Relating Defeasible Logic and Well-Founded Semantics

Gerhard Brewka
University of Leipzig, Germany
brewka@informatik.uni-leipzig.de

1. Directly skeptical systems
2. Defeasible logic
3. Well-founded semantics with priorities
4. Results
5. Conclusions

1. Directly skeptical systems

- Standard nonmonotonic systems define theorems via acceptable belief sets (extensions, expansions, stable models, ...)
- Some systems have been defined which construct theorems without this deviation
- Usually based on some kind of least fixed point construction
- Highly skeptical (i.e., miss some desired conclusions)
- Advantage: polynomially computable (for propositional case)

2. Defeasible logic

- Developed by Nute in 1987
- Never as prominent as some other nonmonotonic logics
- Recent revival due to active group in Australia
- New theoretical investigations, more interesting new versions
- Nice computational properties, implementations
- Original version not ambiguity propagating
- Here: most interesting variant (Antoniou et al, ECAI 2000)

Ambiguity propagation

Defeasible logic based on strict rules $A \rightarrow p$ and defeasible rules $A \Rightarrow p$ where A is a set of literals and p a literal.

Conflicts among rules solved through preference relation $>$:
$r > r'$ stands for: r has higher priority than r'.

Example 1:

1) $\Rightarrow p$
2) $\Rightarrow \neg p$
3) $\Rightarrow q$
4) $p \Rightarrow \neg q$

Nute's logic concludes q although there is a conflicting undefeated argument!

Antoniou et al.'s Defeasible logic

- Defeasible theory $T = (R, >)$, R finite set of strict and defeasible rules, $>$ preference relation among R.
- Conclusion of T tagged literal:
  $+\Delta q$ means q is strictly provable,
  $+\delta q$ means q is defeasibly provable,
  $+\sigma q$ means q is supported.
  tags with minus-signs: corresponding negated expressions.
- Proof: finite sequence P of tagged literals.
- Notation:
  $P(i)$ i-th element of P, $P(1..i)$ initial segment of length i.
  $-q$ complement of q. $R[q]$ rules with head q, $R_s[q]$ subset of $R[q]$ consisting of all strict rules. $A(r)$ antecedents of a rule r.

Inference rules

$+\Delta$: If $P(i+1) = +\Delta q$ then
$\exists r \in R_s[q]\ \forall a \in A(r) : +\Delta a \in P(1..i)$.

$-\Delta$: If $P(i+1) = -\Delta q$ then
$\forall r \in R_s[q]\ \exists a \in A(r) : -\Delta a \in P(1..i)$.

$+\delta$: If $P(i+1) = +\delta q$ then

(1) $+\Delta q \in P(1..i)$ or

(2.1) $-\Delta - q \in P(1..i)$ and

(2.2) $\exists r \in R[q]$ such that

$\forall a \in A[r] : +\delta a \in P(1..i)$ and

$\forall s \in R[-q]$:

$[\exists a \in A[s] : -\sigma a \in P(1..i)]$ or $r > s$.

$-\delta$: If $P(i+1) = -\delta q$ then

(1) $-\Delta q \in P(1..i)$ and

(2.1) $+\Delta - q \in P(1..i)$ or

(2.2) $\forall r \in R[q]$:

$\exists a \in A[r] : -\delta a \in P(1..i)$ or

$\exists s \in R[-q]$ such that

$[\forall a \in A[s] : +\sigma a \in P(1..i)]$ and $r \not> s$.

$+\sigma$: If $P(i+1) = +\sigma q$ then

$+\Delta q \in P(1..i)$ or

$\exists r \in R[q]$ such that

$\forall a \in A[r] : +\sigma a \in P(1..i)$ and

$\forall s \in R[-q]$:

$\exists a \in A[s] : -\delta a \in P(1..i)$ or

$s \not> r$.

$-\sigma$: If $P(i+1) = -\sigma q$ then

$-\Delta q \in P(1..i)$ and

$\forall r \in R[q]$:

$\exists a \in A[r] : -\sigma a \in P(1..i)$ or

$\exists s \in R[-q]$ such that

$\forall a \in A[s] : +\delta a \in P(1..i)$ and

$s > r$.

3. Prioritized well-founded semantics

An extended logic program is a finite set P of rules of the form

$$c \leftarrow a_1, \ldots, a_n, not \;\; b_1, \ldots, not \;\; b_m$$

where the a_i, b_j and c are propositional literals.

A prioritzed logic program is a pair $(P, >)$ where P is an extended logic program and $>$ an acyclic preference relation on P.

A rule r of the form above is defeated by a literal l if $l = b_i$ for some $i \in \{1, \ldots, m\}$.

r is defeated by a set of literals X if X contains a literal that defeats r.

The Gelfond/Lifschitz operator

Closure $Cl(R)$: smallest set of literals closed under rules R.

Consequences $Cn(R)$: smallest set of literals (1) closed under R, and (2) logically closed, i.e., consistent or equal to the set of all literals.

Default negated literals in R are neglected!

Let P be a logic program, X a set of literals. The X-reduct of P, denoted P^X, is obtained from P by deleting rules defeated by X.

The Gelfond/Lifschitz operator:

$$\gamma_P(X) = Cn(P^X)$$

Assume X are literals known to be derivable. $\gamma_P(X)$ literals which still could potentially be derivable.

WFS and γ

For normal logic programs (no strong negation) the set of atoms true according to WFS is the least fixed point of γ^2.

For extended logic programs we slightly modify the fixed point operator: compute lfp of $\gamma\gamma^*$ where γ^* rather than yielding the consequences $Cn(P^X)$ yields the closure $Cl(P^X)$.

Yields larger set of well-founded conclusions without violating correctness wrt answer set semantics.

Argumentation theoretic reformulation

Definition 1. Let P be an extended logic program.

- l S-potential defeater iff l is in the closure of the rules in P not defeated by S.
- r S-undefeatable iff r is not defeated by any S-potential defeater.
- l S-derivable iff l is a consequence of S-undefeatable rules in P.

Now obviously l S-derivable iff $l \in \gamma(\gamma^*(S))$.

Preferences

Definition 2. Let r and r' be rules, S a set of literals. We say r S-dominates r' iff

- $r > r'$, and
- $Cl(\{r\} \cup \{s : s \ is \ S\text{-undefeatable}\})$ defeats r'.

Definition 3. Let $(P, >)$ be a prioritized logic program.

- l S-potential r-defeater iff l is in the closure of rules in P which are (1) not defeated by S and (2) not S-dominated by r.
- r S-safe iff r is not defeated by any S-potential r-defeater.
- l S-derivable iff l is consequence of the set of S-safe rules in P.

Example 3

$$1)\ c \leftarrow not\ \neg c, a$$
$$2)\ \neg c \leftarrow not\ c$$
$$3)\ a$$

Let $1 > 2$.

3 $\emptyset$-safe since there is no way of defeating a rule without default negation.

1 $\emptyset$-safe since the closure of 1 together with the $\emptyset$-undefeatable rule 3 defeats 2 and thus 1 $\emptyset$-dominates 2.

Set of $\emptyset$-derivable literals: $\{c, a\}$.

This set is already the lfp.

The translation

Defeasible theory $T = (R, >)$ translated to extended logic program $Trans(T) = (Trans(R), >')$

$Trans(R) = \{Trans(r) : r \in R\}$ and the translation of each rule is defined as:

$$
\begin{array}{lcl}
\{a_1, \ldots , a_n\} \rightarrow b & \text{becomes} & b \leftarrow a_1, \ldots , a_n \\
\{a_1, \ldots , a_n\} \Rightarrow b & \text{becomes} & b \leftarrow not\ -b, a_1, \ldots , a_n
\end{array}
$$

Furthermore, $Trans(r) >' Trans(r')$ iff $r > r'$.

(from now on we use the same symbol for the two preference relations).

Incorrectness

Proposition 4. *Defeasible logic is incorrect wrt prioritized WFS: there is a defeasible theory $T = (R, >)$ and a formula q such that $+\delta q$ is a consequence of T but $q \notin WFS(Trans(T))$.*

Example 4

1)	$\Rightarrow p$	$p \leftarrow not\ \neg p$
2)	$p \rightarrow q$	$q \leftarrow p$
3)	$\Rightarrow \neg q$	$\neg q \leftarrow not\ q$

Let $3 > 2$. $+\delta\neg q$ is a conclusion with proof $[-\Delta p, -\Delta q, +\delta\neg q]$.

WFS concludes q but not $\neg q$: although 3 has higher priority than 2 it does not dominate 2 since a strict rule can never be defeated.

Restricted correctness

Proposition 5.
Let $T = (R, >)$ *be a defeasible theory such that* $>$ *is defined on defeasible rules only,* $Trans(T) = (P, >)$ *its translation.*

If $+\delta q$ *is a conclusion of* T *then* $q \in WFS(Trans(T))$.

Incompleteness I

Example 5:

1)	$\Rightarrow \neg p$	$\neg p \leftarrow not\ p$
2)	$p \rightarrow p$	$p \leftarrow p$

Assume no preferences.

No proof for $+\delta\neg p$. Although 2 can never be used to derive p, the existence of 2 is regarded sufficient reason not to conclude $\neg p$.

WFS concludes $\neg p$: p is not a potential $\emptyset$-defeater, 1 is thus $\emptyset$-undefeatable and used to derive $\neg p$.

WFS performs a kind of loop checking lacking in defeasible logic.

Incompleteness II

Example 6:

1)	$\Rightarrow p$	$p \leftarrow not\ \neg p$
2)	$p \rightarrow q$	$q \leftarrow p$
3)	$\Rightarrow \neg q$	$\neg q \leftarrow not\ q$

Assume no priorities.

$q \in WFS(Trans(T))$ but $+\delta q$ not a conclusion of T.

Major difference in the way strict rules are treated.

WFS: strict rule applied whenever its antecedents are accepted.

Defeasible logic: if one of the antecedents is only defeasibly derivable then strict rule is treated like a defeasible rule. May be blocked by a conflicting defeasible rule.

Incompleteness III

Example 7:

1)	$\Rightarrow p$	$p \leftarrow not\ \neg p$
2)	$p \rightarrow \neg q$	$\neg q \leftarrow p$
3)	$\Rightarrow q$	$q \leftarrow not\ \neg q$
4)	$q \rightarrow \neg p$	$\neg p \leftarrow q$

Assume $1 > 3$.

Defeasible logic does not conclude $+\delta p$. Only preferences of rules with complementary heads taken into account.

WFS concludes p. Rule 1 is $\emptyset$-safe since the closure of 1 together with the $\emptyset$-undefeatable rule 2 defeats 3, that is 1 $\emptyset$-dominates 3.

Additional conclusions obtained by WFS seem perfectly reasonable.

Conclusions

- Studied relationship between 2 directly skeptical approaches
- Under minor restrictions defeasible logic correct wrt WFS
- Defeasible logic incomplete wrt WFS for following reasons:
  1) lack of loop checking
  2) ambivalent role of strict rules
  3) limited preference handling
- WFS preferable from a semantic point of view: simpler definition, larger class of theories, more reasonable conclusions
- Question: any computational advantages of defeasible logic which outweigh its semantical disadvantages?