Answer Set Programming
Syntax and Semantics

Martin Baláž

Department of Applied Informatics
Comenius University in Bratislava

2009/2010
Alphabet

Definition (Alphabet)

An alphabet $\mathcal{A}$ consists of:

- **variables** $V = \{X, Y, Z, \ldots\}$
- **function symbols** $F = \{f, g, h, \ldots\}$ with arity
  - **constants**
- **predicate symbols** $P = \{p, q, r, \ldots\}$ with arity
  - **propositional variables**
- **logical connectives**
  - nullary $\{\bot, \top\}$
  - unary $\{\sim\}$
  - binary $\{\land, \lor, \leftarrow\}$
- **quantifiers** $\{\forall, \exists\}$
- **punctuation symbols** $\{“(”, “)”, “;”\}$
A language $\mathcal{L}$ is a triple $(F, P, \text{arity})$ where

- $F$ is a set of function symbols (or constants)
- $G$ is a set of predicate symbols (or propositional variables)
- $\text{arity}$ is an arity function $F \cup P \mapsto \mathbb{N}$
Definition (Term)
A *term* is inductively defined as follows:

- A variable is a term.
- A constant is a term.
- If $f$ is an $n$-ary function symbol and $t_1, \ldots, t_n$ are terms then $f(t_1, \ldots, t_n)$ is a term.

A term is said *ground* if no variable occurs in it. A *Herbrand universe* is the set of all ground terms.
Definition (Atom)

An *atom* is defined as follows:

- A propositional variable is an atom.
- If $p$ is an $n$-ary predicate symbol and $t_1, \ldots, t_n$ are terms then $p(t_1, \ldots, t_n)$ is an atom.

An atom is said *ground* if no variable occurs in it. A *Herbrand base* is the set of all ground atoms.
Definition (Formula)

A formula is inductively defined as follows:

- An atom is a formula.
- A nullary connective is a formula.
- If $F$ is a formula, then $\sim F$ is a formula.
- If $F$ and $G$ are formulae, then $F \land G$, $F \lor G$, and $G \leftarrow F$ are formulae.
- If $X$ is a variable and $F$ is a formula, then $(\forall X)F$ and $(\exists X)F$ are formulae.
Definition (Literal)

A *default literal* is an atom preceded by the symbol \( \sim \). A *literal* is either an atom or a default literal.

Definition (Rule)

A *rule* is a formula of the form

\[
L_1 \lor \cdots \lor L_m \leftarrow L_{m+1} \land \cdots \land L_n
\]

where \( L_i, 1 \leq i \leq n \) are literals.

The formula \( L_1 \lor \cdots \lor L_m \) is called the *head* and the formula \( L_{m+1} \land \cdots \land L_n \) is called the *body* of a rule.

A rule with an empty body \( (m = n) \) and a single disjunct in the head \( (m = 1) \) is called a *fact*.

A rule with an empty head \( (m = 0) \) is called a *constraint*. 
Logic Programs

Definition (Logic Program)

A *logic program* is a set of rules. A *positive* logic program does not contain default negation. A *normal* logic program can contain default negation only in the bodies of rules. A *generalized* logic program can contain default negation also in the heads of rules. If a logic program is *disjunctive*, it can contain rules with disjunction in the head, otherwise it can not.

<table>
<thead>
<tr>
<th>generalized logic program</th>
<th>normal logic program</th>
<th>positive (definite) logic program</th>
</tr>
</thead>
<tbody>
<tr>
<td>generalized disjunctive logic program</td>
<td>normal disjunctive logic program</td>
<td>positive disjunctive logic program</td>
</tr>
</tbody>
</table>
variable: \( X \)
constant: 0
function symbol: \( s \) (arity 1)
predicate symbol: \( p \) (arity 1)
terms: \( X, 0, s(X), s(0), s(s(X)), s(s(0)), \ldots \)
atoms: \( p(X), p(0), p(s(X)), p(s(0)), \ldots \)
literals: \( p(X), \sim p(X), p(0), \sim p(0), \ldots \)
Herbrand universe: 0, \( s(0), s(s(0)), \ldots \)
Herbrand base: \( p(0), p(s(0)), p(s(s(0))), \ldots \)
fact: \( p(0). \)
rule: \( p(s(s(X))) \leftarrow p(X). \)
logic program: \( \{ p(s(0)) \leftarrow, p(s(s(X))) \leftarrow p(X) \} \)
An interpretation $I$ of a language $\mathcal{L} = (F, P, \text{arity})$ is a pair $(D, i)$ where

1. $D$ is a domain (non-empty set)
2. $i$ is an interpretation function
   - $i(f) : D^{\text{arity}(f)} \rightarrow D$ for $f \in F$
   - $i(p) : D^{\text{arity}(p)} \rightarrow \{0, 1\}$ for $p \in P$
Valuation

Definition (Variable Valuation)
A variable valuation $\nu$ is a function $V \mapsto D$ from the set of variables $V$ to the given domain $D$.

Definition (Term Valuation)
Let $I = (D, i)$ be an interpretation. The value of a term $t$ with respect to a variable valuation $\nu$ (denoted by $t[\nu]$) is inductively defined as follows:

- $\nu(t)$ if $t$ is a variable
- $i(f)(t_1[\nu], \ldots, t_n[\nu])$ if $t = f(t_1, \ldots, t_n)$ is a function
Definition (Satisfiability)

An interpretation $I = (D, i)$ satisfies a formula $F$ with respect to the variable valuation $\nu$ (denoted by $I \models F[\nu]$) if

- $i(p)(t_1[\nu], \ldots, t_n[\nu]) = 1$ if $F = p(t_1, \ldots, t_n)$ is an atom
- $F = \top$ if $F$ is a nullary connective
- not $I \models F'$ if $F = \neg F'$
- $I \models F_1[\nu]$ and $I \models F_2[\nu]$ if $F = F_1 \land F_2$
- $I \models F_1[\nu]$ or $I \models F_2[\nu]$ if $F = F_1 \lor F_2$
- $I \models F_2[\nu]$ whenever $I \models F_1[\nu]$ if $F = F_2 \leftarrow F_1$
- for all $a \in D$ holds $I \models F'[\nu(X \mapsto a)]$ if $F = (\forall X)F'$
- there exists $a \in D$ such that $I \models F'[\nu(X \mapsto a)]$ if $F = (\exists X)F'$
Definition (Model)

An interpretation $I$ satisfies a formula $F$ (denoted by $I \models F$) if $I$ satisfies a formula $F$ with respect to all variable valuations. An interpretation $I$ is a model of a set of formulae $S$ if $I$ satisfies all formulae in $S$.

Example

$(\forall Y)((\forall X) \sim taller(X, Y) \Rightarrow wise(Y))$
A \textit{Herbrand interpretation} of a language \( L = (F, P, \text{arity}) \) is an interpretation \((D, i)\) where

- \(D\) is a Herbrand universe
- \(i(f): (t_1, \ldots, t_n) \mapsto f(t_1, \ldots, t_n)\)
Theorem

A logic program has a model iff it has a Herbrand model.

Sketch of Proof.

Clearly, if a logic program has a Herbrand model, it has a model. Let $I$ be a model of a logic program $P$. Let $H$ be a Herbrand interpretation such that

$$ H \models A \iff I \models A $$

where $A$ is a ground atom. By induction on the structure of $r \in P$ we can prove that $H$ is a model of $r$, i.e. $H$ is a model of $P$. □
Example

man(dilbert).
single(X) :- man(X), not husband(X).
husband(X) :- man(X), not single(X).
:- single(X), husband(X).

\[ M_1 = \{ \text{man(dilbert), husband(dilbert)} \} \]
\[ M_2 = \{ \text{man(dilbert), single(dilbert)} \} \]