Defeasibility in Answer Set Programs via Argumentation Theories

4th International Conference on Web Reasoning and Rule Systems
September 22-24, 2010
Bressanone/Brixen, Italy

Hui Wan*
Michael Kifer**
Benjamin Grosof***

* IBM, TJ Watson
** Stony Brook University
*** Vulcan, Inc.
Outline

\- Introduction
  \- Defeasible reasoning
  \- Difficulties in defeasible reasoning
\- LPDA-ASP framework
  \- Syntax
  \- Semantics
  \- Reduction theorems
\- Advantages of LPDA-ASP
Defeasible Reasoning

A form of common sense reasoning: rules can be true by default but may be defeated

Application domains:
- policies, regulations, and law
- actions, change, and process causality
- Web services
- inductive/scientific learning
- natural language understanding

Existing approaches:
- Courteous Logic Programs (Grosof)
  - The main approach used commercially (IBM Common Rules 1999)
- Defeasible logic (Nute et al.)
- Prioritized defaults (Gelfond & Son)
- Preferred answer sets (Brewka & Eiter)
- Compiling preferences (Delgrande et al.)
- … …
Example of Defeasible Reasoning

\[
@d1 \quad \text{flies}(X) :- \text{bird}(X).
\]

\[
@d2 \quad \text{neg \ flies}(X) :- \text{penguin}(X).
\text{bird}(\text{tweety}).
\text{penguin}(\text{tweety}).
\text{#overrides}(d2,d1).
\]

Answer: \{ \text{bird}(\text{tweety}), \text{penguin}(\text{tweety}), \text{neg \ flies}(\text{tweety}) \}
Problems

- Many approaches
- Many intuitions
- Many semantics
- Many incompatible implementations
- No known approach is satisfactory in all cases
Challenges

\- Integration of different intuitions in a single system

\- Extending the analysis and reasoning techniques from LP with NAF to LP with defeasible defaults
  \- Semantics.
  \- Proof theory / inference.
  \- Would like this to be systematic and straightforward.

\- Extending to higher-order logics (e.g., HiLog) and frames (e.g., F-logic)
The LPDA Framework

Logic Programming with Defaults and Argumentation theories

LPDA program

Strict (non-defeasible) rules

Defeasible rules

Semantics: WFS or ASP

Candidate Argumentation Theories (ATs)

An AT decides when a defeasible rule is defeated
Advantages of LPDA-ASP

- Unifies many previous defeasible LP/ASP approaches to defeasibility
  \ Just one semantics for everything

- Simplifies implementation of defeasible LP
  \ Just one implementation for everything: only the Ats vary

- Leverages most of the previous LP algorithms & optimizations
Argumentation Theory (AT)

- Composed of strict rules (can be generalized)
- Has a unary predicate $\texttt{defeated}$
- May contain auxiliary predicates to define $\texttt{defeated}$. E.g., in **courteous** ATs, defeasible logic AT:
  - \#overrides (prioritization)
  - \#opposes (exclusion constraint)
    - \#opposes( in(boat,sheep), in(boat,wolf) )
  - $\texttt{refuted}$, $\texttt{rebutted}$, $\texttt{conflict}$ (concepts used in argumentation)

- Most existing defeasible LP approaches can be described by some AT
LPDA Theories

\ LPDA-WFS: our earlier work [H. Wan et. al., ICLP-2009]
  \ The initial LPDA framework for the well-founded semantics
  \ Captures a wide range of approaches to defeasible reasoning that are based on WFS
  \ Implemented in Flora-2 and SILK

\ LPDA-ASP: this paper
  \ An LPDA framework for the ASP semantics
  \ Unifies approaches to defeasible reasoning that are based on ASP
  \ Uniquely (among defeasible reasoning approaches) allows disjunctions in rule heads
  \ Implemented modulo a hookup to an ASP reasoner
Example: Yale Shooting

@kpld  
loaded(?Gun,?Time+1) :- loaded(?Gun,?Time).  // Frame axiom 1

@kpunld
neg loaded(?Gun,?T+1) :- neg loaded(?Gun,?T).  // Frame axiom 2

@dd
neg alive(?Time+1) :- neg alive(?Time).  // Frame axiom 3

@liv
alive(?Time+1) :- alive(?Time).  // Frame axiom 4

// A gun becomes unloaded after being fired
@sht1
neg loaded(?Gun,?Time+1) :- shoot(?Gun,?Time).

// The turkey dies after a loaded gun is fired at it
@sht2
neg alive(?Time+1) :- shoot(?Gun,?Time), loaded(?Gun,?Time).

// Axioms for the initial state
alive(1).  // The turkey is alive initially

@unld
neg loaded(g1,1) ∨ neg loaded(g2,1).  // One gun is unloaded initially

@ld
loaded(g1,1) ∨ loaded(g2,1).  // One gun is loaded initially

shoot(g1,1).  // Fire g1 at time 1

// If g1 is unloaded at time 1, fire g2 at time 2.
shoot(g2,2) :- not loaded(g1,1).

// axioms for contradiction and rule priorities
#opposes(alive(?Time), neg alive(?Time)).
#overrides(sht1, kpld).
#overrides(sht2, liv).
Example: An AT in the style of Courteous LP

\[ \text{defeated}(\texttt{?R}) \quad :- \quad \text{defeats}(\texttt{?S}, \texttt{?R}). \]
\[ \text{defeated}(\texttt{?R}) \quad :- \quad \text{trans_defeats}(\texttt{?R}, \texttt{?R}). \]
\[ \text{defeats}(\texttt{?R}, \texttt{?S}) \quad :- \quad \text{refutes}(\texttt{?R}, \texttt{?S}), \not \text{defeated}(\texttt{?R}), \not \text{strict}(\texttt{?S}). \]
\[ \text{trans_defeats}(\texttt{?X}, \texttt{?Y}) \quad :- \quad \text{defeats}(\texttt{?X}, \texttt{?Y}). \]
\[ \text{trans_defeats}(\texttt{?X}, \texttt{?Y}) \quad :- \quad \text{defeats}(\texttt{?X}, \texttt{?Z}), \text{trans_defeats}(\texttt{?Z}, \texttt{?Y}). \]
\[ \text{refutes}(\texttt{?R}, \texttt{?S}) \quad :- \quad \text{conflict}(\texttt{?R}, \texttt{?S}), \text{overrides}(\texttt{?R}, \texttt{?S}), \texttt{?R} = \text{handle}(\texttt{?T}, \texttt{?L}), \texttt{?L}. \]
\[ \text{conflict}(\texttt{?R1}, \texttt{?R2}) \quad :- \quad \texttt{?R1} = \text{handle}(\texttt{?T} \texttt{1}, \texttt{?L1}), \texttt{?R2} = \text{handle}(\texttt{?T} \texttt{2}, \texttt{?L2}), \text{candidate}(\texttt{?R1}), \text{candidate}(\texttt{?R2}), \text{opposes}(\texttt{?L1}, \texttt{?L2}). \]
\[ \text{candidate}(\texttt{?R}) \quad :- \quad \text{body}(\texttt{?R}, \texttt{?B}), \texttt{?B}. \]
\[ \text{overrides}(\texttt{?R}, \texttt{?S}) \quad :- \quad \text{strict}(\texttt{?R}), \not \text{strict}(\texttt{?S}). \]
\[ \text{opposes}(\texttt{?L1}, \texttt{?L2}) \quad :- \quad \text{opposes}(\texttt{?L2}, \texttt{?L1}). \]
\[ \text{opposes}(\texttt{?L}, \texttt{neg ?L}). \]
\[ :- \quad \texttt{?L1}, \texttt{?L2}, \text{opposes}(\texttt{?L1}, \texttt{?L2}). \]
Example: Yale shooting + Courteous(ASP)

\(\Rightarrow\) Two answer sets:

\(\Rightarrow\) neg loaded(g1,1), loaded(g2,1), neg alive(3)

\(\Rightarrow\) loaded(g1,1), neg loaded(g2, 1), neg alive(3)

\(\Rightarrow\) neg – explicit, “classical” style negation
Another use case: An AT for Defeasible Logic (G. Antoniou, D. Billington, G. Governatori, and M. J. Maher)

$\text{defeated}\left(\text{handle}(?T, ?L)\right) :- \#\text{defeated aux}(?T)$.  \textit{// no handles in AT-DL}

$\#\text{defeated aux}(?T) :- \text{\$conflict}(?T, ?S), \text{\head}(?S, ?L), \text{\$definitely}(?L)$.

$\#\text{defeated aux}(?T) :- \#\text{defeater}(?T)$.  \textit{// defeaters produce no inferences}

$\text{\$definitely}(?L) :- \#\text{strict}(?T), \text{\head}(?T, ?L), \text{\body}(?T, ?B), \text{each definite}(?B)$.

$\text{\$overruled}(?T) :- \text{\$conflict}(?T, ?S), \text{\$candidate}(?S), \text{not \$refuted}(?S)$.

$\text{\$refuted}(?S) :- \text{\$conflict}(?T, ?S), \text{\$candidate}(?T), \\
#\text{overrides}(?T, ?S), \text{not \#defeater}(?T)$.

$\text{\$conflict}(?T, ?S) :- \text{\head}(?T, ?L), \text{\head}(?S, \text{neg } ?L), \\
\text{\$candidate}(?T), \text{\$candidate}(?S)$.

\textbf{Negation:}
\begin{itemize}
  \item \text{\textit{neg} - explicit negation}
  \item \text{\textit{not} - default negation}
\end{itemize}

\textbf{Note: just plug that into the system, and one has the defeasible logic KR}
LPDA-ASP Semantics: Least Model

\[ P : \text{an lpda over language } \mathcal{L} \]
\[ AT : \text{an AT over language } \mathcal{L} \]
\[ M : \text{a Herbrand interpretation} \]
\[ \text{a set of not-free literals in the Herbrand Base over language } \mathcal{L} \]

\[ M \text{ is a model of } (P, AT) \text{ when it satisfies} \]
\[ - \text{every plain rule in } P \cup AT \]
\[ - \text{every defeasible rule } r \text{ in } P \text{ such that } \text{defeated}(r) \text{ is not in } M \]

\[ M \text{ is a least model of } (P, AT) \text{ if it is minimal with respect to inclusion} \]
Answer Sets of an LPDA

\ LPDA-ASP Quotient:
\ Let $P$ be a set of disjunctive lpda rules, and $J$ a Herbrand interpretation for $P$
\ The quotient $P/J$ is obtained by:
\ \ Delete the rules in $P$ that contain $\text{not } L$ with $J(\text{not } L) = f$
\ \ For every defeasible rule $\text{at } L_1 \lor \ldots \lor L_n \leftarrow \text{Body}$ in $P$
\ \ \ Delete every $L_i$ such that $\text{defeated(handle(r, L_i))} \in J$
\ \ \ If all the $L_i$ are deleted, delete the entire rule.
\ \ Remove all not-literals and tags from the remaining rules.

The resulting LPDA-ASP quotient is a set of plain Horn rules, hence has a least model.

\ Let: $M$ – Herbrand interpretation, $P$ – disjunctive lpda
\ $M$ is an answer set of $P$ with respect to the argumentation theory AT, if $M$ is a minimal Herbrand model of $P \cup AT$. 

ASP Reduction Theorem

Let $P$ be a (disjunctive) lpda and $AT$ an argumentation theory. The following two sets coincide:

\[ \text{The set of answer sets for the lpda } P \text{ with respect to } AT. \]
\[ \text{The set of answer sets for the ordinary logic program } P' \cup AT', \text{ where } P' \text{ is obtained from } P \text{ by converting every defeasible rule} \]
\[ @r \ L_1 \lor \ldots \lor L_n : \text{- Body } \in P \]
\[ \text{into a collection of plain rules} \]
\[ \lor_{i \in K} L_i : \text{- Body } \land \land_{i \in K} \text{ not } defeated(handle(r, L_i)) \]
\[ \land \land_{j \in N-K} defeated(handle(r, L_j)). \]

for each subset $K \subseteq N = \{1, \ldots, n\}$ and removing all the remaining tags; and $AT'$ is obtained from $AT$ by simply removing all the tags.
Reduction to Non-disjunctive Case

\[ \text{Shifting transformation:} \]
\[ p \lor q \lor s : \text{- body} \]
\[ \text{to} \]
\[ p : \text{- body, not } q, \text{ not } s \]
\[ q : \text{- body, not } p, \text{ not } s \]
\[ s : \text{- body, not } q, \text{ not } p \]

\[ \text{Known result (Ben-Eliyahu & Dechter) about the shifting transformation:} \]
\[ \text{Shifting is not an equivalence transformation in general} \]
\[ \text{But it is in the special case of head-cycle-free disjunctive rules.} \]

\[ \text{An adaptation of this result holds also for LPDA-ASP} \]
Advantages of LPDA-ASP

- Straightforwardly generalizes defeasible LP to:
  - HiLog-style higher-order logics
  - F-logic style object-oriented features
- Unifies previous defeasible LP approaches in one KR
  - Can combine multiple approaches in one system
  - Much simpler to analyze theoretically
- Simplifies implementation
  - Argumentation theories are typically ~ 20-30 vs. 1000’s of lines of code in other approaches
  - Easy to debug and experiment with argumentation theories
- Inherently incremental
  - No need to re-run a complex transformation (unlike some previous approaches)
- Reuses existing LP algorithms and optimizations
Thank you!