

What are the Necessity Rules in Defeasible Reasoning?

Ho-Pun Lam^{1,2} and Guido Governatori²

¹ School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia

² NICTA*, Queensland Research Laboratory, Brisbane, Australia

Abstract. This paper investigates a new approach for computing the inference of defeasible logic. The algorithm proposed can substantially reduce the theory size increase due to transformations while preserving the representation properties in different variants of DL. Experiments also show that our algorithm outperforms traditional approach by several orders of magnitude.

1 Introduction

Defeasible reasoning [1] is a simple rule-based skeptical approach. This approach offers two main advantages over other mainstream nonmonotonic reasoning formalisms: (i) low computation complexity (linear time w.r.t. the size of a theory) [2] and (ii) its built-in preference handling facilities allowing one to derive plausible conclusions from incomplete and contradictory information in a natural and declarative way.

The strategy to compute the extension of a defeasible theory is to apply a series of preprocessing transformations that transform a defeasible theory into an equivalent theory without superiority relations and defeaters and then apply the reasoning algorithm [2] to the transformed theory. For instance, consider the example below:

Example 1. Let us consider the defeasible theory:

$$\begin{array}{lll} r_1 : \Rightarrow a & r'_1 : a \Rightarrow c & r_1 > r_2 \\ r_2 : \Rightarrow \neg a & r'_2 : \Rightarrow \neg c & \end{array}$$

The transformation of the above theory is:

$$\begin{array}{lll} r_1.a : \Rightarrow \neg inf^+(r_1) & r'_1.a : a \Rightarrow \neg inf^+(r'_1) & \\ r_1.c : \neg inf^+(r_1) \Rightarrow a & r'_1.c : \neg inf^+(r'_1) \Rightarrow c & \text{and } s_1^+ : \neg inf^+(r_2) \Rightarrow inf^+(r_1) \\ r_2.a : \Rightarrow \neg inf^+(r_2) & r'_2.a : \Rightarrow \neg inf^+(r'_2) & s_1^- : \neg inf^+(r_2) \Rightarrow inf^-(r_1) \\ r_2.c : \neg inf^+(r_2) \Rightarrow \neg a & r'_2.c : \neg inf^+(r'_2) \Rightarrow \neg c & \end{array}$$

It is clear that these transformations were designed to provide incremental transformations to the theory, and systematically introduce new literals and rules to emulate the features removed [3]. However, as pointed out in [4], such transformations are profligate in their introduction of propositions and generation of rules, which would result in an increase in theory size by at most a factor of 12.

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

In addition, such transformations cannot preserve the representation properties of DL in different variants³. For instance, consider the theory as shown in Example 1. Under ambiguity propagation variant the conclusions derived should be $+\partial\neg a$, $-\partial a$, $+\partial\neg c$, $-\partial c$. However, in the transformed theory, as the superiority relation is removed, the support of a in $r_1.c$ cannot be blocked, which subsequently propagated and supported the conclusions of $r'_1.a$ and $r'_1.c$. Hence the conclusions derived in the transformed theory becomes $+\partial\neg a$, $-\partial a$, $-\partial\neg c$, $-\partial c$, instead of the one desired.

Due to the deficiencies above and the inability of current reasoning algorithm to handle superiority relation directly, the focus of this paper is on finding the necessity rules in deriving the conclusions associated with superiority relations. Thus the aim is on finding conditions under which rules redundant under superiority relations and can be removed from the theory. We believe that our approach, in general, can also be applied to other rule-based nonmonotonic formalisms containing preference operator(s) that describes the relative strength of rules, such as preference logic.

2 Basics of Defeasible Logic

In this section we provide a short outline of DL and the construction of variants capturing different intuitions of non-monotonic reasoning based on modular and parametrized definition of the proof theory of the logic. For the full details, please refer to [3,6,7].

A defeasible theory D is a triple $(F, R, >)$ where F and R are finite set of facts and rules respectively, and $>$ is an acyclic superiority relation on R . Facts are logical statements describing indisputable facts, represented by (atomic) propositions (i.e. literals). A rule r describes the relations between a set of literals (the antecedent $A(r)$, which can be empty) and a literal (the consequence $C(r)$). DL supports three kinds of rules: *strict rules* ($r : A(r) \rightarrow C(r)$), *defeasible rules* ($r : A(r) \Rightarrow C(r)$) and *defeaters* ($r : A(r) \rightsquigarrow C(r)$). *Strict rules* are rules in classical sense, the conclusion follows every time the antecedents hold; a *defeasible rule* is allowed to assert its conclusion in case there is no contrary evidence to it. *Defeaters* cannot support conclusions but can provide contrary evidence to them. The *superiority relation* describes the relative strength of rules, and is used to obtain a conclusion where there are applicable conflicting rules.

DL is able to distinguish positive conclusions from negative conclusions, that is literals that can be proved and literals that are refuted. In addition, it is able to determine the strength of a conclusion, i.e., whether something is concluded using only strict rules and facts or whether we have a defeasible conclusion, a conclusion can be retracted if more evidence is provided. According, for a literal p we have the following four types of conclusions, called tagged literals: $+\Delta p$ (p is definitely provable), $-\Delta p$ (p is definitely refuted), $+\partial p$ (p is defeasible provable), and $-\partial p$ (p is defeasible refuted).

At the heart of DL we have its proof theory that tells us how to derive tagged literals. A *proof* is a sequence of tagged literals obeying proof conditions corresponding to inference rules. The inference rules establish when we can add a literal at the end of a sequence of tagged literals based on conditions on the elements of a theory and the

³ Several variants (such as *ambiguity propagation (AP)*, *well-founded semantics (WF)*) of DL have been proposed to capture the intuitions of different non-monotonic reasoning formalism. Readers interested please refer to [5] for details.

previous tagged literals in the sequence. The structure of the proof conditions has an argumentation flavour. For example, to prove $+∂p$:

- Phase 1: There is an applicable rule for p and
- Phase 2: For every rule for $¬p$ (the complement of p) either
 - Sub-Phase 1: the rule is discarded, or
 - Sub-Phase 2: the rule is defeated by a (stronger) rule for p

The notion of a rule being applicable means that all the antecedents of the rule are provable (with the appropriate strength); a rule is discarded if at least one of the antecedents is refuted (with the appropriate strength), and finally a rule is defeated, if there is a (stronger) rule for the complement of the conclusion that is applicable (again with the appropriate strength).

The above structure enables us to define several variants of DL [5] –such as ambiguity blocking and ambiguity propagation– by giving different parameters (i.e., this is what we mean ‘with the appropriate strength’ in the previous paragraph).

3 Inferiorly defeated rules

As mentioned before, the *superiority relation* in DL is used to define the preference, or relative strength, of rules, i.e., it provides information about which rules can overrule which other rules. And in some variants, such as ambiguity propagation, the superiority relation also provides information on whether the consequences of rules are supported or not. Based on these, we introduce the notion of *superiority chain*.

Definition 1. A superiority chain is a superiority relation hierarchy such that, apart from the first and last element of the chain, there exists a superiority relation between rules r_k and r_{k+1} :

$$r_1 > r_2 > \dots > r_n$$

where n is the length of the chain, and $C(r_k) = ¬C(r_{k+1})$, $\forall 1 \leq k < n$.

Notice that the superiority relation is not transitive, i.e., unless otherwise specified, there exists no superiority relation between a rule r and another rule in the chain. Consider the theory in example 2 below: r_1 and r_4 are in the same superiority chain but r_1 is not superior to r_4 , the consequence of r_1 cannot be used to overrule the consequence of r_4 .

Lemma 1. Let $D = (\emptyset, R, >)^4$ be a defeasible theory (in regular form) over a language \mathcal{L}_D . Then a rule $r \in R_{sd}[q]$ is inferiorly defeated if $\exists s \in R_{sd}[¬q] : A(s) = \emptyset$ and $s > r$.

The key observation of this lemma is that *irrespective to whether an inferiorly defeated rule r is derivable or not, its conclusion is going to be overruled by a superior rule s* . Consider again the theory in Example 1. Since r_1 and r_2 are both derivable and r_2 is superior than r_1 , r_1 is inferiorly defeated and its conclusion is overridden by r_2 . So, under this situation, r_1 is redundant and cannot be used to derive any positive conclusion. Removing it from the theory and falsifying its conclusion does not affect the conclusions derived. And the same applies even when $A(r_1) \neq \emptyset$.

However, the example above is just oversimplified. Consider the example below:

⁴ A defeasible theory $D = (F, R, >)$ can be transformed to an equivalent theory $D' = (\emptyset, R', >')$ (without fact) using the algorithm proposed in [3].

Example 2. Let us consider the defeasible theory (D):

$$r_1 := a \quad r_2 := \neg a \quad r_3 := a \quad r_4 := \neg a \quad \text{and } r_1 > r_2 > r_3 > r_4$$

All rules above are derivable but r_2 , r_3 and r_4 are inferiorly defeated and r_1 is the only rule that can be used to derive positive conclusion. So, the conclusions inferred should be $+\partial a, -\partial \neg a$. However, if we remove an inferiorly defeated rule arbitrarily, say r_3 , then the theory will become (D'):

$$r_1 := a \quad r_2 := \neg a \quad r_4 := \neg a \quad r_1 > r_2$$

Then, only r_2 is inferiorly defeated and the conclusions derived will become $-\partial a, -\partial \neg a$, implying that $D \neq D'$. Hence a rule is inferiorly defeated is not an adequate condition all by itself to enable it to be removed from the theory without changing the conclusions. If we take into account a line of superiority chain then additional conditions are needed.

4 Statement of Results

To recap, our main focus lies in characterizing a set inferiorly defeated rules that cannot be used to derive positive conclusions and can be removed from the theory without changing its conclusions.

Definition 2. Let $D = (\emptyset, R, >)$ be a defeasible theory (in regular form) over a language \mathcal{L}_D . Then, R_{inf} is the set of inferiorly defeated rules in D with number of weaker rules equal to zero. That is, $\forall r \in R_{inf}[q], \exists s \in R[\neg q]$ s.t. $r > s$ (i.e., r is the weakest rules along the line of superiority chain).

Theorem 1. Let $D = (\emptyset, R, >)$ be a defeasible theory (in regular form), and $r \in R_{inf}$. Let $D' = (\emptyset, R \setminus \{r\}, >')$ be the reduct of D , denoted by $\text{reduct}(D)$, where $>'$ is defined by the following condition:

$$\forall s \in R, A(s) = \emptyset, s > r \text{ (} >' \Leftrightarrow \setminus \{s > r\} \text{)}$$

Then $D \equiv D'$. In addition, $-\partial q$ can be derived if $R[q] = \emptyset$ after the removal of r .

This theorem looks simple but plays a fundamental role in differentiating the set of necessity rules that are required in the inference process from the set of rules that are redundant and can be removed from the theory. From the theorem, we can conclude that if an inferiorly defeated rule is the weakest rule of a superiority chain, then removing it from the theory will not cause any undesired effects to the inference process. For instance, consider the theory again in Example 2 and apply the theorem recursively, then, r_4 , r_3 and r_2 will be removed subsequently leaving r_1 as the only rule in the superiority chain, which thus give us the results: $+\partial a$ and $\partial \neg a$, as desired.

Therefore, by incorporating the theorem into the inference process, the transformation of eliminating the superiority relation is no longer necessary, which thus reduced the transformation time as well as the number of propositions introduced and rules generated. In addition, upon satisfying the conditions stated above, an inferiorly defeated rule can be removed from the theory immediately even without knowing any information about the provability of its body, which can further enhance the performance of the reasoning engine. Furthermore, the result is general and it applies to the ambiguity blocking (AB), ambiguity propagating (AP) and well-founded (WF) variants of DL.

4.1 Implementation and some experimental results

The above theorem has been incorporated into the inference algorithm proposed in [2] (for AB) and [8] (for AP and WF) and is implemented in the latest version of SPINdle [9]. The implementation is tested using the UAV navigation theory described in [10] (consists of about a hundred rules and 40 superiority relations) and the superiority relation test theories generated by a tool that comes with Deimos [11]. The test has been measured on a Pentium 4 PC (3GHz) with Window XP and 2GB main memory. Table 1 and Figure 1 below show the performance result and their memory usage.

As shown, our approach outperforms the traditional approach under the same model complexity. Even with theory 1000 rules and 500 superiority relation, our approach can complete the whole inference process in less than half a second (289ms), while it takes 9 for the approach based on a transformation to remove the superiority relation. In terms of memory usage, our approach only increases linearly with the original theory size while the traditional approach a significantly larger memory size.

| Algorithm | Superiority relation removal (ms) | Conclusions generation (ms) | Total Reasoning Time (ms) | Memory Usage (MB) |
|-------------|-----------------------------------|-----------------------------|---------------------------|-------------------|
| Traditional | 47 | 156 | 203 | 1.52 |
| New | - | 93 | 93 | 1.36 |

Table 1: Reasoning time used in UAV navigation theory

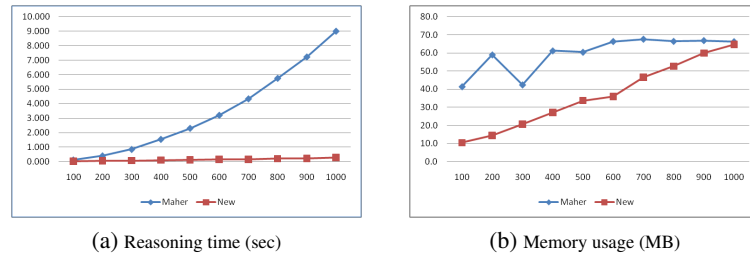


Fig. 1: Superiority relations test theories - Reasoning time and Memory usage (with different theories size)

5 Related works and Conclusions

A number of defeasible logic reasoners and systems have been proposed in recent years to cover different variants of defeasible reasoning and other intuitions of nonmonotonic reasoning. Most of the implementations rely on either transformations, or translation to other logic formalism (such as extended logic programming) [12,13] and carry out the inference process using the underlying reasoning engine. However, as mentioned

before, the representation properties of DL cannot be preserved after transformation. Also, most of the system are query based and do not compute directly the extension of a theory, which may leads to some counterintuitive result [14]. The meta-program approach is at the foundation of the approaches based on transformation in other formalism. DR-Prolog [12], which directly implement the meta-programs of [5], provides a Semantic Web enabled implementation of DL.

In this paper, we have presented a theorem that allow us to reason on a defeasible theory without removing the superiority relation. The essence of this method lies in the ability in identifying inferiorly defeated rules that are redundant and can be removed from the theory, which help in preserving the representation properties of defeasible logic across different variants and simplified the work in subsequent processes. Our result shows that the performance gains are significant. As a future work, we will work on studying the relation of conclusions between different variants, with special interest in ambiguity propagation and well-founded semantics.

References

1. Nute, D.: Defeasible logic. In Gabbay, D., Hogger, C., eds.: *Handbook of Logic for Artificial Intelligence and Logic Programming*. Volume III. Oxford University Press (1994) 353–395
2. Maher, M.J.: Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming* **1**(6) (2001) 691–711
3. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. *ACM Transactions on Computational Logic* **2**(2) (2001) 255–286
4. Bryant, D., Krause, P.: A review of current defeasible reasoning implementations. *Knowl. Eng. Rev.* **23**(3) (2008) 227–260
5. Antoniou, G., Billington, D., Governatori, G., Maher, M., Rock, A.: A family of defeasible reasoning logics and its implementation. In: *Proc. ECAI-2000*. (2000) 459–463
6. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: A flexible framework for defeasible logics. In: *AAAI-2000*, AAAI/MIT Press (2000) 401–405
7. Billington, D., Antoniou, G., Governatori, G., Maher, M.J.: An inclusion theorem for defeasible logic. *ACM Transactions in Computational Logic* **12**(1) (2010)
8. Lam, H.P., Governatori, G.: On the problem of computing ambiguity propagation and well-founded semantics in defeasible logic. In Rotolo, A., Hall, J., Dean, M., Tabet, S., eds.: *Proc. RuleML-2010*. Number 6403 in LNCS, Berlin, Springer (2010) 119–127
9. Lam, H.P., Governatori, G.: The making of SPINdle. In Paschke, A., Governatori, G., Hall, J., eds.: *Proc. RuleML 2009*. Number 5858 in LNCS, Berlin, Springer (2009) 315–322
10. Lam, H.P., Thakur, S., Governatori, G., Sattar, A.: A model to coordinate uavs in urban environment using defeasible logic. In Hu, Y.J., Yeh, C.L., Laun, W., Governatori, G., Hall, J., Paschke, A., eds.: *Proc. RuleML-2009 Challenge*. Number 549 in CEUR Workshop Proceedings (2009)
11. Maher, M.J., Rock, A., Antoniou, G., Billington, D., Miller, T.: Efficient defeasible reasoning systems. *International Journal on Artificial Intelligence Tools* **10**(4) (2001) 483–501
12. Antoniou, G., Bikakis, A.: DR-Prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Trans. Knowl. Data Eng.* **19**(2) (2007) 233–245
13. Madalińska-Bugaj, E., Lukaszewicz, W.: Formalizing defeasible logic in cake. *Fundam. Inf.* **57**(2-4) (2003) 193–213
14. Antoniou, G.: A discussion of some intuitions of defeasible reasoning. In: *Methods and Applications of Artificial Intelligence*. Volume 3025 of LNCS., Berlin, Springer (2004) 311–320