Computational Logic

Logic Programming

Martin Baláž

Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

2011

Motivation

Logic Program:

$$
\begin{aligned}
father(abraham, isaac) &\leftarrow \\
mother(sarah, isaac) &\leftarrow \\
father(isaac, jacob) &\leftarrow \\
parent(X, Y) &\leftarrow father(X, Y) \\
parent(X, Y) &\leftarrow mother(X, Y) \\
grandparent(X, Z) &\leftarrow parent(X, Y), parent(Y, Z) \\
ancestor(X, Y) &\leftarrow parent(X, Y) \\
ancestor(X, Z) &\leftarrow parent(X, Y), ancestor(Y, Z)
\end{aligned}
$$

Query:

$$(\exists X)(\exists Y)ancestor(X, Y)?$$

Answer:

Yes for $X = abraham, Y = isaac; X = sarah, Y = isaac;$
$X = abraham, Y = jacob.$

Logic Program

A *literal* is an atom or the negation of an atom. A *positive literal* is an atom. A *negative literal* is the negation of an atom.

A *rule* is a formula of the form

$$L_1 \vee \cdots \vee L_m \leftarrow L_{m+1} \wedge \cdots \wedge L_n$$

where $0 \leq m \leq n$ and each L_i, $1 \leq i \leq n$, is a literal.

The rule represents a (disjunctive) clause

$$(\forall x_1) \ldots (\forall x_k)(L_1 \vee \cdots \vee L_m \vee \sim L_{m+1} \vee \cdots \vee \sim L_n)$$

where $x_1, \ldots, x_k$ are all variables occurring in $L_1, \ldots, L_n$.

A *logic program* is a finite set of rules.

Definite Logic Program

A *definite rule* is a rule of the form

$$A_0 \leftarrow A_1 \wedge \cdots \wedge A_n$$

where $0 \leq n$ and each A_i, $0 \leq i \leq n$, is an atom.

A *definite logic program* is a finite set of definite rules.

$P \models (\exists x_1) \ldots (\exists x_k)(A_1 \wedge \cdots \wedge A_n)$?
Is $P \cup \{(\forall x_1) \ldots (\forall x_k)(\neg A_1 \vee \cdots \vee \neg A_n)\}$ unsatisfiable?
Is $P \cup \{\leftarrow A_1 \wedge \cdots \wedge A_n\}$ unsatisfiable?

A *definite goal* is a goal of the form

$$\leftarrow A_1 \wedge \cdots \wedge A_n$$

where $0 \leq n$ and each A_i, $1 \leq i \leq n$, is an atom.

Example

$$
\begin{aligned}
p(c, Y, Y) &\leftarrow \\
p(f(X), Y, Z) &\leftarrow \quad p(X, f(Y), Z)
\end{aligned}
$$

- domain $\mathbb{N}$
- signature $(\{c, f\}, \{p\}, \{c \mapsto 0, f \mapsto 1, p \mapsto 3\})$
- interpretation I_1
  - $c^I = 0$
  - $f^I(x) = x + 1$
  - $p^I(x, y, z) \Leftrightarrow z = x + y$
- interpretation I_2
  - $c^I = 1$
  - $f^I(x) = 2 * x$
  - $p^I(x, y, z) \Leftrightarrow z = 2^{x+y}$

Herbrand Interpretation

A term is *ground* if it does not contain variables. Similarly, a formula is *ground* if it does not contain variables.

The *Herbrand universe* $\mathcal{U}$ is the set of all ground terms. Similarly, the *Herbrand base* $\mathcal{B}$ is the set of all ground atoms.

The *Herbrand interpretation* is an interpretation given by the following:

1. The domain is the Herbrand universe
2. If f is a function symbol with arity n, then $f^I = (t_1, \ldots, t_n) \mapsto f(t_1, \ldots, t_n)$

Example

$$\begin{aligned} p(c, Y, Y) &\leftarrow \\ p(f(X), Y, Z) &\leftarrow \quad p(X, f(Y), Z) \end{aligned}$$

- domain $\mathcal{U} = \{c, f(c), f(f(c)), f(f(f(c))), \dots\}$
- signature $(\{c, f\}, \{p\}, \{c \mapsto 0, f \mapsto 1, p \mapsto 3\})$
- interpretation I
  - $c^I = c$
  - $f^I(x) = f(x)$
  - $p^I(x, y, z) \Leftrightarrow x = f^a(c) \wedge y = f^b(c) \wedge z = f^{a+b}(c)$

We will denote Herbrand interpretation as the set of all satisfied grounded atoms.

Properties

An interpretation I is a *model of* a logic program P iff each rule in P is satisfied by I.

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

Proof:
If I is a model of P, then

$$I' = \{p(t_1, \dots, t_n) \in \mathcal{B} \mid I \models p(t_1, \dots, t_n)\}$$

is a Herbrand model of P.

Example:
$S = \{p(a), (\exists x) \sim p(x)\}$, $\mathcal{U} = \{a\}$, $\mathcal{B} = \{p(a)\}$

Let P be a definite logic program and $\mathcal{M}$ be a non-empty set of Herbrand models of P. Then $\bigcap_{M \in \mathcal{M}} M$ is a Herbrand model of P.

Every definite logic program P has the least Herbrand model (denoted M_P).

Proof: The set of all Herbrand models is non-empty, because Herbrand base is a model of P. Therefore the intersection of all Herbrand models is the least model of P.

Let P be a definite logic program. Then $M_P = \{A \in \mathcal{B}_P \mid P \models A\}$.

Proof: $P \models A$ iff $P \cup \{\sim A\}$ is unsatisfiable iff $P \cup \{\sim A\}$ has no Herbrand models iff $\sim A$ is false w.r.t. all Herbrand models of P iff A is true w.r.t. all Herbrand models of P iff $A \in M_P$.

Immediate Consequence Operator

Let P be a definite logic program. An *immediate consequence operator* T_P is defined as follows:

$$T_P(I) = \{A \in \mathcal{B}_P \mid A \leftarrow A_1 \wedge \cdots \wedge A_n \in Ground(P), \{A_1, \ldots, A_m\} \subseteq I\}$$

The iteration $T_P \uparrow n$ is defined as follows:

$$\begin{array}{rcl} T_P \uparrow 0 & = & \emptyset \\ T_P \uparrow (n+1) & = & T_P(T_P \uparrow n) \\ T_P \uparrow \omega & = & \bigcup_{n<\omega} T_P \uparrow n \end{array}$$

Let M_P be the least model of P. Then $M_P = T_P \uparrow \omega$.

SLD-resolution $\equiv$ Linear resolution with Selection function for Definite clauses.

Let G be a goal $A_1 \wedge \cdots \wedge A_k \wedge \cdots \wedge A_m$ and r be a rule $B_0 \leftarrow B_1 \wedge \cdots \wedge B_n$. We say that a goal G' is *a resolvent derived from* G and r *using* θ if θ is the most general unifier of A_k and B_0 and G' has the form
$\leftarrow (A_1 \wedge \cdots \wedge A_{k-1} \wedge B_1 \wedge \cdots \wedge B_n \wedge A_{m+1} \wedge \cdots \wedge A_m)\theta$.

A *SLD-derivation* of $P \cup \{G\}$ is a (posibly infinite) sequence of goals $G_0, \ldots, G_i, \ldots,$ where

- $G_0 = G$
- G_{i+1} is obtained from G_i and a rule r_{i+1} from P using θ_{i+1}

A *successful derivation* ends in empty goal $\leftarrow$. A *failed derivation* ends in non-empty goal with the property that all atoms does not unify with the head of any rule. An *infinite derivation* is an infinite sequence of goals.

Let P be a definite logic program and G be a definite goal. An *answer for* $P \cup \{G\}$ is a substitution for variables in G. An answer θ for $P \cup \{G\}$ is *correct* iff $P \models (A_1 \wedge \cdots \wedge A_n)\theta$ where $G = \leftarrow A_1 \wedge \cdots \wedge A_n$.

Let P be a definite logic program and G be a definite goal G. Let $G_0, \ldots, G_n$ be a successful derivation using $\theta_1, \ldots, \theta_n$. Then $\theta_1 \ldots \theta_n$ restricted to the variables of G is the *computed answer*.

Properties

Let P be a definite logic program and G be a definite goal. Then every computed anwer for $P \cup \{G\}$ is a correct aswer for $P \cup \{G\}$.

Let P be a definite logic program and G be a definite goal. For every correct answer θ for $P \cup \{G\}$ there exists a computed answer σ for $P \cup \{G\}$ and a substitution γ such that $\theta = \sigma\gamma$.

Let P be a definite logic program and G be a definite goal. Then $P \cup \{G\}$ is unsatisfiable iff there exists a successful derivation of $P \cup \{G\}$.

Let M_P be the least model of a definite logic program P. Then $M_P = \{A \in \mathcal{B}_P \mid P \cup \{\leftarrow A\}$ has a successful derivation$\}$.