Preference handling in logic programs under answer set semantics

Alexander Šimko

Departmetnt of Applied Informatics
Comenius University, Bratislava

August 29, 2011

Intro

My interest

- logic programs under answer set semantics
- semantics for preference on rules

The way I see preference handling

- selective approach
- order in which rules are applied is already defined
- comparison of generating sets
- existence of a preferred answer set if a standard one exists

Warranted derivations of preferred answer sets

- with Ján Šefránek
- selective approach to preference handling
- selection as a form of argumentation
- rules (of a program) as a argumentation structures
- preference on rules $\rightarrow$ attack on rules
- derivation rules to derive answer sets from argumentation structures
- derivation rules to derive attacks on argumentation structures

Preferred Answer Sets – Banned Generating Set Approach

- attempt to simplify Warranted derivations

Features

- defined directly on generating sets
- direct definition of attack (no derivation rules)
- only preferences on blocking rules are considered

Main idea

Attacked generating set cannot generate preferred answer set

Attacks

r_1 blocks r_2

$r_1 : a \leftarrow b$

$r_2 : c \leftarrow not\ a$

Attack on rules: r_1 attacks r_2

- r_1 blocks r_2
- r_1 is preferred over r_2

Attacks (2)

Attack on generating sets

R_1 attacks R_2 iff there is $r_1 \in R_1$ and $r_2 \in R_2$ such that r_1 attacks r_2.

Basic idea

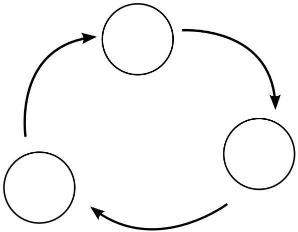
Generating set being attacked cannot generate preferred answer set

- Principle III (there is a preferred answer set when a standard one exists)
- cyclic attacks

Handling cyclic attacks

Effective attack

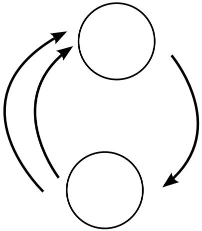
Only effective attack attacks generating set



Attacks from attacked generating set are ineffective.

Mutual attacks

In case of mutual attack of two generating sets, compare number of attacks. (compatibility with Warranted derivations)



Attacks of generating set with less attacks are ineffective.

Pros and Cons

Pros

- there is a preferred answer set whenever there is a standard answer set
- problematic examples from literature are handled correctly due to the Principle IV
- preference on non generating rules plays no role

Cons

Lack of intuition behind:

- definition of effective attack – (technically) oriented to satisfy Principle III
- attack on rules – why to only consider preference on blocking rules?

Accepting the natural order of rules in a logic program with preferences

Preference importance

- when considering two answer sets, find a “core” of the program that is responsible for multiple answer sets.
- preferences over “core” rules are more important
- answer sets can be computed in a interative manner with a branching points (splitting sequence theorem)

Preference importance

$r_1: \quad a \leftarrow not\ b$

$r_2: \quad b \leftarrow not\ a$

$r_3: \quad c \leftarrow a$

$r_4: \quad d \leftarrow b$

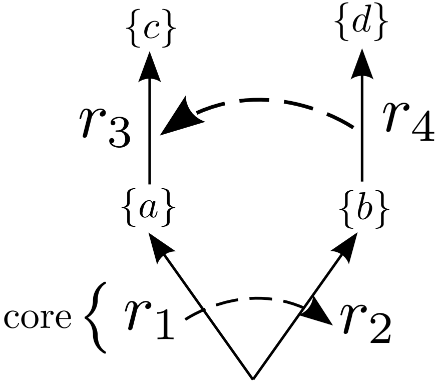
r_1 is preferred over r_2

r_4 is preferred over r_3

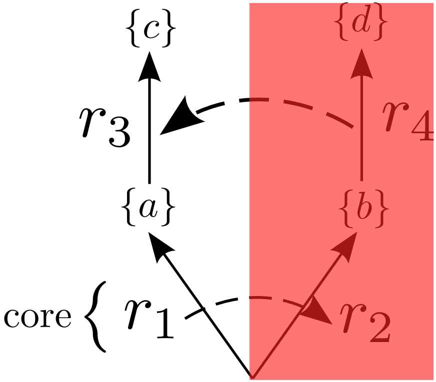
$A_1 = \{a, c\} \qquad R_1 = \{r_1, r_3\}$

$A_2 = \{b, d\} \qquad R_2 = \{r_2, r_4\}$

Preference importance



Preference importance



Preference importance – top

What if there is no preference on rules in "core"?

Whether to use preferences from "top" seems to be domain specific – what preference really is

$r_1 : \quad x \leftarrow$

$r_2 : \quad y \leftarrow$

$r_3 : \quad a \leftarrow x, not\ b$

$r_4 : \quad b \leftarrow y, not\ a$

$r_5 : \quad c \leftarrow b$

r_5 is preferred over r_3

Preference relation on answer sets

- preference relation on rules → preference relation on generating sets
- preference relation on generatingsets → preference relation on answer sets
- select maximally preferred answer sets

Preference relation on answer sets

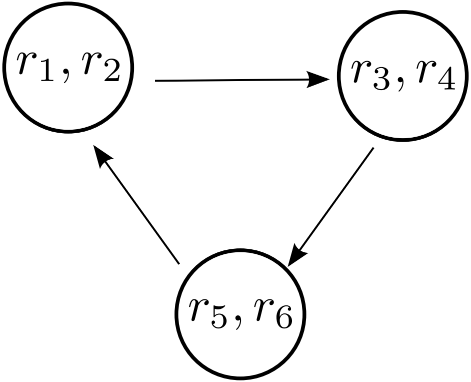
$r_1 \qquad a_1 \leftarrow not\ a_3, not\ d_2$
$r_2 \qquad d_1 \leftarrow not\ a_3, not\ d_2$

$r_3 \qquad a_2 \leftarrow not\ a_1, not\ d_3$
$r_4 \qquad d_2 \leftarrow not\ a_1, not\ d_3$

$r_5 \qquad a_3 \leftarrow not\ a_2, not\ d_1$
$r_6 \qquad a_3 \leftarrow not\ a_2, not\ d_1$

r_1 is preferred over r_4
r_3 je preferred over r_5
r_6 je preferred over r_2

Preference relation on answer sets



There is no maximally preferred answer set

Preference as a constraint

- R_1 generates A_1,
- R_2 generates A_2 ($A_1 \neq A_2$),
- $r_1 \in R_1$, $r_2 \in R_2$ and r_1 is preferred over r_2,

- If we use R_2 to generate A_2 and R_1 is not used to generate A_1 then constraint "r_1 is preferred over r_2" is not met.
- (R_1, R_2) is in a constraint relation on generating sets

- preference relation on rules $\rightarrow$ constraint relation on generating sets
- constraint relation on generating set $\rightarrow$ constraint relation on answer sets

Preference as a constraint

Answer set A is not constrained if there is no constraint (B, A) for any answer set B.

Let S be a set of all answer sets that are not constrained.

Set X of answer sets is preference model iff:

- $X \neq \emptyset$,
- $S \subseteq X$,
- if $A \in X$ and (B, A) is constraint then $B \in X$

Let set Y be a set of all preference models.

A is an answer set iff $A \in \bigcap_{X \in Y} X$