SCADR Literature Review

Joel Hamilton
University of Cape Town
Cape Town, South Africa
joelhamiltondev@gmail.com

ABSTRACT

Knowledge representation and reasoning (KRR) is an approach to artificial intelligence (AI) in which a system has some information about the world represented formally (a knowledge base), and is able to reason about this information. Defeasible reasoning is a non-classical form of reasoning that enables systems to reason about knowledge bases which contain seemingly contradictory information, thus allowing for exceptions to assertions. Entailment checking is the process of determining whether or not a statement can be inferred from a knowledge base. Currently, no programming framework exists that supports defeasible entailment in the propositional logic context. We aim to improve the scalability of defeasible entailment algorithms, as well as implement these algorithms in a form that allows for incorporation into a larger framework.

CCS CONCEPTS

• Theory of computation → Automated reasoning; **Constraint and logic programming**; **• Computing methodologies → Non-monotonic, default reasoning and belief revision**.

1 INTRODUCTION

Artificial Intelligence (AI) has been around for many years, with its two core aspects being machine learning (ML) and knowledge representation and reasoning (KRR) [5]. Our research will focus on the latter. Knowledge representation refers to the notion of using some formal set of symbols or notation in order to represent information about the world. Reasoning refers to the idea of drawing inferences from this information, with the key idea for this research being the use of algorithms to reason about information (i.e., automated reasoning). We will utilise logics (a mechanism for formalising ways to reason [3]) to represent information.

This review will begin by discussing the syntax and semantics of propositional logic [3], before delving into the notion of entailment and automated ways of determining satisfiability. Then in section 4, we present a brief overview of classical reasoning and the shortfalls it presents. Following this, we introduce defeasible reasoning and specifically, the KLM approach to defeasible reasoning [15] [17]. We will then provide an overview of Rational Closure [17], and the algorithm to compute it [8]. We conclude by painting the picture of the gap in the literature of this field which we aim to fill with this project.

2 PROPOSITIONAL LOGIC

Propositional logic is a framework for modelling information about the world. In this framework, statements are built up using *atomic propositions* (*propositional atoms*), which are simply sentences which can be attributed a *truth value* (a value of true or false). Formulas are statements which are built up from a set of propositional atoms which are combined using *Boolean operators*. This section will introduce the syntax and semantics of propositional logic as defined in [3].

2.1 Syntax

2.1.1 Alphabet Definition. The *alphabet* $\mathcal{P}$ used to construct propositional formulas consists of:

(1) A set of propositional atoms: e.g., $\{p, q, r, \ldots\}$
(2) Boolean operators, a subset of which are represented in the following table:

Operator Name	Operator Symbol
Negation	$\neg$
Disjunction	$\lor$
Conjunction	$\land$
Implication	$\rightarrow$
Equivalence	$\leftrightarrow$

Table 1: Boolean operators and their symbols

The negation operator is *unary* (it takes in a single operand), and the other operators are *binary* (they take in two operands).

2.1.2 Formula Definition. A *formula* is a string of symbols from an alphabet defined as in 2.1.1, and the set of all formulas $\mathcal{L}$ is built up recursively in the following way:

- Every atom $p \in \mathcal{P}$ is in $\mathcal{L}$.
- The negation of any atom $p \in \mathcal{P}$, i.e. $\neg p$, is in $\mathcal{L}$
- If α and β are in $\mathcal{L}$, then $\neg\alpha, \alpha \land \beta, \alpha \lor \beta, \alpha \rightarrow \beta$ and $\alpha \leftrightarrow \beta$ are all in $\mathcal{L}$.

The constants $\top$ and $\bot$ are also in $\mathcal{L}$, where they are simply formulas that are always true and always false, respectively.

2.2 Semantics

2.2.1 Formula Semantics. Every formula in $\alpha \in \mathcal{L}$ has a truth value, where the truth value of α is dependent on the meaning of the Boolean operators and the truth values of the atoms which comprise the formula.

An *interpretation* is a function $I : \mathcal{P} \rightarrow \{T, F\}$ which assigns a single truth value to each propositional atom, and $\mathcal{W}$ is the set of all interpretations. The truth value of a formula α under a given interpretation I is written as $I(\alpha)$, and if $I(\alpha)$ is true for some formula α, then it is said that α is satisfied by the interpretation I. This means one of the following conditions must hold:

- α is an atom, e.g. p, and $I(p) = T$
- α is the negation of another formula, e.g. $\neg\beta$, and $I(\beta) = F$.
- α is the conjunction of two other formulas, e.g. $\beta \land \gamma$, and $I(\gamma) = T$

- α is the disjunction of two other formulas, e.g. $\beta \vee \gamma$, and at least one of $I(\beta) = T$, or $I(\gamma) = T$, holds.
- $\alpha = \beta \rightarrow \gamma$, and at least one of $I(\neg\beta) = T$ or $I(\gamma) = T$ holds.
- $\alpha = \beta \leftrightarrow \gamma$, and $I(\beta) = I(\gamma)$.

If an interpretation I satisfies a formula α, then I is called a *model* of α, and $Mod(\alpha)$ is defined as the set of all models of α.

2.2.2 Knowledge Bases. A *knowledge base* is a finite set of propositional formulas, where a knowledge base $\mathcal{K}$ is satisfied by an interpretation I if for every $\alpha \in \mathcal{K}$, $I(\alpha) = T$. The set of models of a knowledge base is the set of interpretations $I \in \mathcal{W}$ which satisfy $\mathcal{K}$. If a given formula α is true in every model of $\mathcal{K}$ (that is, $Mod(\mathcal{K}) \subseteq Mod(\alpha)$), then $\mathcal{K}$ entails α, written $\mathcal{K} \models \alpha$, which means that α is a logical consequence of the knowledge base $\mathcal{K}$. The following section will discuss checking satisfiability of formulas and knowledge bases, as well as how entailment and satisfiability are related.

3 SATISFIABILITY CHECKING

Checking whether or not a formula (or knowledge base) is satisfiable (that is, there is some interpretation under which the formula, or every formula in the knowledge base, is true), is a fundamental problem in mathematics, known as the *Boolean Satisfiability Problem*. While this problem has been proved to be NP-complete (proof available in [12]), there exist algorithms for checking the satisfiability of a knowledge base, such as the *Semantic Tableaux* algorithm for propositional logic, as in [3].

A *satisfiability (SAT) solver* is an algorithm (or a program implementation thereof) which, given a boolean formula, will return whether or not there exists an interpretation which satisfies the formula. Given that every formula in propositional logic is Boolean (has a truth value), there are many implementations of satisfiability solvers which specifically return whether or not a propositional logic knowledge base is satisfiable. A large amount of research has taken place that has looked at optimising the satisfiability checking process.

3.1 Satisfiability Algorithms

For a satisfiability algorithm to return that a particular knowledge base $\mathcal{K}$ in unsatisfiable, it must be the case that there are two contradictory statements (e.g., an atom p, and its negation $\neg p$) that need to be both be true for any interpretation I for it to be a model of $\mathcal{K}$.

3.1.1 The DPLL Algorithm. The *DPLL algorithm* [10] is a complete satisfiability algorithm (it will always return that a problem/formula is satisfiable if it is in fact the case) which works by assigning values to variables speculatively in such a way that tends to return the variable assignment that satisfies the problem relatively quickly. This algorithm makes use of backtracking, as well as utilises a process known as Boolean Constant Propagation (BCP) which constantly ensures that all variables which need to be true to satisfy a problem are kept true. DPLL formed the basis of many satisfiability solvers, however aspects of how DPLL works have been improved upon since it was proposed.

3.1.2 The CDCL Process. GRASP [27] [18], introduced the *Conflict Driven Clause Learning (CDCL)* process, which allows for more efficient satisfiability checking, as it caches the variable assignments which cause potential conflicts, and thus does not rely on chronological backtracking, as in the DPLL algorithm proposed in [10].

3.1.3 Other advancements. Several more advanced SAT solvers have come about which utilise low-level optimisations such as different storage structures and parallelism, in order to obtain better performance [30]. Examples of such SAT-solving algorithms include GridSAT [9], which was proposed in 2003, which is an improvement on Chaff [25], which is itself a relatively efficient solver that had been proposed two years prior. Due to the significant research being done in developing efficient SAT solving algorithms, many implementations of SAT solvers are now available, and thus the ability to check the satisfiability of a propositional formula (or knowledge base) is now as simple as the click of a button.

3.2 Reducing Entailment to Satisfiability Checking

Checking whether or not a statement constructed in propositional logic, as defined in [3], is entailed by a knowledge base can be reduced to checking satisfiability, as if a formula α is true for all models of a knowledge base $\mathcal{K}$, then there should exist no model of $\mathcal{K}$ such that α is false, and thus, we can check entailment as follows:

Input: A knowledge base $\mathcal{K}$, and a formula α such that we wish to determine whether or not $\mathcal{K} \models \alpha$.

(1) Consider the knowledge base $\mathcal{K}' = \mathcal{K} \cup \{\neg\alpha\}$
(2) Use the Semantic Tableaux algorithm, (or an implementation of a more advanced satisfiability solving algorithm) to determine whether or not $\mathcal{K}'$ is satisfiable.
    - If $\mathcal{K}'$ is satisfiable, then we know that there exists at least one interpretation I which is a model of $\mathcal{K}'$ such that $I(\alpha) = F$, and thus $K \not\models \alpha$.
    - If $\mathcal{K}'$ is unsatisfiable, then we know $\mathcal{K} \models \alpha$, as there are no models of $\mathcal{K}'$ such that $I(\alpha) = F$.

A simple example of this is as follows:
If we have a knowledge base $\mathcal{K}$ containing the following statements:

- Every employee pays tax.
- Every student is an employee.
- There exists a student.

with these statements formalised into propositional logic as follows:

- $e \rightarrow t$
- $s \rightarrow e$
- s

And we wish to investigate whether or not the statement "No one pays tax", formalised as $\neg t$, is entailed by the knowledge base, then we will consider the knowledge base $\mathcal{K}' = \mathcal{K} \cup \{\neg(\neg t)\}$, and given that $\neg(\neg t)$ is equivalent to t, we have:

$$\mathcal{K}' = \{e \rightarrow t, s \rightarrow e, s, t\}$$

A SAT solver (or the Semantic Tableaux algorithm) would then return that this knowledge base is in fact satisfiable, and thus we would be able to conclude that the statement "No one pays tax", is not entailed by the knowledge base $\mathcal{K}$, as there is at least one interpretation I which is a model of K' where $I(\neg t)$ is false.

Note that the above algorithm for entailment works solely in the context of *classical reasoning*, which is a reasoning framework which defines how new information is determined to be a logical consequence of a knowledge base. This will be elaborated on in the following section.

4 CLASSICAL REASONING

4.1 Tarskian Operators

Classical reasoning is a framework for inferring information from given knowledge, and while the way in which to reason classically about propositional knowledge bases may seem relatively intuitive (e.g. for $p \rightarrow q$ to be satisfied, it must be the case that if p is true, then q is also true), there exists a formal basis for classical reasoning, which makes use of concepts known as *Tarskian consequence operators* [23] [28] [29]. A *consequence operator* is a mapping from arbitrary sets of formulas in a given logical language to other sets of formulas in the language, and such a consequence operator is *Tarskian* if it satisfies the following set of properties, as outlined in [14]:

Given a knowledge base $\mathcal{K}$ and a consequence operator C_n:

- *Inclusion*: $\mathcal{K} \subseteq C_n(\mathcal{K})$
- *Monotonicity*: if $\mathcal{K} \subseteq \mathcal{K}', \ then \ C_n(\mathcal{K}) \subseteq C_n(\mathcal{K}')$
- *Idempotence*: $C_n(\mathcal{K}) = C_n(C_n(\mathcal{K}))$

Where, in the context of classical reasoning in propositional logic:

- Inclusion means that any knowledge base is a subset of the set of all formulas which it entails.
- Monotonicity means that any information that is entailed by a knowledge base $\mathcal{K}$ will still be entailed by K even if K has new information added to it (additional formulas).
- Idempotence (also referred to as closure) means that deduction (i.e., determining what is entailed by a knowledge base K) using classical reasoning will always produce the entire set of all information that is entailed by K.

4.2 Classical Reasoning Limitations and Possible Solutions

Reasoning in such a way limits the ability to model human reasoning with propositional logic, as the property of monotonicity does not hold in the way that humans think.

For example, if a human knows that it rains every Saturday, and that today is a Saturday, then it makes sense to conclude that it is raining today, however if we are then told that it is not raining today, a human would interpret the addition of this new fact to mean that we have simply come across an exception to the notion of it raining every Saturday. Given that classical reasoning does not allow for exceptions, this example is a clear instance where classical deduction is not a "common-sense" approach to reasoning [6]. Classical reasoning simply notes that a contradiction has occurred, and we are now no longer able to infer that it is raining today, and thus the property of monotonicity cannot be maintained in the presence of somewhat contradictory information. What may have allowed the additional information to not cause a contradiction is if the initial knowledge we had rather stated that "typically, it rains every Saturday", as opposed to the general assertion that it rains every Saturday.

Several approaches to nonmonotonic reasoning exist, with some key approaches having been outlined by Kaliski in [14], namely Belief Revision [2] [1], Default logic [26], Propositional Typicality Logic [4], Circumscription [19] [20], and Nonmonotonic Modal Logic [22] [21] to name a few, however the approach used for this research is the KLM approach, as proposed in [15] [17].

5 KLM-STYLE DEFEASIBLE REASONING

5.1 The KLM Approach

Defeasible reasoning is a nonclassical form of reasoning which allows for the concepts of *typicality*, and *exceptionality*. In the KLM approach [15] [17], the notation of propositional logic is extended to include statements of the form $\alpha \mid\sim \beta$, to be read as "*α typically implies β*", the meaning of which is simply that if α is true, then it is likely that β is true. The formal way in which the language of all formulas $\mathcal{L}$ over an alphabet in propositional logic changes in order to support statements of this nature is as follows:

$$If\ \alpha, \beta \in \mathcal{L},\ then\ we\ add\ \alpha \mid\sim \beta\ to\ \mathcal{L}$$

Note the specific phrasing "we add", as α and β are thus restricted to being formulas of the propositional form as defined in section 2 of this literature review (we can refer to statements of this form as classical statements or classical formulas). In using this restrictive phrasing, it implies there is no nesting of the typical implication operator ($\mid\sim$).

It is important to note that defeasible entailment is not unique (i.e. the information entailed by a knowledge base can differ based on the defeasible entailment approach used), however the KLM framework defines a list of properties which must be satisfied by a defeasible entailment method in order to be considered acceptable, and it is for this reason that the KLM approach is preferred. An in-depth overview of the KLM approach and its intricacies is provided by Kaliski in [14].

5.2 Ranked Interpretations

Before delving into how to reason about a defeasible knowledge base (a knowledge base containing statements which use the $\mid\sim$ operator), we must first define a ranked interpretation [8].

Informally, this structure is simply a ranking of all interpretations in $\mathcal{W}$ (the set of all interpretations), in order of typicality, starting at rank 0 (indicating the most typical interpretations) and ending at rank n, which will include those interpretations which are the least typical, but are still plausible), followed by a single infinite rank, indicating those interpretations which are impossible to occur. In ranked interpretations, no rank can be empty.

Formally, a ranked interpretation is a function $R : \mathcal{W} \rightarrow \mathbb{N} \cup \{\infty\}$, such that $R(I) = 0$ for some $I \in \mathcal{W}$, and for every $r \in \mathbb{N}$, if $R(I) = r$, then for every j such that $0 \leq j \leq r$, there is an $I \in \mathcal{W}$ for which $R(I) = j$. (Note that the definition in [14] uses slightly different symbols, e.g. in place of $\mathcal{W}$ and I. These symbols were changed in order to be consistent with those introduced in section 2 of this literature review.)

If a formula α is true in all non-infinite ranks of a ranked interpretation R, then we say that R "satisfies" α, denoted $r \vdash \alpha$. For defeasible statements (statements which use the $\mid\sim$ operator), we say R satisfies $\alpha \mid\sim \beta$ if in the lowest rank (the most typical rank) where α holds, β also holds. We can construct an ordering on the set of all possible ranked interpretations for a knowledge base $\mathcal{K}$, where $R_1 \leq_{\mathcal{K}} R_2$ if for every interpretation $I \in \mathcal{W}$, $R_1(I) \leq R_2(I)$. There is a unique minimal element R_m of the ordering $\leq_{\mathcal{K}}$ such that for every other ranked interpretation R_i in the set of all ranked interpretations for a given knowledge base, $R_m \leq_{\mathcal{K}} R_i$, as shown in [13].

5.3 The KLM Properties

As mentioned in section 5.1., the KLM approach introduced several properties which need to be met by any form of defeasible entailment [15]. If any given approach to defeasible entailment satisfies all these properties, then such an approach is referred to as *LM-rational*. The extended KLM properties proposed in [17] are as follows:

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

(RW) $\dfrac{\mathcal{K} \approx \alpha \mid\sim \beta,\ \beta \models \gamma}{\mathcal{K} \approx \alpha \mid\sim \gamma}$ (LLE) $\dfrac{\alpha \equiv \beta,\ \mathcal{K} \approx \alpha \mid\sim \gamma}{\mathcal{K} \approx \beta \mid\sim \gamma}$

(Or) $\dfrac{\mathcal{K} \approx \alpha \mid\sim \gamma,\ \mathcal{K} \approx \beta \mid\sim \gamma}{\mathcal{K} \approx \alpha \vee \beta \mid\sim \gamma}$ (And) $\dfrac{\mathcal{K} \approx \alpha \mid\sim \beta,\ \mathcal{K} \approx \alpha \mid\sim \gamma}{\mathcal{K} \approx \alpha \mid\sim \beta \wedge \gamma}$

(RM) $\dfrac{\mathcal{K} \approx \alpha \mid\sim \gamma,\ \mathcal{K} \not\approx \alpha \mid\sim \neg\beta}{\mathcal{K} \approx \alpha \wedge \beta \mid\sim \gamma}$ (CM) $\dfrac{\mathcal{K} \approx \alpha \mid\sim \beta,\ \mathcal{K} \approx \alpha \mid\sim \gamma}{\mathcal{K} \approx \alpha \wedge \beta \mid\sim \gamma}$

Two methods of defining defeasible entailment that are both LM-rational are rational closure [17] and lexicographic closure [16], and two which are not LM-rational [8] are ranked entailment [17] and relevant closure [7]. The following section will show the definition of Rational Closure as per [8] and [14].

5.4 Rational Closure

5.4.1 Using the Unique Minimal Ranked Interpretation. As noted in section 5.2, there is a unique minimal element $R_{\mathcal{K}}$ of the set of all ranked interpretations for a given knowledge base $\mathcal{K}$. If $R_{\mathcal{K}} \vdash \alpha \mid\sim \beta$, then this is written $\mathcal{K} \approx \alpha \mid\sim \beta$, and it is said that $\mathcal{K}$ defeasibly entails $\alpha \mid\sim \beta$, or that $\alpha \mid\sim \beta$ is in the rational closure of $\mathcal{K}$. Given that this defeasible entailment relation can be generated from a ranked interpretation, it is LM-rational, as found by Lehmann and Magidor in [17].

5.4.2 The Base Ranks Algorithm. In order to algorithmically determine whether or not a formula is in the rational closure of a knowledge base, we require the base rank algorithm, and the actual algorithm for checking entailment.

Before showing how the algorithm works, we need to define the *materialisation* of a knowledge base $\mathcal{K}$ as

$$\overrightarrow{\mathcal{K}} = \{\alpha \rightarrow \beta : \alpha \mid\sim \beta \in \mathcal{K}\}$$

This essentially just converts defeasible implication statements into classical implication statements.

(1) First, we create a sequence of materialisations $E_0, E_1, ..., E_{n-1}, E_\infty$ where $E_0 = \overrightarrow{\mathcal{K}}$, and each $E_i = \{\alpha \rightarrow \beta \in E_{i-1} : E_{i-1} \models \neg\alpha\}$ for $i > 0$. (This essentially means that each E_i contains only those statements $\alpha \rightarrow \beta$ in E_{i-1} such that α can be proven false according to those statements not in E_{i-1}). This process terminates when $E_i = E_{i-1}$ (or if $E_{i-1} = \emptyset$) and we set n=i-1 and $E_\infty = E_n$.
(2) We then create a ranking such that $E_n \backslash E_{n-1}$ is in the bottom rank, (call this R_∞) $E_{n-2} \backslash E_{n-1}$ is in the next highest rank, and so on. The ranking becomes such that the classical statements in $\mathcal{K}$ are in the bottom rank (the infinite rank), and statements are more general the higher the rank they are in. If a statement $\alpha \rightarrow \beta$ is in a rank i, then it is said that $\alpha \rightarrow \beta$ has base rank i. What this means is that in every ranked interpretation r of E_i, $\alpha \rightarrow \beta$ will be true in at least one of the interpretations in the most typical rank of r.

5.4.3 Checking Defeasible Entailment. Input: a knowledge base $\mathcal{K}$ and a defeasible implication statement $\alpha \mid\sim \beta$.

(1) First, we check if $\neg\alpha$ is entailed by those statements in the infinite rank.
(2) We then remove sets of classical implications rank by rank, starting at the highest rank and working our way down until we find a rank such that all the statements in that rank and those remaining do not entail $\neg\alpha$ (thus, α is satisfiable w.r.t. the set of statements which are in the current rank and the remaining ranks).
(3) We then check whether or not these remaining statements entail $\alpha \rightarrow \beta$.
(4) If we get to the stage where we only have the bottom rank remaining, and the statements in this rank entail $\neg\alpha$ then we can simply conclude that it is the case that $\mathcal{K} \approx \alpha \mid\sim \beta$.

This algorithm only returns that a statement is defeasibly entailed by a knowledge base if this is true in terms of the minimal ranked interpretation for the knowledge base [11].
The following is an example which may better explain how the entailment check occurs:

If we have a knowledge base

$$\mathcal{K} = \{s \rightarrow \neg x, s \rightarrow c, t \rightarrow x, t \rightarrow s, f \rightarrow s\}$$

which has been ranked according to the Base Ranks algorithm as follows:

Rank 0	$s \rightarrow \neg x, s \rightarrow c$
Rank 1	$t \rightarrow x$
Rank ∞	$t \rightarrow s, f \rightarrow s$

Figure 1: Base Ranking of Knowledge base $\mathcal{K}$

And we wish to investigate whether or not the statement $t \mid\sim x$ is entailed by the knowledge base:

(1) We check if $\neg t$ is entailed by $\overrightarrow{\mathcal{K}}$. It is not, so we throw away the top rank.
(2) We then check if $\neg t$ is entailed by the remaining statements in ranks 1 and ∞. We see that this is not the case, so we check whether $t \rightarrow x$ is entailed by the remaining statements. We see this is true, so we conclude $t \mid\sim x$ is entailed by $\mathcal{K}$.

5.5 Defeasible Reasoning Implementations

It should be obvious that the algorithm for determining whether or not a defeasible statement is in the rational closure of a knowledge base simply reduces to several classical entailment checks, which itself reduces to several classical satisfiability checks, as discussed in section 3.2.

Although much research has been done in terms of optimising classical satisfiability solving, as discussed in section 3, there currently exists no defeasible satisfiability solver which can determine whether or not a defeasible statement is in the rational closure of a given defeasible knowledge base. It is for this reason that there has also been no attempt to optimise a program implementation of rational closure, as there exists no program implementation of it in the first place.

Some work has been done which looks at implementing the ability to reason defeasibly in the context of description logics, such as in [24], but the lack of a similar implementation in the propositional case appears as a large gap in the literature regarding nonmonotonic reasoning, as well as opens up the opportunity for an investigation into the scalability of established LM-rational defeasible entailment algorithms such as Rational Closure [17] and Lexicographic Closure [16].

6 CONCLUSIONS

This review began with an introduction to the concepts of propositional logic as defined in [3], as well as the notions of entailment, and satisfiability. We then presented classical reasoning and how it is simply not expressive enough to model human thought. We then introduced the concept of defeasible reasoning and how this approach is more expressive than propositional logic, given its support for the notions of typicality and exceptionality. We then laid out the KLM Approach [15] [17], and discussed the KLM properties and the concept of LM-rationality. We then provided an overview of rational closure [17], and an algorithm for determining the rational closure of a knowledge base, as presented in [8]. We concluded by pointing out the lack of program implementations of defeasible SAT solvers, and the opportunity that this presents (namely, the ability to investigate and improve the scalability of defeasible entailment algorithms).

REFERENCES

[1] Carlos E. Alchourron and David Makinson. On the logic of theory change: Contraction functions and their associated revision functions. *Theoria*, 48(1):14–37, 1982.

[2] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50:510–530, 06 1985.

[3] Mordechai Ben-Ari. *Mathematical Logic for Computer Science, 3rd Edition.* Springer, 2012.

[4] R. Booth, T. Meyer, and I. Varzinczak. Ptl: A propositional typicality logic. Sep 2012.

[5] Zied Bouraoui, Antoine Cornuéjols, Thierry Denœux, Sebastien Destercke, Didier Dubois, Romain Guillaume, João Marques-Silva, Jérôme Mengin, Henri Prade, Steven Schockaert, Mathieu Serrurier, and Christel Vrain. From shallow to deep interactions between knowledge representation, reasoning and machine learning (kay r. amel group), 12 2019.

[6] G. Casini, T. Meyer, K. Moodley, and I. Varzinczak. Towards practical defeasible reasoning for description logics. Jul 2013.

[7] Giovanni Casini, Thomas Meyer, Kodylan Moodley, and Riku Nortjé. Relevant closure: A new form of defeasible reasoning for description logics. In Eduardo Fermé and João Leite, editors, *Logics in Artificial Intelligence*, pages 92–106, Cham, 2014. Springer International Publishing.

[8] Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. Taking defeasible entailment beyond rational closure. In Francesco Calimeri, Nicola Leone, and Marco Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 182–197. Springer, 2019.

[9] Wahid Chrabakh and Rich Wolski. Gridsat: A chaff-based distributed sat solver for the grid. In *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, SC '03, page 37, New York, NY, USA, 2003. Association for Computing Machinery.

[10] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962.

[11] Michael Freund. Preferential reasoning in the perspective of poole default logic. *Artificial Intelligence*, 98(1):209–235, 1998.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences).* W. H. Freeman, first edition edition, 1979.

[13] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. Semantic characterization of rational closure: From propositional logic to description logics. *Artif. Intell.*, 226:1–33, 2015.

[14] Adam Kaliski. An overview of klm-style defeasible entailment, 2020_.

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

[16] Daniel Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15(1):61–82, Mar 1995.

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

[18] Joaão P. Marques-silva and Karem A. Sakallah. Grasp: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48:506–521, 1999.

[19] John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1):27–39, 1980. Special Issue on Non-Monotonic Logic.

[20] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116, 1986. Reprinted in [**?**].

[21] Drew McDermott. Nonmonotonic logic ii: Nonmonotonic modal theories. *Journal of the ACM (JACM)*, 29(1):33–57, 1982.

[22] Drew McDermott and Jon Doyle. Non-monotonic logic i. *Artificial intelligence*, 13(1-2):41–72, 1980.

[23] J. Donald Monk. The contributions of alfred tarski to algebraic logic. *Journal of Symbolic Logic*, 51(4):899–906, 1986.

[24] Kody Moodley, Thomas Meyer, and Ivan Varzinczak. A protégé plug-in for defeasible reasoning. volume 846, 06 2012.

[25] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient sat solver. In *Proceedings of the 38th Annual Design Automation Conference*, DAC '01, page 530–535, New York, NY, USA, 2001. Association for Computing Machinery.

[26] Raymond Reiter. A logic for default reasoning. *Artificial intelligence*, 13(1-2):81–132, 1980.

[27] João P. Marques Silva and Karem A. Sakallah. Grasp—a new search algorithm for satisfiability. In *Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '96, page 220–227, USA, 1997. IEEE Computer Society.

[28] Alfred Tarski. O pojciu wynikania logicznego. *Przeglad Filozoficzny*, 39(1):58–68, 1936.

[29] Alfred Tarski. On the concept of following logically. *History and Philosophy of Logic*, 23(3):155–196, 2002.

[30] Yakir Vizel, Georg Weissenbacher, and Sharad Malik. Boolean satisfiability solvers and their applications in model checking. *Proceedings of the IEEE*, 103(11):2021–2035, 2015.