Answer Set Programming

Stable Models

Martin Baláž

Department of Applied Informatics
Comenius University in Bratislava

2009/2010

Outline

Positive Logic Programs

Definition (Positive Logic Program)

A *positive (definite) logic program* is a set of rules

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

where $n \geq 0$ and $A_i, 0 \leq i \leq n$ are atoms.

Example

```
edge(a, b).
...
path(X, Y) :- edge(X, Y).
path(X, Z) :- edge(X, Y), path(Y, Z).
```

Least Model

Theorem

The intersection of the Herbrand models of a positive logic program is its unique minimal Herbrand model.

Sketch of proof.

Every positive logic program has a model - the Herbrand base is a model. If M_1 and M_2 are models, then $M_1 \cap M_2$ is a model too. $\square$

Immediate Consequence Operator

Definition (Immediate Consequence Operator)

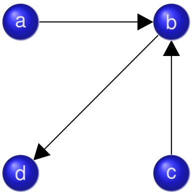
Let Π be a positive logic program. An *immediate consequence operator* is defined as follows:

$$
\begin{aligned}
T_\Pi(I) &= \{A \in \mathcal{B}_\Pi \mid \exists r \in \Pi : head(r) = A, I \models body(r)\} \\
T_\Pi \uparrow \alpha &= \begin{cases} \emptyset & \text{if } \alpha = 0 \\ T_\Pi(T_\Pi \uparrow \beta) & \text{if } \alpha \text{ is a successor ordinal of } \beta \\ \bigcup_{\beta < \alpha} T_\Pi \uparrow \beta & \text{if } \alpha \text{ is a limit ordinal} \end{cases}
\end{aligned}
$$

Theorem

$T_\Pi \uparrow \omega$ is the least model of a positive logic program Π.

Example



Π_D:
```
edge(a, b).
edge(b, d).
edge(c, b).
```
Π_{PS}:
```
path(X, Y) :- edge(X, Y).
path(X, Z) :- edge(X, Y), path(Y, Z).
```

$T_\Pi \uparrow 0$	$=$	$\emptyset$
$T_\Pi \uparrow 1$	+=	{edge(a, b), edge(b, d), edge(c, b)}
$T_\Pi \uparrow 2$	+=	{path(a, b), path(b, d), path(c, b)}
$T_\Pi \uparrow 3$	+=	{path(a, d), path(c, d)}
$T_\Pi \uparrow 4$	$=$	$T_\Pi \uparrow 3$
	...	
$T_\Pi \uparrow \omega$	$=$	$T_\Pi \uparrow 3$

Example

Example

```
p(0).
p(f(X)) :- p(X).
```

$$
\begin{aligned}
T_\Pi \uparrow 0 &= \emptyset \\
T_\Pi \uparrow 1 &\mathrel{+}= \{\texttt{p(0)}\} \\
T_\Pi \uparrow 2 &\mathrel{+}= \{\texttt{p(f(0))}\} \\
T_\Pi \uparrow 3 &\mathrel{+}= \{\texttt{p(f(f(0)))}\} \\
T_\Pi \uparrow 3 &\mathrel{+}= \{\texttt{p(f(f(f(0))))}\} \\
&\dots \\
T_\Pi \uparrow \omega &= \{\texttt{p(0), p(f(0)), p(f(f(0))), ...}\}
\end{aligned}
$$

Normal Logic Programs

Definition (Normal Logic Program)

A *normal logic program* is a set of rules

$$A_0 \leftarrow L_1 \wedge \cdots \wedge L_n$$

where $n \geq 0$, A_0 is an atom, and $L_i, 1 \leq i \leq n$, are literals.

Example

```
man(dilbert).

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

Immediate Consequence Operator

$$T_\Pi(I) = \{A \in \mathcal{B}_\Pi \mid \exists r \in \Pi : head(r) = A, I \models body(r)\}$$

Example

```
a :- not b.
b :- not a.
```

$$
\begin{aligned}
T_\Pi \uparrow 0 &= \emptyset \\
T_\Pi \uparrow 1 &= \{\texttt{a, b}\} \\
T_\Pi \uparrow 2 &= \emptyset \\
T_\Pi \uparrow 3 &= \{\texttt{a, b}\} \\
&\ldots
\end{aligned}
$$

Default Negation

Example

```
fly(X) :- bird(X), not ab(X).
ab(X) :- penguin(X).
bird(X) :- penguin(X).

bird(tweety).
penguin(skippy).
```

Stable Model

Definition (Reduct)

Let I be an interpretation. A *reduct* of a normal logic program Π (denoted by Π^I) is a positive logic program obtained from Π by deleting

- rules containing a default literal L, $I \not\models L$
- default literals L, $I \models L$ from remaining rules

Definition (Stable Model)

An interpretation I is a *stable model* of a normal logic program Π iff I is the least model of Π^I.

Example

Example

```
fly(X) :- bird(X), not ab(X).
ab(X) :- penguin(X).
bird(X) :- penguin(X).

bird(tweety).
penguin(skippy).
```

$$M = \{\texttt{bird(tweety)}, \texttt{penguin(skippy)}, \texttt{bird(skippy)}, \texttt{ab(skippy)}, \texttt{fly(tweety)}\}$$

Example

```
a :- not a.
```

Positive Disjunctive Logic Programs

Definition (Positive Disjunctive Logic Program)

A *positive disjunctive logic program* is a set of rules

$$A_1 \vee \cdots \vee A_m \leftarrow A_{m+1} \wedge \cdots \wedge A_n$$

where $n \geq m \geq 1$ and $A_i, 1 \leq i \leq n$, are atoms.

Example

```
man(dilbert).

single(X) v husband(X) :- man(X).
```

Minimal Models

Theorem

Every positive disjunctive logic program has a Herbrand model.

Sketch of Proof.

The Herbrand base is a model of a positive disjunctive logic program. □

Properties

There exist more minimal models of Π^I.

Example

```
man(dilbert).

single(X) v husband(X) :- man(X).
```

$$M_1 = \{\texttt{man(dilbert)},\ \texttt{single(dilbert)}\}$$
$$M_2 = \{\texttt{man(dilbert)},\ \texttt{husband(dilbert)}\}$$

Normal Disjunctive Logic Programs

Definition (Normal Disjunctive Logic Program)

A *normal disjunctive logic program* is a set of rules

$$A_1 \vee \cdots \vee A_m \leftarrow L_{m+1} \wedge \cdots \wedge L_n$$

where $n \geq m \geq 1$ and $A_i, 1 \leq i \leq m$, are atoms, $L_i, m < i \leq n$ are literals.

Example

```
man(dilbert).

single(X) v husband(X) :- man(X).
```

Stable Model

Definition (Reduct)

Let I be an interpretation. A *reduct* of a normal disjunctive logic program Π (denoted by Π^I) is a positive disjunctive logic program obtained from Π by deleting

- rules containing a default literal $L, I \not\models L$
- default literals $L, I \models L$ from remaining rules

Definition (Stable Model)

An interpretation I is a *stable model* of a normal disjunctive logic program Π iff I is a minimal model of Π^I.