehmann uses the following two properties to define it [15]:

- 1. Set size: where violating a smaller set is less serious
- 2. Specificity of elements: less specific elements cause a lower level of violation

When constructing the ranked lexicographic closure, the second rule (specificity) is applied first and then any further conflicts are settled using the first rule [12]. With these rules in place, the levels can be defined with the highest rank being ∞ and the lowest rank being 0.

5 PROTOTYPICAL COMPARED TO PRESUMPTIVE ENTAILMENT

This section focuses on how prototypical and presumptive entailment differ in a defeasible context. Both these concepts are ways of selecting or separating conflicting statements from statements which are consistent with the knowledge base.

5.1 Prototypical

Rational closure is one of the forms of closure that uses prototypical entailment. Earlier in this literature review, rational closure was said to apply as few base assumptions as possible to the knowledge base. Similarly, it infers the fewest inferences, compared to other forms of non-monotonic closure[12].

Prototypical reasoning says that atypical members of a class do not inherit from that class, while typical members do inherit [16]. An example of this is that since penguins are atypical, they do not inherit the traits of a bird (having wings).

5.2 Presumptive

Lexicographic closure is an example of a form of closure that uses presumptive entailment [16]. This uses the inverse logic of prototypical entailment, where it is assumed that members of a class do inherit from other members, until proven otherwise. This means that penguins are assumed to have wings (since birds typically have wings). Only if there is a formula showing that penguins should not have wings, can it be said that penguins do not have wings.

6 KNOWLEDGE BASE CREATION

Artificial intelligence (AI) has seen a rise in public popularity over the last few years, but it has been a source of research in propositional logic for several decades. One of the key problems with building AI algorithms for propositional logic is getting enough data to train and test models. Although algorithms for rational closure do not require as much data as large language models, there are very few knowledge base sets available to train these models.

For this reason, the first part of our project is to create a knowledge base generator(KBG), which will generate a knowledge base given some parameters as input (such as $\mathcal{K}$ size).

6.0.1 Report Generator. This generator, created by Kukich, takes in data and creates a report[14]. The generator performs two main functions, it infers semantic messages from the data provided and then maps those to phrases.

[14] Uses stock markets as an example of taking in data and generating a report from it. It shows how using a data generator, called Ana, it can take in this data and produce a report in a natural language.

As part of our data generation it could be useful to integrate this, as a way to validate the knowledge bases created. For example, if $\mathcal{K}$ is generated with atoms a, b, c, and the statement $a\bar{b}c$ is created, this could be turned into a sentence in a natural language (which can then be more easily checked by a human).

This could form part of either or both our user interface and KBG, as a way for the user to see what the statement mean in a natural language.

6.0.2 Genomator. The paper[3] describes a Genomator, a method to generate synthetic data using SAT solvers (boolean satisfiability solvers).

This algorithm works by taking in some example data, finding the patterns in this data and then applying these patterns to generate synthetic data.

In [3] the example of using this on medical papers to generate synthetic data is presented. It could be possible to use these ideas to create a KBG, which takes in some example $\mathcal{K}s$ and generates synthetic $\mathcal{K}s$.

6.0.3 Limitations. There are a lot of scientific papers on data generations, especially for AI (and machine learning), but there is limited information on KBG in the literature. For our project we will use ideas of data generation from other disciplines and apply them to propositional logic knowledge base generation.

7 ALGORITHM PROJECT SECTION

This section will review the existing algorithm for rational closure and then discuss it drawbacks.

7.1 RC Algorithm

Given the literature examined in previous chapters, it is possible to examine a common rational closure algorithm.

The first step is to create an algorithm to determine the base rank for each formula in $\mathcal{K}$, where the *br* links to some level from $0 \ldots \mathbb{N}, \infty$ [12]. This program takes in $\mathcal{K}$ as input, but mostly acts over $\overrightarrow{\mathcal{K}}$, resulting in outputting sets of classical implication.

The second program required is the rational closure algorithm. This takes in $\mathcal{K}$ and a defeasible query on the knowledge base, then returns either true, if it is entailed, or false, if it is not entailed[12][7].

7.2 Algorithm Comparison

For this project, we can compare two different algorithms for rational closure. We could judge our algorithms based off of speed and accuracy to ascertain which algorithm is best for calculating defeasible entailment.

There are multiple possible algorithms that attempt to solve problems in propositional logic. For the purposes of our project, we could compare the standard rational closure algorithm (as described in the previous section) to ABA or ABA+.

7.2.1 Assumption-base Algorithm (ABA). This algorithm works by initially guessing a set with a SATm solver[17]. We then generate conclusions based on this and if these conclusions are correct the algorithm ends, otherwise it goes to the next candidate set.

Research has been done to extended this to ABA+. This considers preference on assumptions, rather than on defeasible rules[6]

8 USER INTERFACE AND XAI

The last part our our project involves creating a user interface used to explain the algorithm we used. The primary goal of this system is to show the user how our algorithm reached its conclusion. This field of study is referred to as explainable artificial intelligence (XAI).

In machine learning it is often unclear how conclusions are derived, but in the real world logical questions need to be verifiable. For example, if our program were used to determine the rights of someone detained by police, it is important that the steps used by our program are verifiable[2]. For this reasoning it is vital that our program can show its steps in a way that is user friendly.

8.0.1 XAI. There is no formal definition of XAI, but Gunning [11] defines it as being able to "produce explainable models, while maintaining a high level of learning performance (prediction accuracy); and enable human users to understand [...] artificial intelligence". This will be the definition used to guide our implementation of XAI.

8.1 Interpretability vs Explanation

Our user interface needs to be explainable, not just interpretable. This will allow a wider range of users to access the utility provided by our programme.

8.1.1 Interpretability. This is a system which can be understood by a person, such as python code or a uniform distribution [10]. It requires some knowledge about a field, but is understandable with this knowledge.

8.1.2 Explainability. Explainability can be seen as an extension to interpretability, where people need additional information and explanations to make them understand[10]. For example, comments in code or a short description of how a model works.

9 CONCLUSIONS

This paper covered sections of literature pertaining to propositional logic, with a focus on defeasible reasoning. Within this sphere of research, it was found that classical entailment was too limiting to model the real world, so defeasible entailment was introduced as a better form of entailment for our project.

The KLM model was then discusses, showing how ranked interpretations could be used to come to a conclusion and what rules needed to be followed to be considered LM-rational. This was then extended by Lehmann and Magidor [16] to facilitate ranked entailment and produce a ranked entailment based on the minimal rank.

Non-monotonic reasoning was found to be preferred to monotonic reasoning, as in the real world new information should allow for conclusions to be changed. For this reason rational closure and lexicographic closure where discussed, with one main difference being that rational closure uses prototypical entailment, compared to lexicographic closure which uses presumptive entailment. These were found to both be reasonable and valid forms of closure, but for the purposes of our project, we will use rational closure.

The last three chapters of this literature review were about the three parts of our project. For KBG, it was found that there were limited resources on generating knowledge bases for propositional logic. Research on data generation, such as creating synthetic data via SAT solvers, provides some similar ideas on taking in a data set and using the patterns identified to generate data. From reviewing the literature, it is possible to compare different rational closure algorithms. The first algorithm could be a basic RC algorithm proposed by [7] and the second could be an ABA algorithm[17]. The final part of the project will be creating a user interface incorporating XAI. It will be important to ensure explainability of our algorithm and to provide a logical breakdown of how our algorithms reach their conclusions.

REFERENCES

[1] Ben-Ari, M. *Mathematical logic for computer science.* Springer Science & Business Media, 2012.

[2] Billi, M. e. a. Argumentation and defeasible reasoning in the law. *Multidisciplinary Scientific Journal 4*, 4 (2021), 897–914.

[3] Burgess, M. e. a. Privacy-hardened and hallucination-resistant synthetic data generation with logic-solvers.

[4] Casini, G., Harrison, M., Meyer, T., and Swan, R. Arbitrary ranking of defeasible subsumption.

[5] Casini, G., Meyer, T., and Varzinczak, I. Taking defeasible entailment beyond rational closure. In *Logics in Artificial Intelligence: 16th European Conference, JELIA 2019, Rende, Italy, May 7–11, 2019, Proceedings 16* (2019), Springer, pp. 182–197.

[6] Cyras, K. Toni, F. Properties of aba+ for non-monotonic reasoning.

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

[8] Gallier, J. H. *Logic for Computer Science Foundations of Automatic Theorem Proving.* Universtity of Pennsylvania, 2003.

[9] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence 226* (2015), 1–33.

[10] Guidotti, R., e. a. A survey of methods for explaining black box models. *ACM Computing Surveys 51*, 5 (2018), 1–42.

[11] Gunning, D., A. D. Darpa's explainable artificial intelligence program. *AI Magazine 40*, 2 (2017), 44–58.

[12] Kaliski, A. An overview of klm-style defeasible entailment. *NA* (2020).

[13] Kraus, S., Lehmann, D., and Magidor, M. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence 44*, 1-2 (1990), 167–207.

[14] Kukich, K. Design of a knowledge-based report generator. *ACL Anthology* (1983), 145–150.

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

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

[17] Lehtonen, T. Wallner, J. J. M. *Algorithms for Reasoning in a DefaultLogic Instantiation of Assumption-BasedArgumentation.* IOS Press, 2022.

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