Propositional Typicality Reasoning (PTR)

A Project Proposal for an Implementation of Reasoning of Propositional Typicality Logic

Guy Green
University of Cape Town
grnguy001@myuct.ac.za

Andrew Howe-Ely
University of Cape Town
hwland004@myuct.ac.za

1 BACKGROUND

Some basic background is needed to explain some of the terms and concepts used in this document. Propositional logic forms the basis of the work for this project. Propositional logic is a logic system that builds up statements using propositional atoms and logical connectives or operators [1]. Propositional atoms are either true or false, and any sentence built from such atoms is also either true or false. The main such operators are negation (¬), conjunction (∧), disjunction (∨), implication (→) and equivalence (↔). Given a sentence there are many interpretations, each represents a different possible world that assigns a value of true or false to each propositional atom. The sentence as a whole can then be evaluated as true or false. Interpretations are a fundamental part of reasoning in such a system. We call a sentence satisfiable if it evaluates as True for some interpretation of the atoms. We call this satisfying interpretation a Model. A knowledge base is a set of sentences in a language. We say a sentence is entailed by a knowledge base if every model for that knowledge base is also a model for the sentence. Entailment in propositional logic is essentially computing whether the models of a knowledge base are a subset of the models of the sentence being entailed. Entailment is the crux of reasoning in this system.

Defeasibility refers to the ability in a system to make assumptions and then retract that assumption when presented with conflicting information. A defeasible statement tells us what is typically the case. The operator $\mid\sim$ is used to create defeasible conditional statements such as $A \mid\sim B$, read as *typically, if A then B*. Defeasible reasoning is non-monotonic, which informally means that we can retract knowledge learned after new information is added. In contrast, reasoning in propositional logic is monotonic. With this, statements like "birds typically fly" can be made. But, when an exception like "penguins don't fly" occurs, we can reason about this. This is known as the *Tweety* example in the literature. If birds typically fly and *Tweety* is a bird, we first reason that *Tweety* can fly. Then if we learn *Tweety* is a penguin and penguins don't fly, we can reason that *Tweety* doesn't fly, and that *Tweety* is an exceptional bird.

2 PROJECT DESCRIPTION

2.1 The Problem

Propositional Logic is one of the most basic forms of Logics used to describe the world. It is the basis for many of the other Logics. Propositional Logic does have the major problem of not being expressive enough to describe the real world situations and not being able to handle exceptions. Defeasible Reasoning with the Defeasibility Operator has been suggested as an addition to Propositional Logic to attempt to deal with this problem. As an addition to Defeasibility in Propositional Logic, the language Propositional Typicality Logic (PTL) has been suggested. This allows for the idea of typicality anywhere in a propositional logic sentence whereas Defeasibility in Propositional Logic only allows typicality for the antecedent.

Our project focuses on PTL and, in specific, entailment or reasoning for PTL. The main focus of the project is to create an implementation of the two algorithms suggested in Booth et al [3]. These algorithms are called LM-Entailment and PT-Entailment. We have divided the project into three different sections: a theoretical section, an implementation section and an experimental section.

The theoretical section focuses on proving that PT-Entailment and LM-Entailment are appropriate in terms of Entailment for PTL. This is proving that each algorithm satisfies a subset of postulates for entailment for PTL described in Booth et al [3]. Proving that each algorithm satisfies their respective subsets proves that each of the algorithms are appropriate and unique algorithms for entailment in PTL. We will be proving that the algorithms satisfy the definition of entailment in PTL and also produce the correct answer.

The implementation section focuses on creating an implementation for PT-Entailment and LM-Entailment. We will be creating these implementations based on the high-level description of the algorithm in Booth et al [3]. We will test this implementations with sample knowledge bases to check that they are getting the same answer the algorithms in theory, and also if this corresponds to the intuitive understanding of the knowledge base and what should be entailed.

The experimental section will focus on testing that the implementations of PT-Entailment and LM-Entailment using sample knowledge bases. This will be testing whether the algorithms are producing answers that correspond to the intuitive understanding of the knowledge base and what should be entailed. We will be starting on simple sample knowledge bases and move onto more complex knowledge bases if there is time. If we move onto more complex knowledge bases, there creates the problem of how to find or create more complex sample knowledge bases.

2.2 Project Significance

Right now, PTL is a relatively new logic to be suggested to model real world situations. It is an attempt to solve some of the previous problems with Propositional Logic in modeling the real world problems. Real world problems create knowledge bases that are incredibly complex and it becomes impractical to manually produce entailment or reasoning for these complex knowledge bases. Therefore, creating implementations of the algorithms for entailment for these complex knowledge bases is incredibly important. And even more so in terms of creating an implementation for the two algorithms of entailment for PTL, as these real world situations are what PTL was designed for.

Our project is an attempt to make us a step closer, if not the completed step, towards an implementation for the two algorithms of entailment for PTL.

2.3 Possible Issues and Difficulties

(1) Theory Section
  1 Lack of Understanding
  2 Difficult to estimate time to Prove Satisfiability of Entailment for PTL
(2) Implementation Section
  1 Difficult to estimate time to implement algorithm
  2 Difficulty with implementing algorithm
  3 Difficulty with compatibility of SAT-Solvers
(3) Experimental Section
  1 Difficult to find sample knowledge bases
  2 Difficult to create sample knowledge bases

3 PROBLEM STATEMENT

3.1 Aims

The aims of this project are:

- to show whether the forms of entailment for PTL proposed are appropriate
- to implement the algorithms from the high-level descriptions of entailment for PTL
- to conduct experimental evaluations of the implementations

3.2 Research Questions

PTR can be divided into three mains categories of Research Questions. The first category of questions are the theoretical component of the honours project, the second category of questions are the implementation component of the honours project, with the third category being the experimental component of the honours project.

The research questions follow below:

(1) Are there algorithms that solve the problem of entailment for PTL?
  1 Does the LM-Entailment algorithm satisfy the conditions for and solve the problem of entailment for PTL?
  2 Does the PT-Entailment algorithm satisfy the conditions for and solve the problem of entailment for PTL?
(2) Can the algorithms that solve the problem of entailment for PTL be implemented?
  1 Can an implementation be made that satisfies the high-level description of LM-Entailment for PTL?
  2 Can an implementation be made that satisfies the high-level description of PT-Entailment for PTL?
(3) Do the implementations of the algorithms successfully entail in PTL for sample knowledge bases?
  1 Does the implementation made for LM-Entailment successfully entail in PTL for sample knowledge bases?
  2 Does the implementation made for PT-Entailment successfully entail in PTL for sample knowledge bases?

4 APPROACH

The project has been divided into three different sections as outlined in the Problem Statement and Project Description sections. The three sections each have their own approach. The three sections are the theoretical component, implementation and experimental work. Each of these sections are looking to answer each of the categories of research questions stated earlier respectively.

4.1 Theoretical Section

The theoretical component of the project focuses on proving LM-Entailment and PT-Entailment (suggested in Booth et al. [3]) for PTL are appropriate and satisfy the conditions for entailment in PTL. We will also be needing to prove that the algorithms produce what intuitively makes sense in terms of the database.

This section mostly involves just reproducing the theoretical work shown in Booth et al [5] and Booth et al.[3], with the possibility of extending the work if needed. We will be mathematically be showing that each algorithm satisfies a unique subset of postulates shown in the papers above. We will also be investigating whether entailment in PTL can be reduced to classical entailment checks. For this section we will be working closely with our supervisor, Tommie Meyer, as he is a co-author of these papers and is an expert in the field.

With our supervisor's help and the three papers ([5],[3], [4]) on this topic, we believe this will be enough resources for us to complete this section of the project.

4.2 Implementation Section

The implementation component of the project focuses on implementing the same two algorithms mentioned in the Theoretical section as a standalone reasoner. This is an extension on the work that has been done on entailment in PTL and will be a new contribution to the field if successful. This is, therefore, we feel, the most important section of the project and that is shown in the Gantt Chart (Figure 1) as we have budgeted the most time for this section. We will be basing our implementations on the high-level description of each of the algorithms shown in Booth et al [**?**]. We will be working closely with our supervisor for this section as well, as he is an expert in this field.

The algorithms produce a ranked structure from which the normal Boolean Satisfiability Problem applies with a few conditions. We will be testing different SAT-solvers as well to attempt to solve this problem and will attempt to select the most convenient and relevant SAT-solver to our problem. A number of different SAT-Solvers exist for this purpose [8] [13] [2] [12]. SAT-Solvers are implementations of algorithms for solving the Boolean Satisfiability problem. These determine if a given formula is satisfiable. Hence, using a SAT-Solver necessitates breaking down the algorithms into a series of these kinds of checks. We will be coding in Python. The software and language we use may change depending on if we deem another software or language to be more appropriate for our problem.

We will need to spend some time accustoming ourselves to SAT-solvers and different software used for reasoning for different logics as we are both relatively new to the topic of reasoning for logics.

4.3 Experimental Section

In this section we will be testing the LM-Entailment and PT-Entailment implementations with sample knowledge bases. This would be to test if the implementations are producing answers that correspond with both the intuitive interpretation of the knowledge base and the answers that the high-level description of the algorithms produce. If the answers do correspond, the test would be considered successful. We would be doing this section concurrently with the implementation Section. This section is referenced as the Testing section in the Gantt chart.

We will first be testing the implementations with simple knowledge bases to test if the implementations work. The most obvious example of a simple knowledge base test would be the *Tweety* example. For this part the experimental section we will more than likely not need much help and would more need help with implementation side.

If we finish the implementation and it works with the simple knowledge base examples, we will move onto testing more complex knowledge base examples on the implementation. For this part, we would have to consult with our supervisor to determine what methods we will use to either create or acquire more complex knowledge bases.

5 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

5.1 Ethical Issues

According to the application for ethics committee, research that does not involve human subjects does not require an application for ethics committee. With that in mind, this project does not have any foreseeable ethics issues.

5.2 Professional Issues

There are no foreseeable Professional issues for this project either. This project is an extension of previous work and can be extended on itself in future years of study.

5.3 Legal Issues

There are also no foreseeable legal issues with this Project. Any software used to create this project will be open source. The final product of this project will belong to the University of Cape Town, Guy Green and Andrew Howe-Ely.

6 RELATED WORK

Much of the related work to this project is in the field of defeasible reasoning, as the work done on this provides the basis for PTL. Kraus, Lehmann and Magidor [9] first introduced the defeasible $\mid\sim$ operator. Their approach to defeasible reasoning is called the KLM-approach. This work investigates the non-monotonic properties of the defeasibility. An important concept from this work is that of ranked interpretations; these are ordered structures introduced in Lehmann and Magidor [11] which allow reasoning with the defeasible operator. A ranked interpretation is an ordering of different interpretations. These are ranked in order of their typicality. Lehmann and Magidor [11] showed that there exists a minimal ranked model which represents the most typical view of the world. The concept of ranked interpretations have been adapted slightly since then with the papers: Booth and Paris (1998) [6] and Giordano et al (2012) [7].

The Rational Closure construction and algorithm was presented by Lehmann and Magidor [11]. This is the main approach to entailment in defeasible reasoning. The construction creates the minimal ranked model from which conclusions can be drawn from a knowledge base. The algorithm on the other hand creates a ranking of defeasible sentences in the knowledge base and determines whether a given sentence is entailed by that knowledge base.

Booth et al. [5] first proposed Propositional Typicality Logic. The introduction of the typicality operator ($\bullet$) to propositional logic makes the language more expressive, with similar semantics to defeasible reasoning. The $\bullet$ operator can be placed anywhere in a sentence, making it more expressive that the previous definition of defeasibility. The authors showed that the same form of entailment for defeasible reasoning is not appropriate for PTL. As a result of this new forms of entailment for PTL were investigated. Two forms of entailment were found as a possible solution to entailment in PTL. This result is seen as a sign that the language of PTL allows for more than one form of entailment because of its expressivity. These two forms of entailment, namely PT-Entailment and LM-Entailment, have advantages and disadvantages. In Booth et al [3], they put forward ten postulates that entailment for PTL could satisfy, however it was shown that they cannot all be satisfied at once. This impossibility result is what gives rise to the possibility of more than one entailment algorithm. The two forms of entailment each satisfy most but not all of these properties. The formal definitions and algorithms for computing entailment in these two forms are provided by Booth et al. [3]. Both are extensions of the rational closure concept and minimal model.

PT-Entailment is based on a version of minimality derived from the characterisation of rational closure found in Giordano [7]. The idea of this form of entailment is to respect the presumption of typicality, from Lehmann [10]. This informally means we should assume every situation is as typical as possible. A new pre-order, $\trianglelefteq_{\mathrm{PT}}$ is defined. To define this the authors Booth et al. first define a height function over a ranked interpretation. The height of a valuation in an interpretation corresponds to the number of the layer it is in, or to ∞. A lower height corresponds to a more typical valuation. A number of minimal models can be given back with the PT pre-order notion of minimality. The definition of PT-Entailment is that a formula α is PT-entailed by a knowledge base $\mathcal{K}$ if and only if the PT-minimum of the models of $\mathcal{K}$ is a subset of the models of α.

LM-Entailment is based on the minimal model concept. The idea is to still make a Minimal Ranked Interpretation model ($\trianglelefteq_{\mathrm{LM}}$). There is just a slight adaption to account for the properties of PTL. The normal process for finding a mimimal model, and you will reach a point where either there are no interpretations left or you will reach a fixed point. The PTL properties are what allow for the option of reaching a fixed point. The idea with LM-entailment is to disregard the interpretations left when you reach the fixed point. You will then have a normal minimal model of which normal reasoning can be done. This algorithm does not satisfy the Strict entailment postulate but does satisfy all the the others. LM-Entailment is a little more restrictive on what can be entailed that PT-Entailment.

7 ANTICIPATED OUTCOMES

The three sections each have their own anticipated outcomes and measures of what will be considered successful in terms of the project. The experimental section and the implementation can be considered to be linked in terms of their anticipated outcomes and the measure of being successful but for this case we shall separate them to make them more clear.

The anticipated outcome for the Theoretical section is that we will be able reproduce the results of the previous papers without any extension and show that PT-Entailment and LM-Entailment satisfy the concept of entailment in PTL and they are appropriate. We expect to be able to show that the algorithms produce answers that correspond to the natural intuition of the knowledge bases. While it is possible for us to extend the theory known in the paper, this is not expected and this section will be considered a success even without any extension to the papers.

The anticipated outcome for the implementation section is that we will be able to produce an implementation of PT-Entailment and LM-Entailment that succeeds in producing the results that satisfy entailment for PTL and produce the same answers as the high-level descriptions of the respective algorithms for the simple knowledge bases. The results will also need to intuitively make sense relative to the knowledge base. This section will be considered a success if that is achieved.

The anticipated outcome for the experimental section is that we will have successfully tested some simple knowledge bases on the PT-Entailment and LM-Entailment implementation, with the *Tweety* example being a compulsory knowledge base. This section will be considered a success if that is achieved. An extension on this section is being able to test the two implementations successfully with more complex knowledge bases, but this is not needed for the section to be considered a success.

The project will be considered a success if all three sections are considered a success.

8 PROJECT PLAN

8.1 Timeline

The timeline for this project runs from 23 March 2018 to 3 October 2018. On 23 March 2018, we were allocated the project and on 3 October 2018 the Reflection Paper is due. For the rest of the Milestones and Deliverables refer to the Milestones and Deliverables section.

The project timeline is divided into three phases. The three phases corresponds to three sections defined earlier in the proposal. The first phase corresponds to the theoretical section of the project. We will be working on answering the first research question during this time. The second phase corresponds to the implementation section of the project. We will be working on answering the second research question during this time. The third phase corresponds to the experimental section of the project. We will be working on answering the third research question during this time. For more information, refer to the Gantt Chart in Appendix A.

Description	Due Date	Deliverable
Literature Review	04/05/2018	Yes
Project Proposal	22/05/2018	Yes
Presentations of Project Proposals	29/05/2018	Yes
Prove Algorithms satisfy	15/06/2018	No
Finish Implementations	24/08/2018	No
Mark Allocation Decision	23/08/2018	Yes
Draft due	27/08/2018	Yes
Final Paper Submission	06/09/2018	Yes
Final Code Submission	07/09/2018	Yes
Final Project Demonstration	17/09/2018	Yes
Poster Completed	19/09/2018	Yes
Web Page	26/09/2018	Yes
Reflection Paper	03/10/2018	Yes

Table 1: Project Milestones

8.2 Milestones & Deliverables

The milestones and deliverables are shown in Table 1 and in the Gantt chart. The milestones of the project consist of predetermined deadlines and deliverables as well as milestones set relative to our project. The main milestones of the project is the implementation of the algorithms for entailment for PTL and the proving that the algorithms are appropriate and satisfies the conditions for entailment for PTL.

8.3 Resources

The resources we will require are:

- Access to Literature on the topic
- Python
- Open Source SAT-Solver software compatible to Python and to our problem
- Computers to run the code on

We will also require our supervisor's help with understanding the three papers on PTL.

8.4 Risks

The risks identified for this project are shown in Table 2. A risk matrix for each of these risks are shown in Table 3. The scale for both the columns, "Impact" and "Probability", are rated on scale from 1 to 10. 10 being the highest impact or probability and 1 being the lowest.

8.5 Work Allocation

The work for this project has been separated into two independent projects. Andrew Howe-Ely will be focusing his project on PT-Entailment for PTL. Guy Green will be focusing on LM-entailment for PTL. Each of us will be answering each of the three research questions relevant to our respective algorithms. So each of our

ID	Risk
1	Supervisor Unavailable
2	Scope Creep
3	Lack of Knowledge to Complete Sections
4	Conflict in Group
5	Poor Time Management
6	Poor Communication
7	Partner dropping out

Table 2: Risks

projects will have a theoretical, implementation and experimental section.

These two independent projects are separate from each other and refer to two different algorithms of entailment for PTL. The two algorithms do have common background theory that we will be learning together but each of three sections of the project will be done separately once we have learned the necessary background information. This does not mean, however, that we will not be able to help each other during the project, as both algorithms do refer to entailment for PTL.

9 APPENDIX A

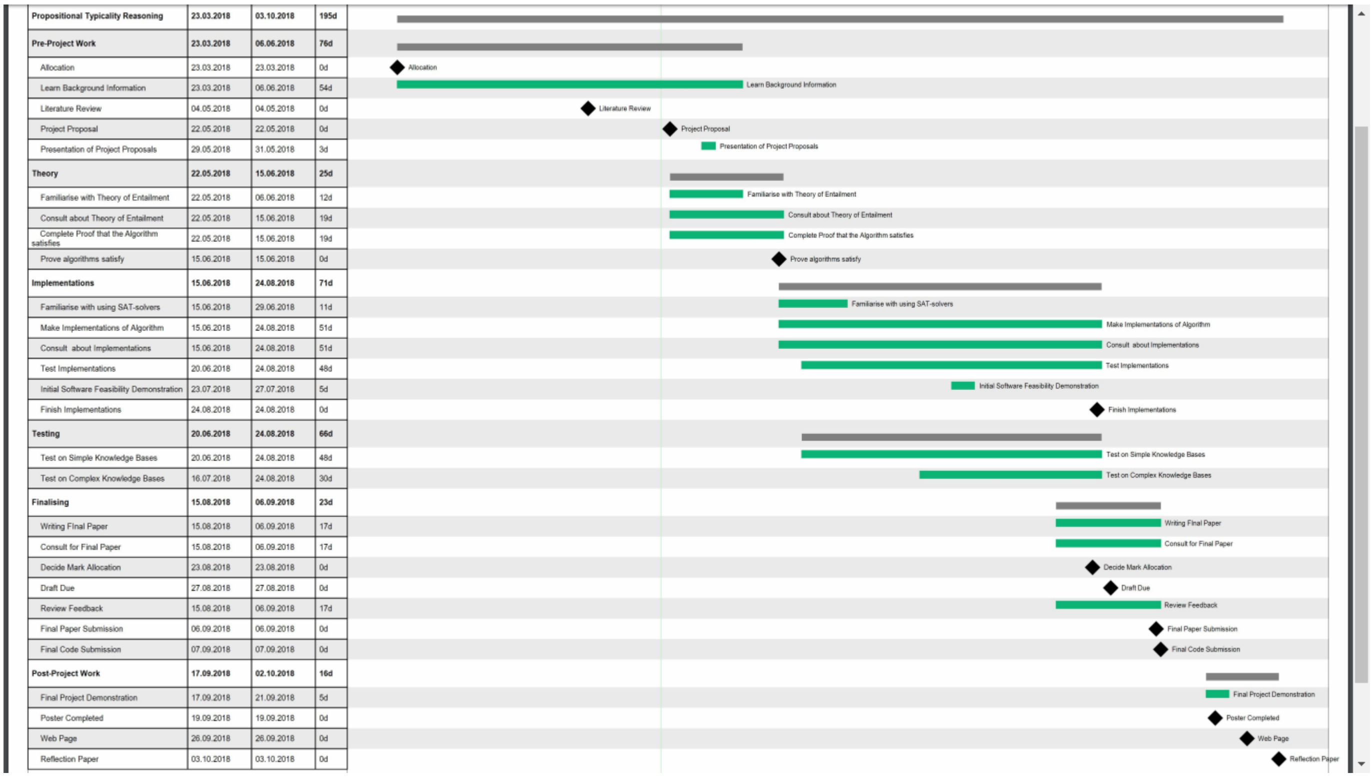

Figure 1: Gantt Chart

10 APPENDIX B

ID	Probability	Impact	Consequence	Mitigation	Monitoring	Management
1	4	6	Increased Possibility in struggling with the scope of the project and all components of the project (especially the theoretical part)	Organise meetings far in advance and make sure meetings are as productive as possible	Keep regularly in contact with supervisor to be aware of possible problems ahead	Send regular emails to keep in contact and to show progress
2	5	9	Increased Possibility of not completing project and possibility of mental health issues in participants	Keep in contact with Supervisor and partner and report back to make sure project is on target and not out of scope	Ensure that all deadlines and milestones are being met and that there is adequate time to complete the rest of the project	Focus on the most important parts of the project and if the project gets out of scope, remove unnecessary parts
3	5	10	High Probability of not being able to complete the project	Organise meetings with either supervisor or partner to make sure we are understanding the theory for the different components of project	Have regular meetings with partner or supervisor monitoring progress	Consult Project Supervisor on way forward
4	2	5	Very uncomfortable work environment. Projects are separate so has very little affect on the project work itself.	Have meetings to make sure that communication is open and have rules on what is appropriate	Being open and honest with partner	Do possible team building activities
5	5	9	Increased Possibility of not completing project and possibility of mental health issues in participants	Enforce strict deadlines and keep up with project	Have regular meetings with partner or supervisor monitoring progress	Raise issue with either supervisor or partner
6	2	5	Increased Possibility one of the partners falls behind	Have regular meetings to make sure that communication is open	Being open and honest with partner. Monitor each other's progress	Keep in contact through either email or WhatsApp and have meetings
7	2	5	Projects are relatively separate, so small affect on the work of the project but can have a mental affect on other partner	Ensure communication between partners. Follow-up on Milestones	Monitor progress and health of Partner. Keep in contact with Partner	Consult Project Supervisor on way forward

Table 3: Risk Matrix

REFERENCES

[1] Mordechai Ben-Ari. 2012. *Mathematical logic for computer science* (3 ed.). Springer Science & Business Media.

[2] Armin Biere. 2010. Lingeling, plingeling, picosat and precosat at sat race 2010. *FMV Report Series Technical Report* 10, 1 (2010).

[3] Richard Booth, Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2015. On the Entailment Problem for a Logic of Typicality. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2805–2811.

[4] Richard Booth, Giovanni Casini, Thomas Meyer, and Ivan Varzinczak. 2015. What Does Entailment for PTL Mean?. In *Proceedings of the 12th International Symposium on Logical Formalizations of Commonsense Reasoning*.

[5] Richard Booth, Thomas Meyer, and Ivan Varzinczak. 2012. PTL: A Propositional Typicality Logic. In *Proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA) (LNCS)*, L. Fariñas del Cerro, A. Herzig, and J. Mengin (Eds.). Springer, 107–119.

[6] Richard Booth and Jeff B Paris. 1998. A Note on the Rational Closure of Knowledge Bases with Both Positive and Negative Knowledge. *Journal of Logic, Language and Information* 7, 2 (1998), 165–190.

[7] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. 2012. A minimal model semantics for nonmonotonic reasoning. In *Logics in Artificial Intelligence*. Springer, 228–241.

[8] Weiwei Gong and Xu Zhou. 2017. A survey of SAT solver. *AIP Conference Proceedings* 1836, 1 (2017), 020059. https://doi.org/10.1063/1.4981999 arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.4981999

[9] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44 (1990), 167–207.

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

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

[12] Bart Selman and Henry Kautz. 1993. Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *IJCAI*, Vol. 93. Citeseer, 290–295.

[13] Niklas Sorensson and Niklas Een. 2005. Minisat v1. 13-a sat solver with conflict-clause minimization. *SAT* 2005, 53 (2005), 1–2.