From Action Descriptions to Logic Programs

In the context of action languages, the term "literal" is reserved for an expression of the form P or $\neg P$, where P is a fluent symbol. In language $\mathcal{C}$, a *general literal* is a literal or a formula of the form A or $\neg A$, where A is an action symbol.

Consider an action description D in language $\mathcal{C}$ such that for all static laws

$$\textbf{caused } F \textbf{ if } G$$

and all dynamic laws

$$\textbf{caused } F \textbf{ if } G \textbf{ after } U$$

in D,

- F is a literal or the formula *False*,
- G is a conjunction of literals,
- U is a conjunction of general literals.

We will define how to transform D into a logic program with constraints, called the *logic programming representation* of D and denoted by $lp(D)$.

The language of $lp(D)$ has atoms of two kinds:

- the fluent names of D followed by (0) or (1), and
- the action names of D followed by (0).

Program $lp(D)$ consists of the following rules:

(i) for every static law

$$\textbf{caused } F \textbf{ if } L_1 \wedge \cdots \wedge L_m$$

in D, the rule

$$F(1) \leftarrow \mathit{not}\ \overline{L_1(1)}, \ldots, \mathit{not}\ \overline{L_m(1)}$$

($F(1)$ is understood to be empty if F is *False*),

(ii) for every dynamic law

$$\textbf{caused } F \textbf{ if } L_1 \wedge \ldots \wedge L_m \textbf{ after } L_{m+1} \wedge \ldots \wedge L_n$$

in D, the rule

$$F(1) \leftarrow \ \textit{not}\ \overline{L_1(1)}, \ldots, \textit{not}\ \overline{L_m(1)}, L_{m+1}(0), \ldots, L_n(0),$$

(iii) the rules

$$\begin{array}{l} \neg B \leftarrow \textit{not}\ B, \\ B \leftarrow \textit{not}\ \neg B \end{array}$$

for all atoms B that end with (0).

For instance, if D is

$$\begin{array}{l} \textbf{default } P \\ A \textbf{ causes } \neg P. \end{array} \tag{1}$$

(Problem 18′) then $lp(D)$ is

$$\begin{array}{l} P(1) \leftarrow \textit{not}\ \neg P(1), \\ \neg P(1) \leftarrow A(0), \\ \neg P(0) \leftarrow \textit{not}\ P(0), \\ P(0) \leftarrow \textit{not}\ \neg P(0), \\ \neg A(0) \leftarrow \textit{not}\ A(0), \\ A(0) \leftarrow \textit{not}\ \neg A(0). \end{array}$$

According to the theorem from Esra Erdem's note on exogenous atoms, the last 4 rules of this program can be equivalently replaced by

$$\begin{array}{l} P(0)\ ;\ \neg P(0) \leftarrow\ , \\ A(0)\ ;\ \neg A(0) \leftarrow\ . \end{array}$$

The following theorem shows that the edges of the transition system described by D can be found by computing answer sets for $lp(D)$.

Theorem 1. *A transition $\langle s, a, s' \rangle$ is causally explained by D iff*

$$\{L(0)\ :\ L \in s \cup a\} \cup \{L(1)\ :\ L \in s'\}$$

is an answer set for $lp(D)$.

Assignment DLV-4. Use DLV to find the causally explained transitions (a) for action description (1) above and (b) for action description (5) from the paper *An Action Language Based on Causal Explanation.*