Scalable Defeasible Reasoning V2 Literature Review

Evashna Pillay
University of Cape Town
Cape Town, South Africa
plleva005@myuct.ac.za

ABSTRACT

Knowledge representation and reasoning formally represents and enables reasoning of information about the world, this is an approach used in artificial intelligence. Reasoning systems based on classical propositional logic are able to perform many reasoning tasks correctly and quite efficiently, however it is likely that as more information is attained, a contradiction occurs. While there has been some work on developing efficient systems for defeasible reasoning, much remains to be done. The goal of this project is to build on a previous honours project to design and implement algorithms for different forms of propositional defeasible reasoning, and to evaluate the extent to which these algorithms are scalable.

CCS CONCEPTS

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

KEYWORDS

artificial intelligence, knowledge representation and reasoning, defeasible reasoning

1 INTRODUCTION

Propositional logic will be defined as well as its syntax and semantics. A brief overview of defeasible reasoning and its underlying approaches will be presented, including entailment and preferential reasoning. Following this, the KLM approach will be introduced, thereafter an overview of rational closure will be provided along with the algorithm to compute it. Finally, the Boolean satisfiability problem will be outlined, additionally two approaches; the DPLL and Semantic tableaux based algorithms will be briefly described.

2 PROPOSITIONAL LOGIC

2.1 Motivation

Propositional logic is the study of logic including reasoning about knowledge or information that human language represents in logical statements. To form ideal statements, we combine and modify propositional statements using connectives such as equivalence, and, or, not and implication, however these statements can be more complex [1] [9]. Every base statement can be attributed to a truth value. The truth value of a combination of statements depends on these base statements. As a result, we can evaluate and rationalize a statement or set of statements from which new conclusions are isolated.

2.2 Syntax

The language of propositional logic consists of atomic assertions (i.e., atoms or propositional letters) [9]. The proposition atom is also called a variable or symbol. For example, the intuition is that $Socrates$ is shorthand for a statement, or proposition such as "Socrates is a philosopher." Atoms are then assigned truth values, they can either be true, indicated by T, or false, indicated by F. The set of statements or propositional atoms will be denoted as $\mathcal{P}$ and is finite. Each propositional atom will be represented using meta variables (e.g., p, q, r,...). Atoms can be combined by a set of connectives (i.e., $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) to create more complex propositional formulas [9]. The negation operator is unary and therefore takes in one operand, whereas the other operators are all binary and takes in two operands. See table below defining connectives and their symbols.

Operator Symbol	Operator Name
$\neg$	Negation
$\wedge$	Conjunction
$\vee$	Disjunction
$\rightarrow$	Implication
$\leftrightarrow$	Equivalence

The set of all propositional formulas are denoted by $\mathcal{L}$, and represented by lowercase Greek letters such as α, β, γ,. Formulas of $\mathcal{L}$ are defined recursively such that $\mathcal{L}$ is defined as follows: for some $p \in \mathcal{P}$ and $\alpha, \beta \in \mathcal{L}$, $\{p,\ \neg\alpha,\ \alpha \wedge \beta,\ \alpha \vee \beta,\ \alpha \rightarrow \beta, \alpha \leftrightarrow \beta\}$. Additionally, constants of $\mathcal{L}$ include $\top$ and $\bot$.

Further explanation on $\mathcal{L}$ below:

- $\top$ is a tautology and is referred to as "top", which is a statement that is always T. In contrast, $\bot$ refers to a statement which is always F.
- $\neg\alpha$ takes in a single statement that reads T when it is F and F when it is T.
- $\alpha \vee \beta$ is a disjunction of two statements such that "α or β" is T if and only if either α or β are T.
- $\alpha \wedge \beta$ is a conjunction of two statements such that "α and β" is T if and only if both are T.
- $\alpha \rightarrow \beta$ means α implies β, such that when α is T, β is always T. Intuitively, the same as "if α then β" and "α only if β ." The exception is when α is T and β is F.
- $\alpha \leftrightarrow \beta$ is read as "α if and only if β," this is T if and only if α and β are both T or both F.

We have defined the syntax of propositional logic. To determine the truth value of a set of propositional formulas we will use a model-theoretic definition, using semantics.

2.3 Semantics

The semantics of logic enables a systemic analysis of the language whereby the meaning of T is defined. The notion of a statement being T is called $Satisfaction$. Propositional atoms are assigned truth values known as valuations and referred to as $worlds$ or $interpretations$, this will be denoted by the Latin alphabet.

The truth tables [13] below illustrate the truth values of the atoms (left) and the resulting evaluations of the propositional formula(s) mentioned in Section 2.2:

- $\neg\alpha$

α	$\neg\alpha$
T	F
F	T

- $\alpha \vee \beta$

α	β	$\alpha \vee \beta$
T	T	T
T	F	T
F	T	T
F	F	F

- $\alpha \wedge \beta$

α	β	$\alpha \wedge \beta$
T	T	T
T	F	F
F	T	F
F	F	F

- $\alpha \to \beta$

α	β	$\alpha \to \beta$
T	T	T
T	F	F
F	T	T
F	F	T

- $\alpha \leftrightarrow \beta$

α	β	$\alpha \leftrightarrow \beta$
T	T	T
T	F	F
F	T	F
F	F	T

Definition 2.3.1 A valuation u is a function, $u : P \mapsto \{T, F\}$. Every propositional atom in the language is assigned either T or F.

E.g., $\mathcal{P} := \{p, q, r\}$, a random valuation $u \in \mathcal{U}$ could be $p\bar{q}r$, which is read as p and r being T and q being F.

Definition 2.3.2 The valuation u satisfies the atom p if an atom is T in the valuation u.

E.g. $p \in \mathcal{P}$, $u(p) = \text{T}$, then $u \models p$, $\models$ denotes that the valuation u satisfies the atom p.

Definition 2.3.3 For any valuation u that satisfies $\alpha \in \mathcal{L}$, let $\hat{\alpha} := \{u \in \mathcal{U} \mid u \models \alpha\}$. For $u \in \hat{\alpha}$, the valuation u is referred to as a model of α. The set of all models of α is denoted by $\hat{\alpha}$.

- If $\hat{\alpha}$ contains at least one model, then α is satisfiable.
- If $\hat{\alpha}$ is empty, then α is unsatisfiable.
- If $\hat{\alpha} = \mathcal{U}$, then α is a tautology, and T for every possible valuation.

E.g., $p \wedge \neg p$ is unsatisfiable, and $p \vee \neg p$ is a tautology.

Definition 2.3.4 Two formulas $\alpha, \beta \in \mathcal{L}$, $\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$. α and β are logically equivalent, this is a meta-level concept not defined as part of $\mathcal{L}$ itself. Logical consequence and logical equivalence represent deductions that can be made about statements in $\mathcal{L}$.

Definition 2.3.5 A knowledge base, $\mathcal{K} \subseteq \mathcal{L}$, is a finite set of propositional formulas. A knowledge base represents a set of facts about the $world$.

E.g., $\mathcal{K} = \{p, p \to q\}$ mean there are two known facts that hold, that is p and $p \to q$. Each formula in a knowledge base restricts the set of valuations in $\mathcal{U}$ that satisfy the knowledge base.

It is possible that for every $u \in \mathcal{U}$ that $u \not\models \mathcal{K}$, or that there is no model of $\mathcal{K}$, therefore the knowledge base is unsatisfiable. The set of valuations satisfying $\mathcal{K}$ is the empty set, $\emptyset$.

3 DEFEASIBLE REASONING

3.1 Motivation

Defeasible reasoning is a form of non-monotonic reasoning, as it allows the reasoner to temporarily dismiss commonly held beliefs when presented with new contradictory information [14]. Atypical reasoning scenarios in which humans fundamentally think differently, whether owing to their beliefs, context, or other conditions, are likewise addressed by defeasible reasoning. AI systems may mimic human thinking more precisely with defeasible reasoning and its methods than with classical reasoning and the underlying idea of non-monotonic reasoning.

Let's use a familiar example to illustrate the defeasible reasoning approach.

E.g., "Tweety is a bird" and the reasoner is aware of the following:

(1) Most birds can typically fly.

(2) Birds have wings.

(3) Penguins are birds.

(4) Tweety is a penguin.

The reasoner can deduce that Tweety is a bird from the consequent of (3). The reasoner can then deduce that Tweety can fly because (1) follows (3). However, we know that penguins cannot fly due to their biological makeup. As a result, penguins are distinct from other birds. With the additional information made available to the reasoner, the reasoner's understanding that birds normally fly is weakened. A variety of non-monotonic reasoning systems have previously been developed. The KLM framework specifically for defeasible reasoning provides a preferential approach to construct defeasible systems [10]. The framework is based on conditional logic and preferential logic [5] [16]. The defeasible reasoning algorithms generated by this technique has proven to be intrinsically computable and as efficient as classical reasoning algorithms [11].

3.2 A Preferential Approach

Preferential reasoning sets the semantic foundation that is shared by many non-monotonic logic. A *preferential consequence relation* is indicated by $\mathrel{|\!\sim}$ in contrast to $\vdash$. It is a set of conditional assertions, written as $\alpha \mathrel{|\!\sim} \beta$ where $\alpha, \beta \in \mathcal{L}$, with the intended interpretation of, from α we are prepared to conclude β unless we receive contradictory information. This implies that any defeasible assertion can be retracted after discovering contradictory information.

Definition 3.2.1 The consequence relation denoted by $\mathrel{|\!\sim}$ represents defeasible implication. Additionally consequence relations satisfy the KLM postulates.

E.g., $bird \mathrel{|\!\sim} flies$ reads as "birds typically fly."

Preferential semantics allows the preference of a valuation u over a valuation v, as a result u will be considered before v. This preference can be interpreted in a variety of ways. One possible interpretation is that typical valuations are chosen. An example to illustrate this would be a valuation in which $birds\ can\ fly$ over a valuation in which $birds\ cannot\ fly$.

Definition 3.2.2 Preferential interpretation $\mathcal{P}$ is a triple $\langle \mathcal{S}, l, \prec \rangle$ where $\mathcal{S}$ denotes an infinite set of states, l is a function such that $l : \mathcal{S} \mapsto \mathcal{U}$ maps states to valuations, and $\prec$ denotes a strict partial order on $\mathcal{S}$.

For $\alpha \in \mathcal{L}$ and $\mathcal{P}$, $[[\alpha]]^{\mathcal{P}} := \{s \in \mathcal{S}, \mathcal{S} \in \mathcal{P}, l(s) \Vdash \alpha\}$. $[[\alpha]]^{\mathcal{P}}$ is the set of all states in $\mathcal{P}$ such that the valuation associated with each state satisfies α.

A subset of preferential interpretations are ranked interpretations. In a ranked interpretation, if given two states then the states are either equal in rank, or either one is preferred. This differs from preference interpretations using a non-modular partial order, in which two states might be incomparable in a variety of ways. As opposed to preferential interpretations, ranked interpretations don't allow duplicate states.

Definition 3.2.3 A ranked interpretation is a function $\mathcal{R} : \mathcal{U} \mapsto \mathcal{N} \cup \{\infty\}$. For every $i \in \mathcal{N}$, if there is some $u \in \mathcal{U}$ where $\mathcal{R}(u) = i$, then $v \in \mathcal{U}$ such that $\mathcal{R}(v) = j$ where $0 \leq j < i$ [7].

The rank of $u \in \mathcal{U}$ in $\mathcal{R}$ shall be denoted by the notation $\mathcal{R}(u)$. Ranks in this case imply that valuations with a lower-rank are more typical, or regular, while infinite-rank valuations are impossible.

E.g., For $\mathcal{P} := \{p, q, r\}$, the ranked interpretation $\mathcal{R}$ is represented below:

∞	$\bar{p}\bar{q}r$ $\bar{p}q\bar{r}$
2	$\bar{p}qr$ $p\bar{q}r$.
1	pqr $pq\bar{r}$ $p\bar{q}\bar{r}$
0	$\bar{p}\bar{q}\bar{r}$

- $\mathcal{R}\ (\bar{p}\bar{q}r) = \mathcal{R}\ (\bar{p}q\bar{r}) = \infty$, therefore $\bar{p}\bar{q}r$ and $\bar{p}q\bar{r}$ are impossible.
- $\mathcal{R}\ (\bar{p}qr) = \mathcal{R}\ (p\bar{q}r) = 2$ and $\mathcal{R}\ (pqr) = \mathcal{R}\ (pq\bar{r}) = \mathcal{R}\ (p\bar{q}\bar{r}) = 1$.
- $\mathcal{R}\ (\bar{p}\bar{q}\bar{r}) = 0$, $\bar{p}\bar{q}\bar{r}$ is the most preferred and signifies the most typical world.

Definition 3.2.4 Given a defeasible knowledge base $\mathcal{K}$ and a defeasible implication $\alpha \mathrel{|\!\sim} \beta$, $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{P}} \alpha \mathrel{|\!\sim} \beta$ if and only if for every preferential model $\mathcal{P}$, of $\mathcal{K}$, $\mathcal{P} \Vdash \alpha \mathrel{|\!\sim} \beta$.

Preferential entailment is analogous to classical entailment such that if a query is satisfied by every preferential model $\mathcal{P}$ of a knowledge base $\mathcal{K}$, then it is preferentially entailed by the knowledge base $\mathcal{K}$.

Ranked preferential entailment functions are analogous to propositional logic. This means that ranked entailment is monotonic.

Definition 3.2.5 Given a knowledge base $\mathcal{K}$ and a defeasible implication $\alpha \mathrel{|\!\sim} \beta$, $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{R}} \alpha \mathrel{|\!\sim} \beta$ if and only if for every ranked interpretation $\mathcal{R}$, where $\mathcal{R} \Vdash \mathcal{K}$, $\mathcal{R} \Vdash \alpha \mathrel{|\!\sim} \beta$.

3.3 KLM-Style Defeasible Entailment

Defeasible reasoning unlike propositional logic does not define a fixed method to determine defeasible entailment. Thus to determine whether a defeasible implication is entailed by a knowledge base, the method must only adhere to the rationality properties defined by Lehmann and Magnidor [12].

The rationality properties for knowledge base $\mathcal{K}$ and propositional formulas α, β, γ [9]:

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

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

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

3.4 Rational Closure

Rational closure is an LM-rational method of defeasible entailment. We will first define minimal ranked entailment, materialisation, Base Rank Algorithm before providing an overview of the Rational Closure Algorithm.

Definition 3.3.1 *Minimal Ranked Entailment.* Ranked entailment is similar to the principle of ranked interpretations in the sense that the lower the ranked model, the more likely there exists a minimum ranked model in the ordering. This principle involves a partial order for all ranked models of knowledge base $\mathcal{K}$ referred to as $\preceq_{\mathcal{K}}$. The rational closure of a knowledge base $\mathcal{K}$ is based on finding this minimum ranked model [6] as the minimum ranked model must satisfy some defeasibile implication for the knowledge base $\mathcal{K}$ to entail it [7].

Definition 3.3.2 *Materialisation.* The material counterpart of a defeasible implication $\alpha \mid\sim \beta$ is the propositional formula $\alpha \rightarrow \beta$. The material counterpart of a defeasible knowledge base $\mathcal{K}$ is denoted as $\overrightarrow{\mathcal{K}}$ and represents the set of material counterparts, $\alpha \rightarrow \beta$, for every defeasible implication $\alpha \mid\sim \beta \in \mathcal{K}$.

E.g., $bird \mid\sim flies$ is replaced with $bird \rightarrow flies$.

Definition 3.3.3 *The Base Ranks Algorithm.* Base Ranks is the initial step performed when implementing rational closure to $\mathcal{K}$ to determine the minimum ranked model. Each formula in $\overrightarrow{\mathcal{K}}$ is mapped to a rank in $\mathcal{N} \cup \{\infty\}$, this denotes the exceptional subset of $\mathcal{K}$, $\mathcal{E}_n^{\mathcal{K}}$.

A walk-through of the Base Ranks Algorithm:

(1) Separate $\mathcal{K}$ into classical (i.e., $\mathcal{K}_C$) and defeasible (i.e., $\mathcal{K}_D$) components.

(2) $\overrightarrow{\mathcal{K}_D}$ in $\mathcal{E}_0^{\mathcal{K}}$ and K_C as is.

(3) For each $\alpha \rightarrow \beta \in \mathcal{E}_0^{\mathcal{K}}$, determine if $\mathcal{E}_0^{\mathcal{K}} \cup K_C \models \neg\alpha$.

(4) If true, α is *exceptional* and all formulas within $\mathcal{E}_0^{\mathcal{K}}$ with α are moved to $\mathcal{E}_1^{\mathcal{K}}$.

(5) Rank $\mathcal{R}_0^{\mathcal{K}}$ is assigned to the formulas $\mathcal{E}_0^{K} \backslash \mathcal{E}_1^{\mathcal{K}}$.

(6) Repeat the process for the next subset, i.e., $\mathcal{E}_1^{\mathcal{K}}$ until no more formulas need to be copied/assigned.

(7) Add the final rank $\mathcal{R}_\infty^{K}$ to K_C.

Definition 3.3.4 *The Rational Closure Algorithm.* Rational Closure will make use of $\mathcal{R}^K$ from the Base Rank Algorithm, given a knowledge base $\mathcal{K}$ and some defeasible implication, $\alpha \mid\sim \beta$ [4].

A walk-through of the Rational Closure Algorithm:

(1) Check if $\overrightarrow{\mathcal{K}}$ entails $\neg\alpha$.

(2) If this is not the case, α is compatible with $\mathcal{K}$. Then check if $\overrightarrow{\mathcal{K}}$ entails $\alpha \rightarrow \beta$.

(3) If this is the case, α is incompatible with $\mathcal{K}$. The most preferred rank is withdrawn from $\mathcal{R}^{\mathcal{K}}$. The resulting knowledge base will be denoted as $\mathcal{K}'$.

  (3.1) If $\mathcal{R}^{\mathcal{K}'}$ is an empty set, then $\mathcal{K} \not\approx \alpha \mid\sim \beta$.

  (3.2) If $\mathcal{R}^{\mathcal{K}'}$ contains at least one rank, we return to (1) with $\mathcal{K}'$.

E.g., The knowledge base $\mathcal{K} = \{b \mid\sim f, p \rightarrow b, p \mid\sim \neg f, R \rightarrow b, b \mid\sim w\}$ is a continuation of the penguins and birds example mentioned in *Section 3.1. Robins are birds* (i.e., $R \rightarrow b$) and *birds typically have wings* (i.e., $b \mid\sim w$) are added to the knowledge base $\mathcal{K}$.

Using Base Rank Algorithm:

- $K_C = \{p \rightarrow b, R \rightarrow b\}$
- $K_D = \{b \rightarrow f, p \rightarrow \neg f, b \rightarrow w\}$.
- p is exceptional and found in $\mathcal{E}_0^{\mathcal{K}}$.

To illustrate the Base Rank Algorithm $\mathcal{R}^{\mathcal{K}}$, a table of results is provided below.

∞	$p \rightarrow b, R \rightarrow b$
1	$p \rightarrow \neg f$
0	$b \rightarrow f, b \rightarrow w$

Using Rational Closure Algorithm:

- Determine if $\mathcal{K}$ entails the query $p \rightarrow \neg f$, that is *penguins don't fly.*
- Determine if $\overrightarrow{\mathcal{K}} \models \neg p$.
- We deduce there is no model u of $\overrightarrow{\mathcal{K}}$ such that $u \Vdash p$. Therefore, we retract the most preferred rank from $\mathcal{R}^{\mathcal{K}}$, that is $\mathcal{R}_0^{\mathcal{K}}$.

To illustrate the Rational Closure Algorithm, an adjusted table of results is provided below.

∞	$p \rightarrow b, R \rightarrow b$
1	$p \rightarrow \neg f$

We repeat the first step:

- Determine if $\overrightarrow{\mathcal{K}} \models \neg p$. This is not the case.
- There exists a model u of $\overrightarrow{\mathcal{K}}$ such that $u \Vdash p$, e.g, $\{pbr\bar{f}\}$.
- Determine if $\overrightarrow{\mathcal{K}}$ entails $p \rightarrow \neg f$.
- This is the case, $p \rightarrow \neg f$ is a formula within $\overrightarrow{\mathcal{K}}$ and as such, any model of $\overrightarrow{\mathcal{K}}$ is also a model of $p \rightarrow \neg f$.

Thus, we can conclude that $\mathcal{K} \mid\approx p \rightarrow \neg f$.

4 SAT SOLVERS

4.1 Boolean Satisfiability Problem

The Boolean Satisfiability Problem is an NP-complete problem that determines if there exists some interpretation which satisfies a given boolean formula [2]. Sat-solvers work with propositional formulas in conjunctive normal form (CNF) [15]. CNF is a language containing clauses and connectives (i.e., $\wedge$). Atomic literals and $\wedge$ connectives are used to form clauses. Atom literals can be assigned T or F values. Given a propositional formula in CNF, $(b \vee p) \wedge (b \vee \neg p)$ and we assign b to true then $(T \vee p) \wedge (T \vee \neg p)$.

4.2 Semantic Tableaux Based Complete Algorithm

The tableau is constructed by decomposing a formula into sets of atomic literals, resulting in a tree-like tableau [1]. Each branch ends with a complementary pair of formulas, that is referred to as a closed branch, or contains a set of non-contradictory literals, that is referred to as an open branch. Each open branch represents a model for the given formula. When no further decomposition can take place, the construction is complete. The initial formula is found to be unsatisfiable when there is a clash. A clash occurs when literals for an atom are found within the same subset, this results in a contradiction. To determine a formula's satisfiability, a completed tableau is required.

4.3 DPLL

The DPLL is a sat-solver algorithm first published in 1960 [3]. Many individuals have made changes since then to the algorithm which have improved its proficiency and allowed for further optimisations [8]. DPLL is a backtracking algorithm. The algorithm allows for the input of a propositional formula in CNF format, in which T is returned if the formula is satisfiable or F is returned if the formula is unsatisfiable. A branching procedure is executed where some atom in α is set to a random truth value. Branching continues until an assignment satisfies α and T is returned, or α is F and the algorithm backtracks.

Backtracking involves retrieving the recent branching assignment and re-branching with a different assignment. If there are no new assignments to branch to, then backtrack. If there exists no new branches to be taken and we cannot backtrack, false is returned and α is unsatisfiable.

An example below to illustrate the DPLL algorithm:

Given

$$(p \vee \neg q) \wedge (\neg p \vee q)$$

DPLL executed as follows:

- *Branch* $p := T$
- $(T \vee \neg q) \wedge (F \vee q)$
- *Branch* $q := F$
- $(T \vee F) \wedge (F \vee F)$
- **Unsatisfied**
- *Backtrack* $q := F$
- $(T \vee \neg q) \wedge (F \vee q)$
- *Branch* $q := T$
- $(T \vee F) \wedge (F \vee T)$
- **Satisfied**
- *Return true*

5 CONCLUSIONS

Principles of propositional logic were defined and explored, as well as concepts such as entailment and satisfiability. The concept of defeasible reasoning was defined and rendered classical reasoning useless when adding contradictory information to an existing knowledge base. Defeasible reasoning was shown to be more expressive than propositional logic due to its non-monotonicity. Additionally, rational closure was defined and its algorithm outlined. It was observed that rational closure reduces defeasible entailment to classical propositional entailment. Thus, it can be implemented using sat solvers. Sat solvers were briefly discussed, specifically the semantic tableaux and DPLL based algorithms, in which the DPLL sat solver was shown to be more efficient.

REFERENCES

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

[2] Stephen A Cook. 1971. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing.* 151–158.

[3] Martin Davis, George Logemann, and Donald Loveland. 1962. A Machine Program for Theorem-Proving. *Commun. ACM* 5, 7 (July 1962), 394–397. https://doi.org/10.1145/368273.368557

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

[5] D. M. Gabbay. 1985. Theoretical Foundations for Non-Monotonic Reasoning in Expert Systems. In *Logics and Models of Concurrent Systems*, Krzysztof R. Apt (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 439–457.

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

[7] Ivan Varzinczak Giovanni Casini, Thomas Meyer. 2018. Defeasible Entailment: from Rational Closure to Lexicographic Closure and Beyond. In *17th International Workshop on Non-Monotonic Reasoning (NMR).* Arizona, USA, 109–118.

[8] Eugene Goldberg and Yakov Novikov. 2007. BerkMin: A fast and robust SAT-solver. *Discrete Applied Mathematics* 155, 12 (2007), 1549–1561.

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

[10] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44, 1-7 (1990). https://doi.org/10.1016/0004-3702(90)90101-5

[11] Daniel Lehmann. 1999. Another perspective on Default Reasoning. *Annals of Mathematics and Artificial Intelligence* 15 (11 1999). https://doi.org/10.1007/BF01535841

[12] 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

[13] P. D. Magnus, Tim Button, Aaron Thomas-Bolduc, Richard Zach, and Robert Trueman. 2020. *Forall X: Calgary. An Introduction to Formal Logic.* Open Logic Project. 1–101,378 pages.

[14] Drew McDermott and Jon Doyle. 1980. Non-monotonic logic I. *Artificial Intelligence* 13, 1 (1980), 41–72. https://doi.org/10.1016/0004-3702(80)90012-0 Special Issue on Non-Monotonic Logic.

[15] Alexander Nadel. 2009. *Understanding and improving a modern SAT solver.* Ph.D. Dissertation. Tel Aviv University.

[16] Y. Shoham. 1987. A semantical approach to nonmonotonic logics. In *LICS 1987.*