KNOWLEDGE REPRESENTATION, REASONING AND DECLARATIVE PROBLEM SOLVING

By Jicheng Zhao
based on the notes of others

Class 13: Specification in Reasoning About Actions

Problem specification

Provide specifications as in software engineering is important for good programming practice:

1. Input: what kind of information or knowledge is provided by the system. There are two types of input:
   - Domain description (denoted as D)
   - Initial state representation (denoted as O)
2. Queries: what kind of questions to be asked.
3. Define what it means for something to be a correct answer in respect to a query.
   I.e. (D,O) $\models$ Q (from D, O, can we make the conclusion Q?)

4. Programming: Prove that our implementation indeed can produce the results that we claim to be, i.e.

$$(D, O) \models Q$$

iff

$$\Pi_1(D, O) \models_{smodels} tr(Q).$$

$tr(Q)$ is some transformation of the query Q.

Specification

1. Domain description D:

$$a \textbf{ causes } f \textbf{ if } p_1, \cdots, p_n, \neg q_1, \cdots, \neg q_r.$$

where

- a is an action, i.e. something that affects the world, such as *shoot*, *load*.
- $p_1, \ldots, p_n, q_1, \ldots, q_r$ are fluents, i.e. properties of the world, such as *alive*, *loaded*.
- f is a fluent literal, such as *alive*, $\neg$*alive*.

So domain description describes how actions affect the fluents in the world.

2. Initial state representation O:

$$\textbf{initially } f.$$

3. Queries Q:

$$f \textbf{ after } a_1, \cdots, a_n.$$

   implies that after doing actions $a_1, \cdots, a_n$, will f be true?

   e.g. $alive$ **after** $load, shoot$.

Incomplete initial state and state transition

- We need further to clarify the interpretation of the initial state. Considering the following example:

Example 1 *Initially, fluent f is false.*

$$\textbf{\textit{initially}}\ \neg f.$$

Actions are

$$\begin{array}{r} a\ \textbf{\textit{causes}}\ f\ \textbf{\textit{if}}\ p. \\ a\ \textbf{\textit{causes}}\ f\ \textbf{\textit{if}}\ \neg p. \end{array} \tag{1}$$

and the query is: will f be true after a?

$$f\ \textbf{\textit{after}}\ a.$$

This specification does not find any information about p or $\neg p$. So, if the initial information is incomplete it will be impossible to tell about the consequences. Because initial information give information about the initial states and if it is incomplete that means multiple states may be possible.

Even though we do not know if p is true or not, we should still be able to conclude that f becomes true after action a, because p can only have two values – either it is true or false.

One solution is to define the initial state and state transitions then it is easier to define the problem.

A state δ is a set of fluents. We say f is true in δ if $f \in \delta$ and $f$is false in δ if $f \notin \delta$.

We will continue this next class.