Lecture 5: Logic Programming

2-AIN-108 Computational Logic

Martin Baláž, Martin Homola

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

23 Oct 2012

Example

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

Syntax

Definition (Literal)

A literal is an atom or the negation of an atom.

Definition (Rule)

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, $0 \leq i \leq n$, is a literal.

Definition (Program)

A logic program is a finite set of rules.

Rules

Each rule

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

can be viewed as a clause

$$L_1 \vee \cdots \vee L_m \vee \neg L_1 \vee \cdots \vee \neg L_n$$

.

A fact is a rule of the form

$$L \leftarrow$$

A constraint is a rule of the form

$$\leftarrow L_1 \vee \cdots \vee L_n$$

Example

$$P = \left\{ \begin{array}{rcl} p(c, Y, Y) & \leftarrow & \\ p(f(X), Y, Z) & \leftarrow & p(X, f(Y), Z) \end{array} \right\}$$

$$L = p(f(c), c, f(c))$$

$$P \models L$$

- domain $\mathbb{N}$
- interpretation function I

- $c^I = 0$
- $f^I = x \mapsto x + 1$
- $p^I = \{(x, y, z) \mid z = x + y\}$

- $c^I = 1$
- $f^I = x \mapsto 2 * x$
- $p^I = \{(x, y, z) \mid z = 2^{x+y}\}$

Herbrand Interpretation

Definition (Herbrand Universe)

A term is ground if it does not contain variables.
The Herbrand universe is the set $\mathcal{U}$ of all ground terms.

Definition (Herbrand Base)

An atom is ground if it does not contain variables.
The Herbrand base is the set $\mathcal{B}$ of all ground atoms.

Definition (Herbrand Interpretation)

A Herbrand interpretation is an interpretation $\mathcal{I} = (\mathcal{U}, I)$ such that

$$f^I = (t_1, \ldots, t_n) \mapsto f(t_1, \ldots, t_n)$$

for each function symbol f with arity n.

Back to the Example

$$P = \left\{ \begin{array}{rcl} p(c, Y, Y) & \leftarrow & \\ p(f(X), Y, Z) & \leftarrow & p(X, f(Y), Z) \end{array} \right\}$$

$$L = p(f(c), c, f(c))$$

$$P \models L$$

- domain $\mathcal{U} = \{c, f(c), f(f(c)), f(f(f(c))), \dots\}$
- interpretation function I
  - $c^I = c$
  - $f^I = x \mapsto f(x)$
  - $p^I = \{(x, y, z) \mid x = f^a(c) \land y = f^b(c) \land z = f^{a+b}(c)\}$

Properties

Theorem

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

Proof.

Each Herbrand model is a model, i.e. if a logic program has a Herbrand model, it has a model.
If $\mathcal{I} = (D, I)$ is a model of P then a Herbrand interpretation $\mathcal{J} = (\mathcal{U}, J)$ such that

$$J(p) = \{(t_1, \ldots, t_n) \mid I \models p(t_1, \ldots, t_n)\}$$

is a Herbrand model of P. □

The previous theorem holds only for clauses, it does not hold for arbitrary closed formulas.

Let S be $\{p(a), (\exists X)\neg p(X)\}$. The Herbrand universe is $\mathcal{U} = \{a\}$ and the Herbrand base is $\mathcal{B} = \{p(a)\}$. We have two Herbrand interpretations, $(\{a\}, I_1)$, $p^{I_1} = \emptyset$ (i.e. $p(a)$ is false), and $(\{a\}, I_2)$, $p^{I_2} = \{(a)\}$ (i.e. $p(a)$ is true). In both cases, S is not satisfied.

But if we take the domain $D = \{0, 1\}$ and the interpretation function I_3 with $a^{I_3} = 0$, $p^{I_3} = \{(0)\}$, then (D, I_3) is a model of S.

Definite Logic Program

Definition (Definite Rule)

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.

Definition (Definite Logic Program)

A logic program is definite if it contains only definite rules.

Definition (Definite Goal)

A definite goal is a rule 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.

Reasoning without Negation

$P \models (\exists X_1) \ldots (\exists X_k)(A_1 \wedge \cdots \wedge A_n)$?
Is $P \cup \{\neg(\exists X_1) \ldots (\exists X_k)(A_1 \wedge \cdots \wedge A_n)\}$ unsatisfiable?
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?

The Least Herbrand Model

Lemma

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.

Theorem

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

Proof.

The set of all Herbrand models is non-empty, because the Herbrand base $\mathcal{B}$ is a model of P. Therefore the intersection of all Herbrand models is the least Herbrand model of P. □

Theorem

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

Definition (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:

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

Theorem

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

Normal Logic Program

Definition (Normal Rule)

A normal rule is a rule of the form

$$A \leftarrow L_1 \wedge \cdots \wedge L_n$$

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

Definition (Normal Logic Program)

A logic program is normal if it contains only normal rules.

Definition (Normal Goal)

A normal goal is a rule of the form

$$\leftarrow L_1 \wedge \cdots \wedge L_n$$

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

Reasoning with Negation

$P \models (\exists X_1) \dots (\exists X_k)(L_1 \wedge \dots \wedge L_n)$?
Is $P \cup \{\neg(\exists X_1) \dots (\exists X_k)(L_1 \wedge \dots \wedge L_n)\}$ unsatisfiable?
Is $P \cup \{(\forall X_1) \dots (\forall X_k)(\neg L_1 \vee \dots \vee \neg L_n)\}$ unsatisfiable?
Is $P \cup \{\leftarrow L_1 \wedge \dots \wedge L_n\}$ unsatisfiable?

$$\begin{aligned} student(joe) &\leftarrow \\ student(bill) &\leftarrow \end{aligned}$$

$$P \models student(jim)?$$

$$P \models \neg student(jim)?$$

$$student(x) \leftrightarrow x = joe \vee x = bill$$

Completion

First step:

$$p(x_1, \ldots, x_m) \leftarrow x_1 = t_1 \wedge \cdots \wedge x_m = t_m \wedge L_1 \wedge \cdots \wedge L_n$$

where $x_1, \ldots, x_m$ are variables not occuring in $L_1 \wedge \cdots \wedge L_n$ and $p(t_1, \ldots, t_m) \leftarrow L_1 \wedge \cdots \wedge L_n$ is a normal rule.

Second step:

$$p(x_1, \ldots, x_m) \leftrightarrow E_1 \vee \cdots \vee E_k$$

where each E_i has the form $x_1 = t_1 \wedge \cdots \wedge x_m = t_m \wedge L_1 \wedge \cdots \wedge L_n$, $E_1, \ldots E_k$ are all transformed rules from the first step with the predicate symbol p in the head, and $x_1, \ldots, x_m$ are new variables.