Stable model

Let Π be a Datalog$^{\neg}$ Program.

- Let $\mathcal{B}(\Pi)$ be the set of all facts of the form $R(a_1, \ldots, a_n)$, where R is a n-ary relation in Π and $a_1, \ldots, a_n$ are constants appearing in Π.
- *ground*(Π), the *ground instance* of Π, is obtained by instantiation the rules from Π with all possible combinations of constants from Π.
- The *reduct* of program Π with respect to a set $S \subseteq \mathcal{B}(\Pi)$ is a Datalog^{+} program obtained from *ground*(Π) by deleting
  - (i) each rule that has a negative subgoal $\neg A$ in its body, where $A \in S$, and
  - (ii) all negative subgoals of the remaining rules.
- A set $S \subseteq \mathcal{B}(\Pi)$ is a *stable model* of Π iff S is the unique minimal model of the reduct of Π with respect to S.
- A program may have more than one stable model.

Example 1

Consider the rule $win(X) \leftarrow move(X, Y), \neg win(Y)$ and the input $\mathcal{E}(move) = \{(a, b), (b, c), (c, d)\}$.

$$
ground(\Pi) : \quad
\begin{array}{l}
move(a, b) \leftarrow true \\
move(b, c) \leftarrow true \\
move(c, d) \leftarrow true \\
win(a) \leftarrow move(a, a), \neg win(a) \\
win(a) \leftarrow move(a, b), \neg win(b) \\
win(a) \leftarrow move(a, c), \neg win(c) \\
win(a) \leftarrow move(a, d), \neg win(d) \\
\vdots \\
win(b) \leftarrow move(b, c), \neg win(c) \\
\vdots \\
win(c) \leftarrow move(c, d), \neg win(d) \\
\vdots
\end{array}
$$

$win := \{a, c\}$ is the only stable model.

Example 2

Consider the same rule $win(X) \leftarrow move(X, Y), \neg win(Y)$ and the input $\mathcal{E}(move) := \{(a, b), (b, c), (a, c)\}$.

$$
ground(\Pi) : \quad
\begin{array}{l}
move(a, b) \leftarrow true \\
move(b, c) \leftarrow true \\
move(a, c) \leftarrow true \\
win(a) \leftarrow move(a, a), \neg win(a) \\
win(a) \leftarrow move(a, b), \neg win(b) \\
win(a) \leftarrow move(a, c), \neg win(c) \\
\vdots \\
win(b) \leftarrow move(b, c), \neg win(c) \\
\vdots \\
win(a) \leftarrow move(a, c), \neg win(c) \\
\vdots
\end{array}
$$

$win = \{a, b\}$ is the only stable model.

Example 3

Consider the rule $win(X) \leftarrow move(X, Y), \neg win(Y)$ and the input $\mathcal{E}(move) := \{(a, b), (b, a)\}$.

$$ground(\Pi) : \quad \begin{array}{l} win(a) \leftarrow move(a, b), \neg win(b) \\ win(b) \leftarrow move(b, a), \neg win(a) \\ \vdots \end{array}$$

- There is only one well-founded model which makes both $win(a)$ and $win(b)$ undefined.
- There are two stable models, namely $win(a)$ and $win(b)$.

what to do when there is more than one stable model for a program?

- Decide the program has no semantics, respectively many possible semantics.
- Chose nondeterministically one of the stable models and call it the programs semantics.

The latter case is attractive when we do not care which of the many models to select.

3-colorability of a graph

Let a graph with node-relation *vertex* and edge-relation *edge* be given. Let Π be the program:

$$\begin{aligned}
&color(V, blue) \leftarrow vertex(V), \neg color(V, green), \neg color(V, red)\\
&color(V, green) \leftarrow vertex(V), \neg color(V, blue), \neg color(V, red)\\
&color(V, red) \leftarrow vertex(V), \neg color(V, green), \neg color(V, blue)
\end{aligned}$$

$$noncoloring \leftarrow edge(V, U), color(V, C), color(U, C)$$

Any stable model of Π which does not contain *noncoloring* is a solution of the 3-colorability problem.

Answer set programming: Representing search problems by logical rules. The stable models correspond to the solutions.