Extending Defeasible Logic and Defeasible Prolog

Ahti-Veikko Pietarinen[1]

Abstract. Defeasible logic (DL) promotes enthymemic, argumentative reasoning on incomplete set of premisses retracted on the presence of contrary information. Defeasible Prolog (d-Prolog) is a Prolog metainterpreter to implement DL. We give proof conditions for the *even-if* clauses of DL with the pre-emption of defeaters to prevent rules from rebutting more specific rules, implemented in d-Prolog.

1 INTRODUCTION

The goal of *defeasible logic* (DL) [4, 5, 6] is to formalise nonmonotonic inferences ('typically, φ's are ψ', 'reasonable grounds for holding φ warrant reasonable grounds for holding ψ'). Such inferences hold only if a defeasible theory contains no rules representing contrary information. *Defeasible Prolog* (d-Prolog) is a Prolog metainterpreter [6] to implement defeasible inferences. There are *defeasible implications* ($\psi \texttt{ := } \Phi$, 'typically, Φ's are ψ') and *defeaters* ($\psi \texttt{ :\^{} } \Phi$, 'if Φ, it might be that ψ'). New two-place operators "`:=`" and "`:^`" are added to Prolog. A defeater does not support conclusions, as its purpose is to interfere with the derivations from defeasible rules. Rule heads are literals and bodies are conjunctions of atomic sentences. Literals and so the rule heads may be negated (`neg`).

Defeasible theory $\mathcal{T}$ is $\langle K, R\rangle$ of a finite set of literals and rules. A proof of a literal ψ is defined as a proof tree t, ψ a root of t and t a finite labelled tree: for every node n of t, a $\mathcal{T}$ exists and some literal is flagged positive (φ^+) or negative (φ^-). Node n with φ^+ means that the literal is defeasibly derivable from $\mathcal{T}$ with respect to proof conditions Σ $(\mathcal{T} \vdash_\Sigma \psi)$. Node n with ψ^- indicates that the literal is demonstrably not derivable $\mathcal{T}$ by the conditions in Σ $(\mathcal{T} \not\vdash_\Sigma \psi)$.

DL is Σ on every n of t such that not ($\mathcal{T} \vdash_\Sigma \psi$ and $\mathcal{T} \not\vdash_\Sigma \psi$). To solve which defeasible rules defeat others, a partial order (superiority relation) is defined for defeasible rules. If r_1 is superior to r_2 $(r_1 \sqsupset r_2)$, the former defeats the latter.

We extract information about partial order by more specific antecedent. Defining superiority as antecedent specificity, antecedents of one rule are derivable from antecedents of another rule. However, some derivations may be based on just a subtheory of $\mathcal{T}$. Thus, labellings may vary on their admissible set of literals and rules upon which the proofs of literals in rules defined by more specific antecedents depend.

Rule $\psi \texttt{ := } \Phi$ is *defeated* by $\eta \texttt{ := } X$ or $\eta \texttt{ :- } X$, if η is either a negation of ψ or some literal that is explicitly incompatible with ψ (η is *contrary to* ψ). Rule $\psi \texttt{ := } \Phi$ is *undercut* by $\eta \texttt{ :\^{}} X$, if η is contrary to ψ and $\psi \texttt{ := } \Phi$ is not superior to $\eta \texttt{ :\^{}} X$. $\psi \texttt{ := } \Phi$ or $\psi \texttt{ :\^{}} \Phi$ is *pre-empted,* whenever (i) literal η is contrary to ψ derivable from the theory or ordinary Prolog rule $\eta \texttt{ :- } X$ with contrary head exists, or (ii) for $\psi \texttt{ := } \Phi$ there is a rule $\eta \texttt{ := } X$ whose head η is contrary to ψ and which is superior, e.g. if $(\eta \texttt{ := } X) \sqsupset (\psi \texttt{ := } \Phi)$.

[1] Department of Philosophy, University of Helsinki, Finland, email: pietarin@cc.helsinki.fi. Supported by the Academy of Finland (1103130).

Literals in X must be defeasibly derivable from $\mathcal{T}$. *Preemption* relaxes that defeasible rules to defeat competing contrary rules are always superior to every other competing rule. When pre-emption is on, once defeasible rule is defeated it loses its capacity to defeat others. We extend d-Prolog with *even-if* rules and *even-if* proof conditions enabling pre-emption.

2 RULES FOR THE 'EVEN-IF' CONDITION

Defeasible even-if rule is $\psi \texttt{ := } \Xi \texttt{ | } \Phi$ ('if Φ, then (typically) ψ even if Ξ holds'), permitting superiority by more specific antecedent:

Example 2.1 *A beneficiary is suspect:*

`suspect(X) := beneficiary(X).`

With an alibi a person usually is not suspect even if he is a beneficiary:

`neg suspect(X) := beneficiary(X) | alibi(X).`

Tom has an alibi: `alibi(tom)`. *He must not be a suspect. In both cases it can be concluded tentatively that* `neg suspect(tom)` *is a derivation from* $\mathcal{T}$. *Let us then add the defeater, according to which a person with an alibi who has not provided reliable documents may well be suspect:*

`(suspect(X) :^ (alibi(X), neg rel_doc(X)).`

If Tom has not provided reliable documents (`neg rel_doc(tom)`*), it is difficult to conclude, with this amount of information, whether Tom is suspect or not. In extended d-Prolog (App.), both* `neg suspect(tom)` *and* `suspect(tom)` *are demonstrably not derivable from the* $\mathcal{T}$*, since the rule*

`suspect(X) :^ (alibi(X), neg rel_doc(X))`

is not an acceptable rule to pre-empt an inferior rule

`neg suspect(X) := beneficiary(X) | alibi(X),`

because pre-emption is possible only if a rule is defeated with a superior rule.

Let n be arbitrary in t, and let $@\psi$ denote when ψ is a tentative conclusion derived from $\mathcal{T}$. Conditions P^+ and P^- hold for all n, the former for literals defeasibly derivable and the latter for literals demonstrably not derivable:

P^+**:** Node n is labelled $\langle K, R, @\psi^+\rangle$ if either n has child $\langle K, R, \texttt{neg } \psi^-\rangle$ or there exists $\psi \texttt{ := } \Xi \texttt{ | } \Phi \in R$ such that 1, 2 and 3 hold:

1. for every $\varphi \in \Phi$, n has child $\langle K, R, @\varphi^+\rangle$ (=every literal in main body is derivable)
2. for every `neg` $\psi \texttt{ := } H \in R$ there exists $\eta \in H$ and child of n $\langle K, R, @\eta^-\rangle$ (=every contrary rule must have nonderivable literal in the body)
3. for every `neg` $\psi \texttt{ := } \Delta \texttt{ | } \Theta \in R$ or `neg` $\psi \texttt{ :\^{} } \Theta \in R$ either a or b:

(a) there exists $\theta \in \Theta$ and child of n $\langle K, R, @\theta^-\rangle$ (=every contrary rule must have nonderivable literal in main body; subsumes clause 2.)

(b) i, ii and iii hold:

- i there exists $\varphi \in \Phi \cup \Xi$ and child of n $\langle \Theta, R, @\varphi^-\rangle$ (=specificity: some literal in antecedents not derivable from antecedents of contraries)
- ii for every $\theta \in \Theta$ there exists child of n $\langle (\Phi \cup \Xi), R, @\theta^+\rangle$ (=specificity: every literal in main condition of contrary is derivable from antecedents of main rule)
- iii for every $\varphi \in \Phi \cup \Xi$ there exists child of n $\langle K, R, @\varphi^+\rangle$ (=pre-emption: every literal in body of main rule is defeasibly derivable).

P^-**:** Node n is labelled $\langle K, R, @\psi^- \rangle$, if either n has child $\langle K, R, @\psi^+ \rangle$ or both 1 and 2:

1. for every ψ `:=` $\Phi \in R$, either a or b holds:

(a) there exists $\varphi \in \Phi$ and child of n $\langle K, R, @\varphi^- \rangle$ (=literal in the body demonstrably not derivable)

(b) there exists ψ `:=` $H \in R$ such that for every $\eta \in H$, n has child $\langle K, R, @\eta^+ \rangle$ (=some other rule with same head has derivable body)

2. for every ψ `:=` Ξ `|` $\Phi \in R$, either a, b or c:

(a) there exists $\varphi \in \Phi$ and child of n $\langle K, R, @\varphi^- \rangle$ (=literal in head demonstrably not derivable; subsumes clause 1.(a))

(b) there exists `neg` ψ `:=` $X \in R$ s.t. for every $\chi \in X$, n has child $\langle K, R, @\chi^+ \rangle$ (=some contrary rule has derivable body)

(c) there exists `neg` ψ `:=` Δ `|` $\Theta \in R$ or `neg` ψ `:^` $\Theta \in R$ s.t. for every $\theta \in \Theta$, n has child $\langle K, R, @\theta^+ \rangle$ and either i, ii or iii:

i there exists $\theta \in \Theta$ and child of n $\langle (\Phi \cup \Xi), R, @\theta^+ \rangle$ (=specificity: some literal in main body is derivable from body of original rule)

ii for every $\varphi \in \Phi \cup \Xi$, n has child $\langle \Xi, R, @\varphi^+ \rangle$ (=specificity: every literal in body of original rule is derivable from main body of contrary rule)

iii there exists $\varphi \in \Psi \cup \Xi$ and child of n $\langle K, R, @\theta^- \rangle$ (=pre-emption: some literal in body of original rule is demonstrably not derivable).

Let $\mathbf{M}$ be monotonic core of four basic conditions, and $\mathbf{SS}^+$ is a semi-strictness condition. Then ψ `:-` Φ may defeat `neg` ψ `:-` Ψ if the antecedent Ψ is only defeasibly derivable. On the depth of t it can be shown that no theory $\mathcal{T}$ and ψ exists such that $\mathcal{T} \vdash_{\mathbf{P}} \psi$ and $\mathcal{T} \not\vdash_{\mathbf{P}} \psi$. Thus $\mathbf{P} = \mathbf{M} \cup \{SS^+, P^+, P^-\}$ is a DL.

3 EXTENDING D-PROLOG

We add a two-place relation `|` (`#` in App.) to distinguish primary condition Φ in ψ `:=` Ξ `|` Φ and *even-if* condition Ξ as in P^+ and P^-. Three additions to defeat and undercut of [6] are needed.

1. ψ `:=` Ξ or ψ `:-` Ξ is defeated (`defeated/2`), if even-if rule φ `:=` Ξ `|` Φ exists whose head φ is contrary to ψ, Φ (without Ξ) is defeasible derivable, and Ξ is a body of r not superior to φ `:=` Ξ `|` Φ.
2. ψ `:=` Ξ `|` Φ is defeated (i) if φ `:=` Θ exists whose head φ is contrary to ψ, Θ is defeasibly derivable, and Ξ is not head of any defeasible rule not superior to φ `:=` Θ, or (ii) if φ `:=` Φ `|` Θ exists whose head φ is contrary to ψ, Θ is defeasibly derivable, and Φ is head of defeasible rule r not superior to φ `:=` Φ `|` Θ.
3. ψ `:=` Ξ `|` Θ is undercut (`undercut/2`) if a defeater φ `:^` Θ exists whose head is contrary to ψ, Θ is defeasibly derivable, and ψ `:=` Ξ `|` Θ is not superior to it.

Preemption (`preempted/2`) is similar to defeating (App.). Two new definitions for the defeasible derivability (`def_der/2`), three for defining superiority relation with defeasible specificity (`sup_rule/2`) and some others for syntax and occurrences of the even-if rules also exist (App.).

Enabling preemption of defeaters increases size of proof trees considerably. These are the sizes without/with the defeater of Ex.2.1:

pre-emption	`@ suspect(tom)`	`@ neg suspect(tom)`
+	23 / 23	90 / 138
–	53 / 53	98 / 270

To limit the complexity of defeasible reasoning one may (i) restrict the depth of t by generating paths only up to a certain limit, and after reaching the limit, proofs backtrack to earlier n; (ii) restrict branching factor of t by keeping the size of R and maximum number of conjunctions small. The former has a side effect: as argumentative inference uses depth-first search, constraining it may cause the horizon effect. Limiting maximal depth amounts to cheap loop checking. The latter reflects defeasible argumentation as only the most pertinent arguments should be used.[2]

[2] In conversations it is often necessary to lay size restrictions to the set of assertions. Usually one or two defeaters is enough to render opposite views unwarranted.

Derivation of incompatible clauses and prevention of cycles depends on programmer. To prevent cyclic dependencies, we may *stratificate* predicates to see which rules to be used to derive conclusion. Prior to referring to a negation of a fact, the fact itself needs to be defined: stratified program partitions its clauses into hierarchical sets in which negative goals are defined in lower-level predicates. If some programs cannot be stratified, in *well-founded semantics* some facts are undefined and in bivalent *stable model semantics* non-stratified programs may have several models or none. It is also possible to consider different grades of stratification in the sense that only certain kinds of rules, such as defeasibe rules, strict rules or defeaters, are subject to stratification, amounting to semistratified programs.[3]

Another development intergrates defeasible rules with *argumentative structures* and defeasibility among arguments [3] in legal, economic and decision-making systems [7]. *Answer-set programming* performs nonmonotonic reasoning [1, 2], including defeasible rules.

4 CONCLUSIONS

We have given proof conditions for the *even-if* clauses of DL with the pre-emption of defeaters to prevent rules from rebutting more specific rules. The extension has been implemented in d-Prolog.

REFERENCES

[1] Brewka, G., Eiter, T. 1999. Preferred answer sets for extended logic programs, *Artificial Intelligence* 109, 297-356.

[2] Brewka, G., Eiter, T. 2000. Prioritizing default logic, Hölldobler, S. ed. *Intellectics and Computational Logic*, Dordrecht: Kluwer, 27-45.

[3] García, A.J., Simari, G.R. 2004. Defeasible logic programming: an argumentative approach, *Theory and Practice of Log. Prog.* 4, 95-138.

[4] Nute, D. 1992. Basic defeasible logic, Fariñas del Cerro, L., Penttonen, M. eds *Intensional Logics for Logic Programming*, Oxford: OUP, 125-154.

[5] Nute, D. 1994. Defeasible logic, Gabbay, D. et al. eds *Handbook of Logic in AI and Logic Programming*, Oxford: Clarendon Press, 353-394.

[6] Nute, D. 1997. Defeasible Prolog, Convington, M. et al. *Prolog Programming in Depth*, Englewood Cliffs: Prentice Hall.

[7] Prakken, H., Wreeswijk, G. 2002. Logical systems for defeasible argumentation, Gabbay, D., Guenthner, F. eds *Handbook of Philosophical Logic*, Dordrecht: Kluwer, 219-318.

A EXTENSION OF D-PROLOG

```
init :- op(1100,fx,@), % defeasible conclusion
        op(900,fx,neg), % negation
        op(1100,xfy,:=), % defeasible implication
        op(1100,xfy,:^), % defeater
        op(1100,xfy,#). % even-if

:- dynamic((neg)/1, (:=)/2, (:^)/2, (#)/2).
:- multifile((neg)/1, (:=)/2, (:^)/2, (#)/2).

def_der(KB,Goal) :- preemption,
  def_rule(KB,(Goal := (ConditionX # Condition))),
  \+ (contrary(Goal,Contrary1), strict_der(KB,Contrary1)), def_der(KB,Condition),
   \+ (contrary(Goal,Contrary2),
       clause(Contrary2,Condition2), Condition2 \== true, def_der(KB,Condition2)),
   \+ (contrary(Goal,Contrary3),
       def_rule(KB,(Contrary3 := (Condition4 # Condition3))),
       def_der(KB,Condition3),
   \+ (preempted(KB,(Contrary3 := (Condition4 # Condition3))))),
   \+ (contrary(Goal,Contrary5), def_rule(KB,(Contrary5 := Condition5)),
       def_der(KB,Condition5),
    \+ (preempted(KB,(Contrary5 := Condition5)))),
    \+ (contrary(Goal,Contrary6), (Contrary6 :^ Condition6),
       def_der(KB,Condition6),
    \+ (preempted(KB,(Contrary6 :^ Condition6)))).

defeated(KB,(Head := (Body1 # Body))) :-
  contrary(Head,Contrary), def_rule(KB,(Contrary := (Condition2 # Condition))),
  def_der(KB,Condition),
  \+ sup_rule((Head := (Body1 # Body)), (Contrary := (Condition2 # Condition))),!.

undercut(KB,(Head := (Body1 # Body))) :-
  contrary(Head,Contrary), (Contrary :^ Condition), def_der(KB,Condition),
  \+ sup_rule((Head := (Body1 # Body)), (Contrary :^ Body)),!.

preempted(KB,(Head := (Body1 # Body))) :-
  contrary(Head,Contrary), def_rule(KB,(Contrary := (Condition2 # Condition))),
  def_der(KB,Condition),
  sup_rule((Contrary := (Condition2 # Condition)), (Head := (Body1 # Body))),!.
```

[3] As far as semantics is concerned, models of DL are *ceteris paribus* situations insensitive to vagueness. We believe that *prototype theory* concerning new information in pre-existing structures of knowledge by relaxing strict category membership is a good candidate for a defeasible model theory.