Encoding of a System Description

The encoding $\Pi(\mathcal{SD})$ of system description $\mathcal{SD}$ consists of the encoding of the signature of $\mathcal{SD}$ and rules obtained from statements of $\mathcal{SD}$.

- **Encoding of the Signature**

  We start with the encoding $sig(\mathcal{SD})$ of the signature of $\mathcal{SD}$.

  - For each constant symbol c which has a sort $sort\_name$ other than $fluent$, $static$ or $action$, $sig(\mathcal{SD})$ contains

  $$sort\_name(c) \tag{1}$$

  - For every static g of $\mathcal{SD}$, $sig(\mathcal{SD})$ contains

  $$static\_name(g) \tag{2}$$

  - For every inertial fluent f of $\mathcal{SD}$, $sig(\mathcal{SD})$ contains

  $$fluent(inertial, f) \tag{3}$$

  - For every defined fluent f of $\mathcal{SD}$, $sig(\mathcal{SD})$ contains

  $$fluent(defined, f) \tag{4}$$

  - For every action a of $\mathcal{SD}$, $sig(\mathcal{SD})$ contains

  $$action(a) \tag{5}$$

- **Encoding of Statements of** $\mathcal{SD}$

  For this encoding we only need two steps, 0 and 1, which stand for the beginning and the end of a transition. This is sufficient for describing a single transition; however, later, we describe longer chains of events and let steps range over $[0, n]$ for some constant n. To allow an easier generalization of the program we encode steps by using constant n for the maximum number of steps, as follows:

  $$\#const\ n = 1. \tag{6}$$

  $$step(0..n). \tag{7}$$

  As in our blocks-world example, we introduce a relation $holds(f, i)$ which says that fluent f is true at step i. To simplify the description of the encoding, we also introduce a new notation, $h(l, i)$ where l is a domain literal and i is a step. If f is a fluent then by $h(l, i)$ we denote $holds(f, i)$ if $l = f$ or $\neg holds(f, i)$ if $l = \neg f$. If l is a static literal then $h(l, i)$ is simply l. We also need relation $occurs(a, i)$ which says that action a occurred at step i; $occurs(\{a_0, \ldots, a_k\}, i) =_{def} \{occurs(a_j, i) : 0 \leqslant j \leqslant k\}$.

  We use this notation to encode statements of $\mathcal{SD}$ as follows:

- For every causal law

$$a \textbf{ causes } l \textbf{ if } p_0, \ldots, p_m$$

$\Pi(\mathcal{SD})$ contains

$$\begin{array}{lll} h(l, I+1) & \leftarrow & h(p_0, I), \ldots, h(p_m, I), \\ & & occurs(a, I), \\ & & I < n. \end{array} \tag{8}$$

- For every state constraint

$$l \textbf{ if } p_0, \ldots, p_m$$

$\Pi(\mathcal{SD})$ contains

$$h(l, I) \leftarrow h(p_0, I), \ldots, h(p_m, I). \tag{9}$$

- $\Pi(\mathcal{SD})$ contains the CWA for defined fluents:

$$\begin{array}{lll} \neg holds(F, I) & \leftarrow & fluent(defined, F), \\ & & not\ holds(F, I). \end{array} \tag{10}$$

- For every executability condition

$$\textbf{impossible } a_0, \ldots, a_k \textbf{ if } p_0, \ldots, p_m$$

$\Pi(\mathcal{SD})$ contains

$$\neg occurs(a_0, I) \text{ or } \ldots \text{ or } \neg occurs(a_k, I) \leftarrow h(p_0, I), \ldots, h(p_m, I). \tag{11}$$

- $\Pi(\mathcal{SD})$ contains the Inertia Axiom:

$$\begin{array}{lll} holds(F, I+1) & \leftarrow & fluent(inertial, F), \\ & & holds(F, I), \\ & & not\ \neg holds(F, I+1), \\ & & I < n. \end{array} \tag{12}$$

$$\begin{array}{lll} \neg holds(F, I+1) & \leftarrow & fluent(inertial, F), \\ & & \neg holds(F, I), \\ & & not\ holds(F, I+1), \\ & & I < n. \end{array} \tag{13}$$

- $\Pi(\mathcal{SD})$ contains CWA for actions:

$$\neg occurs(A, I) \leftarrow\ not\ occurs(A, I). \tag{14}$$