Systems and Solving Techniques for Knowledge Representation

– Disjunctive Logic Programs –

Marco Maratea
University of Genoa, Italy

066 011 Double degree programme Computational Logic (Erasmus-Mundus)
066 931 Computational Intelligence
066 937 Software Engineering & Internet Computing
Institute of Information Systems

ASP Road map

ASP:

Datalog $\leftarrow$ done!

- + Default negation $\leftarrow$ done!
- + Disjunction
- + Integrity Constraints
- + Weak Constraints
- + Aggregate atoms
- + ... and more

Disjunctive Logic Programs Syntax

Rule: (r) $\underbrace{a_1 \mid \ldots \mid a_n}_{head} \text{ :- } \underbrace{b_1, \ldots, b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m.}_{body}$

Atoms, and Literals: a_i , b_i, not b_i
Head of r: $H(r) = \{a_1, \ldots, a_n\}$
Body of r: $B(r) = B^+(r) \cup B^-(r)$
Positive Body: $B^+(r) = \{b_1, \ldots, b_k\}$
Negative Body: $B^-(r) = \{\text{not } b_{k+1}, \ldots, \text{not } b_m.\}$

Variables: as in Datalog, begin with uppercase letter
Safety: Variables must occur in the positive body

Fact: Rule with empty body
Constraint: Rule with empty head

Disjunctive Logic Programs Syntax

Rule: (r) $\underbrace{a_1 \mid \ldots \mid a_n}_{head} \text{ :- } \underbrace{b_1, \ldots, b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m.}_{body}$

Atoms, and Literals: a_i , b_i, $\text{not } b_i$
Head of r: $H(r) = \{a_1, \ldots, a_n\}$
Body of r: $B(r) = B^+(r) \cup B^-(r)$
Positive Body: $B^+(r) = \{b_1, \ldots, b_k\}$
Negative Body: $B^-(r) = \{\text{not } b_{k+1}, \ldots, \text{not } b_m.\}$

Variables: as in Datalog, begin with uppercase letter
Safety: Variables must occur in the positive body

Fact: Rule with empty body
Constraint: Rule with empty head

Examples of Rules

Example (Disjunction)

% Disjunctive knowledge: “A parent P is either a father
% or a mother”

$$mother(P, S) \mid father(P, S) \text{ :- } parent(P, S).$$

Example (Negation, Constraints)

% Default Negation: “Check if an undirected graph
% is not connected”

$$disconnected \text{ :- } node(X), node(Y), \text{not } reachable(X, Y).$$

% Constraints: “Admit only connected graphs.”

$$\text{:- } disconnected.$$

Arithmetic Expressions and Builtins

Arithmetic and comparison operators

- $<, >, <=, >=, =$
- $+, -, *, /$

Example (Fibonacci numbers)

$fib(0, 1).$
$fib(1, 1).$

$fib(N + 2, Y1 + Y2) \coloneq fib(N, Y1), fib(N + 1, Y2).$

For recursive definitions an upper bound for integers (system setting) or a domain has to be specified.

Informal Semantics (1)

Disjunctive Rule:

$$a_1 \mid \ldots \mid a_n \text{ :- } b_1, \ldots, b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m.$$

Informal Semantics:

"If all $b_1, \ldots, b_k$ are true and all $b_{k+1}, \ldots, b_m$ are not true, then at least one among $a_1, \ldots, a_n$ is true".

Example

$isInterestedinASP(john) \mid isCurious(john) \text{ :- } attendsASP(john).$

$attendsASP(john).$

Two (minimal) models encoding two plausible scenarios:

- M_1: $\{isInterestedinASP(john), attendsASP(john).\}$
- M_2: $\{isCurious(john), attendsASP(john).\}$

Informal Semantics (1)

Disjunctive Rule:

$$a_1 \mid \ldots \mid a_n \;{:}\!-\; b_1, \ldots, b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m.$$

Informal Semantics:

"If all $b_1, \ldots, b_k$ are true and all $b_{k+1}, \ldots, b_m$ are not true, then at least one among $a_1, \ldots, a_n$ is true".

Example

$isInterestedinASP(john) \mid isCurious(john) \;{:}\!-\; attendsASP(john).$

$attendsASP(john).$

Two (minimal) models encoding two plausible scenarios:

- M_1: $\{isInterestedinASP(john), attendsASP(john).\}$
- M_2: $\{isCurious(john), attendsASP(john).\}$

Informal Semantics (2)

Constraint:

$$:\!-\; b_1, \ldots, b_k, \mathrm{not}\ b_{k+1}, \ldots, \mathrm{not}\ b_m.$$

Informal Semantics:

"It is not possible that all $b_1, \ldots, b_k$ are true, and all $b_{k+1}, \ldots, b_m$ are false".

Example

$isInterestedinASP(john) \mid isCurious(john) :\!- attendsASP(john).$

$:\!- isNotConvincedinASP(john), isInterestedinASP(john).$

$attendsASP(john). \quad isNotConvincedinASP(john).$

Only one plausible scenario:

- $M_1 : \{isInterestedinASP(john), attendsASP(john).\}$
- $M_2 : \{isCurious(john), attendsASP(john), isNotC.(john).\}$

Informal Semantics (2)

Constraint:

$$:\!-\ b_1, \ldots, b_k, \mathrm{not}\ b_{k+1}, \ldots, \mathrm{not}\ b_m.$$

Informal Semantics:

"It is not possible that all $b_1, \ldots, b_k$ are true, and all $b_{k+1}, \ldots, b_m$ are false".

Example

$isInterestedinASP(john) \mid isCurious(john) :\!-\ attendsASP(john).$

$:\!-\ isNotConvincedinASP(john), isInterestedinASP(john).$

$attendsASP(john).\ \ isNotConvincedinASP(john).$

Only one plausible scenario:

- M_1:$\{isInterestedinASP(john), attendsASP(john).\}$
- M_2:$\{isCurious(john), attendsASP(john), isNotC.(john).\}$

Informal Semantics (3)

Semantics of disjunction is:

- Minimal

  $a \mid b \mid c. \Rightarrow \{a\}, \{b\}, \{c\}$

- Actually subset minimal

  $a \mid b.$

  $a \mid c. \Rightarrow \{a\}, \{b, c\}$

- ...but *not exclusive*

  $a \mid b.$

  $a \mid c.$

  $b \mid c. \Rightarrow \{a, b\}, \{a, c\}, \{b, c\}$

Informal Semantics (3)

Semantics of disjunction is:

- Minimal

  $a \mid b \mid c. \;\Rightarrow \{a\}, \{b\}, \{c\}$

- Actually subset minimal

  $a \mid b.$

  $a \mid c. \;\Rightarrow \{a\}, \{b, c\}$

- ...but *not exclusive*

  $a \mid b.$

  $a \mid c.$

  $b \mid c. \;\Rightarrow \{a, b\}, \{a, c\}, \{b, c\}$

Informal Semantics (3)

Semantics of disjunction is:

- Minimal

  $a \mid b \mid c. \ \Rightarrow \{a\}, \{b\}, \{c\}$

- Actually subset minimal

  $a \mid b.$

  $a \mid c. \ \Rightarrow \{a\}, \{b, c\}$

- ...but *not exclusive*

  $a \mid b.$

  $a \mid c.$

  $b \mid c. \ \Rightarrow \{a, b\}, \{a, c\}, \{b, c\}$

Informal Semantics (4)

- Disjunctive rules "generate models"

  $a \mid b.$

  $a \mid c.$

  $b \mid c.$

  $\Rightarrow \{a, b\}, \{a, c\}, \{b, c\}$

- Integrity constraints "discard" unwanted models:

  % Add:

  $:- a, \text{not } b.$

  $\Rightarrow \{a, b\}, \{b, c\}$

Formal Semantics: Roadmap

1. Instantiation
2. Positive (Ground) Programs
3. Negative Programs
   - via Gelfong & Lifschitz Reduct
   - Positive Programs

Formal Semantics: Program Instantation

Herbrand Universe (U_P): Set of constants occurring in program P
Herbrand Base (B_P): Set of ground atoms constructible from U_P, $pred(P)$
Ground instance of a Rule: Replace each variable in r by constants in U_P

Instantation ground(P): Set of all ground instances of the rules of P

Example (Ground Instantiation)

Consider:

$isInterestedinASP(X) \mid isCurious(X) \text{ :- } attendsASP(X).$
$attendsASP(john).\ \ attendsASP(mary).$

Formal Semantics: Program Instantation

Herbrand Universe (U_P): Set of constants occurring in program P
Herbrand Base (B_P): Set of ground atoms constructible from U_P, $pred(P)$
Ground instance of a Rule: Replace each variable in r by constants in U_P

Instantation ground(P): Set of all ground instances of the rules of P

Example (Ground Instantiation)

Consider:

$$isInterestedinASP(X) \mid isCurious(X) \coloneq attendsASP(X).$$
$$attendsASP(john).\ \ attendsASP(mary).$$

$$U_P = \{john, mary\}$$

$$B_P = \{isInterestedinASP(john), isInterestedinASP(mary), isCurious(john), isCurious(mary), attendsASP(john), attendsASP(mary)\}$$

Formal Semantics: Program Instantation

Herbrand Universe (U_P): Set of constants occurring in program P
Herbrand Base (B_P): Set of ground atoms constructible from U_P, $pred(P)$
Ground instance of a Rule: Replace each variable in r by constants in U_P

Instantation ground(P): Set of all ground instances of the rules of P

Example (Ground Instantiation)

Consider:

$isInterestedinASP(X) \mid isCurious(X) \coloneq attendsASP(X).$
$attendsASP(john). \;\; attendsASP(mary).$

Instantiation:

$isInterestedinASP(john) \mid isCurious(john) \coloneq attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) \coloneq attendsASP(mary).$
$attendsASP(john). \;\; attendsASP(mary).$

$\rightarrow$A non-ground program is a shorthand for its instantiation!$\leftarrow$

Semantics for Positive Programs (1)

Assumptions:

1. programs are ground (obtained from the instantiation)
2. programs are positive, i.e. no negation

Interpretation: A set of atoms $I \subseteq B_P$ of P

- an atom q is true in I if q belongs to I; otherwise it is false
- a literal not q is true w.r.t. I if q is false in I; otherwise it is false
- the head $H(r)$ of a rule r is true w.r.t. I if some literal in $H(r)$ is true w.r.t. I.
- the body $B(r)$ of a rule r is true w.r.t. I if all literals in $B(r)$ are true w.r.t. I.

Semantics for Positive Programs (1)

Assumptions:

1. programs are ground (obtained from the instantiation)
2. programs are positive, i.e. no negation

Interpretation: A set of atoms $I \subseteq B_P$ of P

- an atom q is true in I if q belongs to I; otherwise it is false
- a literal $\text{not}\ q$ is true w.r.t. I if q is false in I; otherwise it is false
- the head $H(r)$ of a rule r is true w.r.t. I if some literal in $H(r)$ is true w.r.t. I.
- the body $B(r)$ of a rule r is true w.r.t. I if all literals in $B(r)$ are true w.r.t. I.

Semantics for Positive Programs (2)

Model:

- *I* is a *model* of *P* if, for every rule *r* in *P*, the head of *r* is true in *I*, whenever the body of *r* is true in *I*

Answer Set: (Positive Program)

- *I* is an *answer set* for a positive program *P* if it is a minimal model (w.r.t. set inclusion) for *P*

  $\rightarrow$ bodies of constraints must be false

Semantics for Positive Programs (2)

Model:

- *I* is a *model* of *P* if, for every rule *r* in *P*, the head of *r* is true in *I*, whenever the body of *r* is true in *I*

Answer Set: (Positive Program)

- *I* is an *answer set* for a positive program *P* if it is a minimal model (w.r.t. set inclusion) for *P*

  $\rightarrow$ bodies of constraints must be false

Semantics for Positive Programs (2)

Model:

- *I* is a *model* of *P* if, for every rule *r* in *P*, the head of *r* is true in *I*, whenever the body of *r* is true in *I*

Answer Set: (Positive Program)

- *I* is an *answer set* for a positive program *P* if it is a minimal model (w.r.t. set inclusion) for *P*

  $\rightarrow$ bodies of constraints must be false

Example of Answer Set for a positive program (1)

$isInterestedinASP(john) \mid isCurious(john) \ {:-}\ attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) \ {:-}\ attendsASP(mary).$
$attendsASP(john). \quad attendsASP(mary).$

$I_1 = \{attendsASP(john)\}$
$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary)\}$
$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$
$I_4 = \{isInterestedinASP(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$
$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$
$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$

Example of Answer Set for a positive program (1)

$isInterestedinASP(john) \mid isCurious(john) \text{ :- } attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) \text{ :- } attendsASP(mary).$
$attendsASP(john).\ \ attendsASP(mary).$

$I_1 = \{attendsASP(john)\}$ (not a model)
$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary),$
$isCurious(mary), attendsASP(mary)\}$
$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary),$
$attendsASP(mary)\}$
$I_4 = \{isInterestedinASP(john), attendsASP(john),$
$isInterestedinASP(mary), attendsASP(mary)\}$
$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary),$
$attendsASP(mary)\}$
$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary),$
$attendsASP(mary)\}$

Example of Answer Set for a positive program (1)

$isInterestedinASP(john) \mid isCurious(john) \text{ :- } attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) \text{ :- } attendsASP(mary).$
$attendsASP(john).\ \ attendsASP(mary).$

$I_1 = \{attendsASP(john)\}$ (not a model)
$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary)\}$ (model, not minimal)
$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$
$I_4 = \{isInterestedinASP(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$
$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$
$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$

Example of Answer Set for a positive program (1)

$isInterestedinASP(john) \mid isCurious(john) \text{ :- } attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) \text{ :- } attendsASP(mary).$
$attendsASP(john).\ \ attendsASP(mary).$

$I_1 = \{attendsASP(john)\}$ (not a model)
$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary)\}$ (model, not minimal)
$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$ (answer set)
$I_4 = \{isInterestedinASP(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary)\}$
$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$
$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary)\}$

Example of Answer Set for a positive program (1)

$isInterestedinASP(john) \mid isCurious(john) :\!- attendsASP(john).$
$isInterestedinASP(mary) \mid isCurious(mary) :\!- attendsASP(mary).$
$attendsASP(john).\ \ attendsASP(mary).$

$I_1 = \{attendsASP(john)\}$ (not a model)
$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary),$
$isCurious(mary), attendsASP(mary)\}$ (model, not minimal)
$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary),$
$attendsASP(mary)\}$ (answer set)
$I_4 = \{isInterestedinASP(john), attendsASP(john),$
$isInterestedinASP(mary), attendsASP(mary)\}$ (answer set)
$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary),$
$attendsASP(mary)\}$ (answer set)
$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary),$
$attendsASP(mary)\}$ (answer set)

Example of Answer Set for a positive program (2)

Let's add:

$:- \; isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + $isNotCASP(john)$.)

$I_1 = \{attendsASP(john), isNotCASP(john)\}$

$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$

$I_4 = \{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$

$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

Example of Answer Set for a positive program (2)

Let's add:

$:- isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + isNotCASP(john).)

$I_1=\{attendsASP(john), isNotCASP(john)\}$ (not a model)

$I_2=\{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_3=\{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$

$I_4=\{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$

$I_5=\{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_6=\{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

Example of Answer Set for a positive program (2)

Let's add:

$:- isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + isNotCASP(john).)

$I_1 = \{attendsASP(john), isNotCASP(john)\}$ (not a model)

$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

(model, not minimal)

$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$

$I_4 = \{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$

$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

Example of Answer Set for a positive program (2)

Let's add:

$:-$ *isNotCASP*(*john*), *isInterestedinASP*(*john*).

isNotCASP(*john*).

(same interpretations as before + isNotCASP(john).)

I_1={*attendsASP*(*john*), *isNotCASP*(*john*)} (not a model)

I_2={*isCurious*(*john*), *attendsASP*(*john*), *isInterestedinASP*(*mary*), *isCurious*(*mary*), *attendsASP*(*mary*), *isNotCASP*(*john*)}
(model, not minimal)

I_3={*isCurious*(*john*), *attendsASP*(*john*), *isInterestedinASP*(*mary*), *attendsASP*(*mary*), *isNotCASP*(*john*)} (answer set)

I_4={*isInterestedinASP*(*john*), *attendsASP*(*john*), *attendsASP*(*mary*), *isInterestedinASP*(*mary*), *isNotCASP*(*john*)}

I_5={*isCurious*(*john*), *attendsASP*(*john*), *isCurious*(*mary*), *attendsASP*(*mary*), *isNotCASP*(*john*)}

I_6={*isInterestedinASP*(*john*), *attendsASP*(*john*), *isCurious*(*mary*), *attendsASP*(*mary*), *isNotCASP*(*john*)}

Example of Answer Set for a positive program (2)

Let's add:

$:- isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + isNotCASP(john).)

$I_1 = \{attendsASP(john), isNotCASP(john)\}$ (not a model)

$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

(model, not minimal)

$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$ (answer set)

$I_4 = \{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$ (not a model)

$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

Example of Answer Set for a positive program (2)

Let's add:

$:- isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + isNotCASP(john).)

$I_1 = \{attendsASP(john), isNotCASP(john)\}$ (not a model)

$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

(model, not minimal)

$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$ (answer set)

$I_4 = \{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$ (not a model)

$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$ (answer set)

$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

Example of Answer Set for a positive program (2)

Let's add:

$:- isNotCASP(john), isInterestedinASP(john).$

$isNotCASP(john).$

(same interpretations as before + isNotCASP(john).)

$I_1 = \{attendsASP(john), isNotCASP(john)\}$ (not a model)

$I_2 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$

(model, not minimal)

$I_3 = \{isCurious(john), attendsASP(john), isInterestedinASP(mary), attendsASP(mary), isNotCASP(john)\}$ (answer set)

$I_4 = \{isInterestedinASP(john), attendsASP(john), attendsASP(mary), isInterestedinASP(mary), isNotCASP(john)\}$ (not a model)

$I_5 = \{isCurious(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$ (answer set)

$I_6 = \{isInterestedinASP(john), attendsASP(john), isCurious(mary), attendsASP(mary), isNotCASP(john)\}$ (not a model)

Semantics for Programs with Negation

Consider *general* programs with negation

Reduct: The *Gelfond-Lifschitz reduct* of a program P w.r.t. an interpretation I is the positive program P^I obtained from P by:

- deleting all rules with a negative literal false in I;
- deleting the negative literals from the bodies of the remaining rules.

Answer Set (or Stable Model): An *answer set* of a general program P^I is an interpretation I such that I is an answer set of P^I.

Example 1

Example (Reduct)

Program:

$a \coloneq d, \text{not } b.$
$b \coloneq \text{not } d.$

$d.$

Consider: $I = \{a, d\}$

Reduct:

$a \coloneq d.$
$d.$

$\rightarrow$ *I is an answer set of* P^I *and therefore is an answer set of* P.

Example 2

Example

Program:

$a \mid b.$

Answer Sets: $\{a\}, \{b\}$

Example 3

Example

Program:

$a \mid b.$
$c \text{ :- } b.$
$c \text{ :- } a.$

Answer Sets: $\{a, c\}, \{b, c\}$

Example 4

Example

Program:

$a \mid b.$
$a \coloneq b.$
$b \coloneq a.$

Answer Sets: $\{a, b\}$

Example 5

Example

Program:

$a \mid b.$
$:\!-\ b.$

Answer Set: $\{a\}$

Support and Answer Sets

Property:

Let A be an answer set of a program P, for all $a \in A$ there exist a rule r such that:

- a is the only true atom in the head of r
- all literals in the body of r are true

Support and Answer Sets

Unfounded Set:

A set of ground atoms X is an unfounded set w.r.t. interpretation I if for each $a \in X$, for each rule r s.t. $a \in H(r)$, one of the following conditions hold

1. the body of r is false, or
2. some literal in the positive body belongs to X
3. an atom in the head of r, distinct from a and other elements in X, is true w.r.t. M.

Property:

Answer sets are unfounded-free interpretations, i.e., no subset is unfounded.

Support and Answer Sets

Unfounded Set:

A set of ground atoms X is an unfounded set w.r.t. interpretation I if for each $a \in X$, for each rule r s.t. $a \in H(r)$, one of the following conditions hold

1. the body of r is false, or
2. some literal in the positive body belongs to X
3. an atom in the head of r, distinct from a and other elements in X, is true w.r.t. M.

Property:

Answer sets are unfounded-free interpretations, i.e., no subset is unfounded.

Thanks to Francesco Ricca for a preliminary version of these slides