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

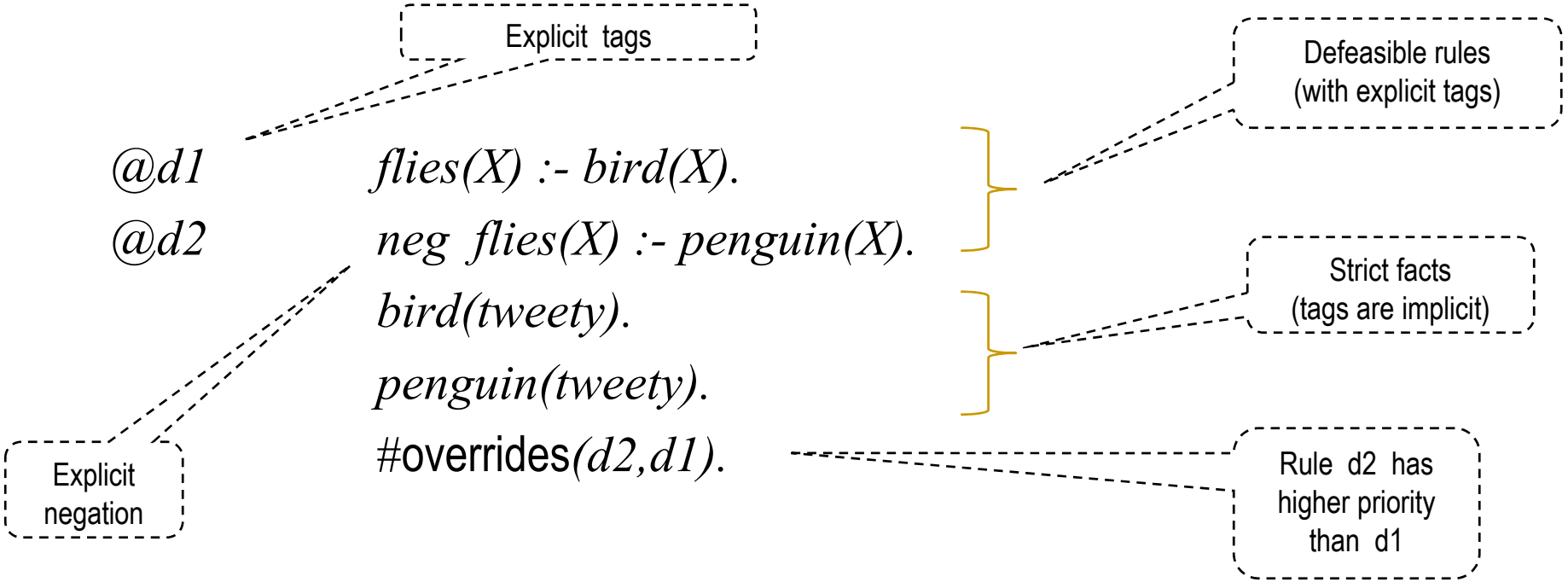
Defeasible Reasoning

- 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



Answer: { *bird(tweety), penguin(tweety), neg flies(tweety)* }

Problems

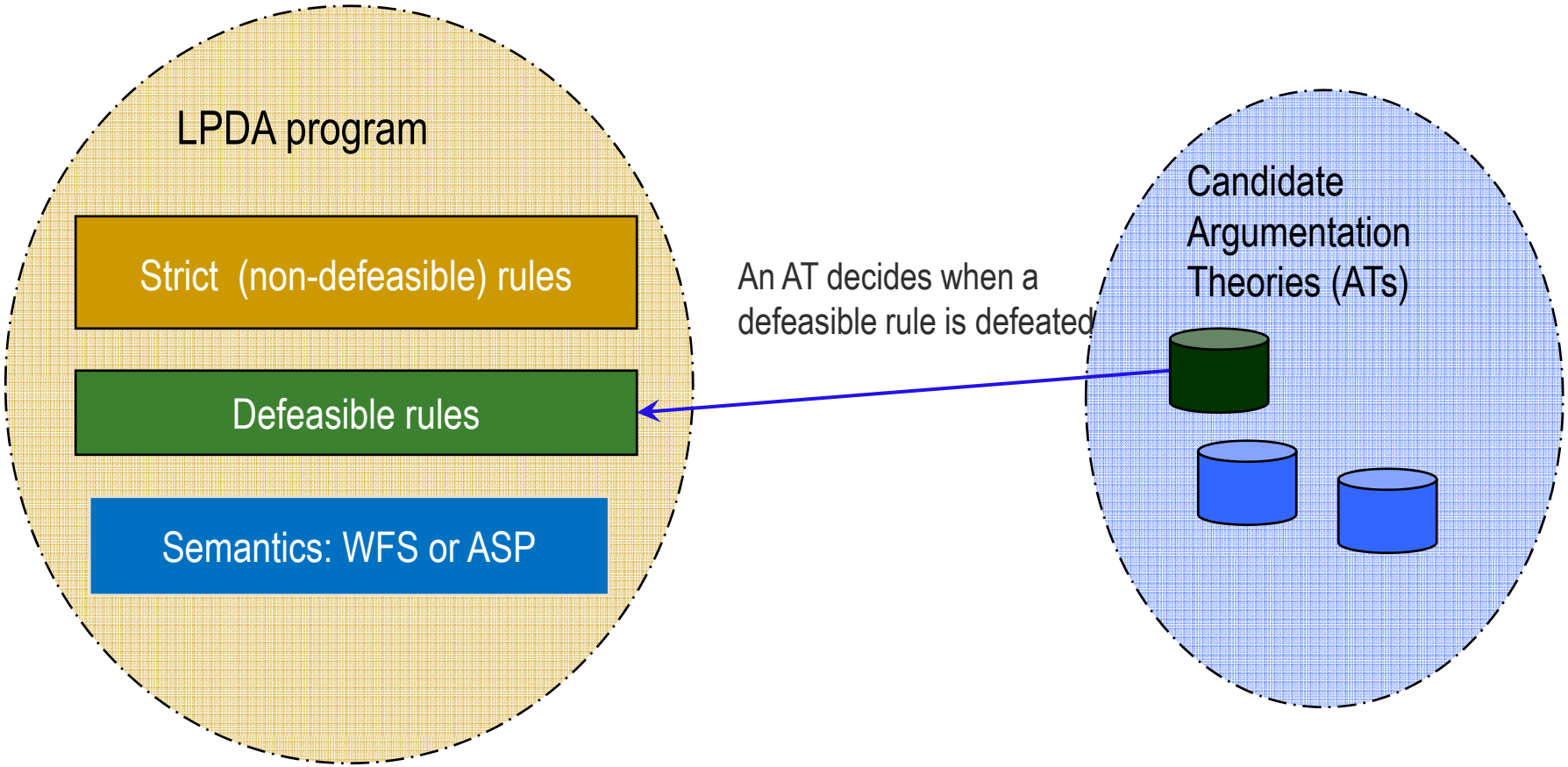
- Many approaches
- Many intuitions
- Many semantics
- Many incompatible implementations
- No known approach is satisfactory in all cases

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



Advantages of LPDA-ASP

- Unifies many previous defeasible LP/ASP approaches to defessibility
  - 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 $defeated
- May contain auxiliary predicates to define $defeated.

  E.g., in courteous ATs, defeasible logic AT:
  - #overrides (prioritization)
  - #opposes (exclusion constraint)
    - #opposes(in(boat,sheep), in(boat,wolf))
  - $refuted, $rebutted, $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) v neg loaded(g2,1).                 // One gun is unloaded initially
@ld       loaded(g1,1) v 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

$defeated(?R) :- $defeats(?S, ?R).
$defeated(?R) :- $trans_defeats(?R, ?R).

$defeats(?R, ?S) :- $refutes(?R, ?S), not $defeated(?R), not #strict(?S).
$trans_defeats(?X, ?Y) :- $defeats(?X, ?Y).
$trans_defeats(?X, ?Y) :- $defeats(?X, ?Z), $trans_defeats(?Z, ?Y).

$refutes(?R, ?S) :- $conflict(?R, ?S), #overrides(?R, ?S), ?R = *handle*(?T, ?L), ?L.

$conflict(?R1, ?R2) :- ?R1 = *handle*(?T 1, ?L1), ?R2 = *handle*(?T 2, ?L2),
$candidate(?R1), $candidate(?R2), #opposes(?L1, ?L2).
$candidate(?R) :- **body**(?R, ?B), ?B.

#overrides(?R, ?S) :- #strict(?R), not #strict(?S).
#opposes(?L1, ?L2) :- #opposes(?L2, ?L1).
#opposes(?L, neg ?L).

:- ?L1, ?L2, #opposes(?L1, ?L2).

?R =handle(?T,?L) - rule handle
?T – rule tag
?L – rule head

not – default negation

Example: Yale shooting + Courteous(ASP)

- Two answer sets:
  - neg loaded(g1,1), loaded(g2,1), neg alive(3)
  - loaded(g1,1), neg loaded(g2, 1), neg alive(3)

- neg – explicit, “classical” style negation

Another use case: An AT for Defeasible Logic
(G. Antoniou, D. Billington, G. Governatori, and M. J. Maher)

```
$defeated(handle(?T, ?L)) :- #defeated aux(?T ).   // no handles in AT -DL
#defeated aux(?T ) :- $conflict(?T, ?S), head(?S, ?L), $definitely(?L).
#defeated aux(?T ) :- #defeater(?T ).                    // defeaters produce no inferences
#defeated aux(?T ) :- $overruled(?T ).

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

$overruled(?T ) :- $conflict(?T, ?S), $candidate(?S), not $refuted(?S).
$refuted(?S) :- $conflict(?T, ?S), $candidate(?T ),
                #overrides(?T, ?S), not #defeater(?T ).
$conflict(?T, ?S) :- head(?T, ?L), head(?S, neg ?L),
                     $candidate(?T ), $candidate(?S).
```

Negation:
neg – explicit negation
not - default negation

- Note: just plug that into the system, and one has the defeasible logic KR

LPDA-ASP Semantics: Least Model

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

- $\boldsymbol{M}$ is a model of $(\boldsymbol{P}, AT)$ when it satisfies
  - every plain rule in $\boldsymbol{P} \cup AT$
  - every defeasible rule r in P such that \$defeated(r) is not in $\boldsymbol{M}$
- M is a least model of $(\boldsymbol{P}, AT)$ if it is minimal with respect to inclusion

Answer Sets of an LPDA

- LPDA-ASP Quotient:
  - Let $\boldsymbol{P}$ be a set of disjunctive lpda rules, and $\boldsymbol{J}$ a Herbrand interpretation for $\boldsymbol{P}$
  - The quotient $\boldsymbol{P/J}$ is obtained by:
    - Delete the rules in $\boldsymbol{P}$ that contain $not\ L$ with $\boldsymbol{J}(not\ L) = \mathrm{f}$
    - For every defeasible rule @r $L_1 \vee \ldots \vee L_n$:- Body in $\boldsymbol{P}$,
      - Delete every L_i such that $defeated(handle(r, L_i)) \in J$
      - 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:

- The set of answer sets for the lpda P with respect to AT.
- The set of answer sets for the ordinary logic program P' ∪ AT' , where P' is obtained from P by converting every defeasible rule

$$@r \;\; L_1 \vee \ldots \vee L_n \;\text{:-}\; \text{Body} \in P$$

into a collection of plain rules

$$\vee_{i \in K} L_i \;\text{:-}\; \text{Body} \wedge \wedge_{i \in K} \text{not } \$\text{defeated}(\text{handle}(r, L_i)) \wedge \wedge_{j \in N-K} \$\text{defeated}(\text{handle}(r, L_j)).$$

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

- Shifting transformation:

  p ∨ q ∨ s :- body

  to

  p :- body, not q, not s
  q :- body, not p, not s
  s :- body, not q, not p

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

- 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!