Complex Action Theories

Tran Cao Son

Department of Computer Science
New Mexico State University
Las Cruces, NM 88011, USA

2007

Example

Story

A car starts driving from Las Cruces to El Paso with the average speed of 45mph at 9am. It consumes 1 gallon of gasoline for 20 miles. Show that if the distance between Las Cruces and El Paso is 60 miles and the car has 10 gallons in the tank then it can go back to Las Cruces without having to buy gasoline. Furthermore, the car can be back in Las Cruces at 5pm.

Question

How can we formulate this problem?

Formalization — Problems

- *Actions takes time to finish*
  Example: driving from Las Cruces to El Paso takes time
- *Actions consume resources*
  Example: driving consumes gasoline
- Actions produce resources
  Example: buying gas “produces” gasoline (for the tank)
- *Effects of actions materialize at different time points*
  Example: the car is in motion ’immediately’ after the action driving is executed; and stays true until the action completed; the distance changes continuously; arriving at the destination is trueonly when the car stops;
- *Fluents are continuous*
  Example: distance (relative) from Las Cruces

Questions?

- What can be simplified?
  Perhaps: action with fixed duration
- How to represent actions and effects?
- What is a state?
- What is a transition?

Observations

- Fluents have their own domains
  - *distance* (from Las Cruces) is non-negative integer (or real); might have upper bound, depending on domain (in the story: less than or equals 60); granularity: miles vs. yeards ves. feets;
  - *gas_in_tank* (the amount of gasoline left in the tank) is non-negative integer (or real); might have upper bound, depending on domain (in the story: less than or equals 20); granularity: gallons vs. quart;
  - *moving* — describes the status of the car, boolean
  - *at*(*lc*), *at*(*elpaso*),
- Effects of actions:
  - becomes true during the execution (*driving*)
  - becomes true after the execution (*at*(*elpaso*))
  - changes during the execution (*gas_in_tank*)
  - becomes true after the complete of the action (pedestrian light becomes white only 10 seconds after the action of pushing the button finishes)

Formalization — Intuitions

1. $drive_{0,t}(v)$: driving from unit 0 to t with speed v ($t = 4$ in our story, each unit is 20 minutes, v is 15 miles per unit;
2. *gas_in_tank*: denote the amount of gasoline available in the tank. 20 miles per gallon means that it needs $v/20$ gallon per unit; So, if the car drives for t units then it needs $v * t/20$ gallons.
3. Effects of the action $drive_{0,t}(v)$:
   - *moving* becomes true right after the execution of the action; false at the end
   - *distance* changes continuously
   - *gas_in_tank* changes continuously

- The executability condition saying that the action $drive_{0,t}(v)$ is executable if $gas\_in\_tank \geq 20$ can be expressed by:
  executable $drive_{0,t}(v)$ **if** $gas\_in\_tank \geq v * t/20$.
- The effect of the action $drive_{0,t}(v)$ on the fluent *distance* (encoding how far from the initial position the car is) is expressed as:
  $drive_{0,t}(v)$ **causes** $distance = distance + v * tp$ **from** 0 **to** t
- We can record the information that the action $drive_{0,t}(v)$ is under execution by the following propositions:
  $drive_{0,t}(v)$ **causes** *moving* **from** 0 **to** t.
  $drive_{0,t}(v)$ **causes** $\neg moving$ **from** t **to** t.
- We can express that driving at a velocity v consumes $v/20$ unit of gasoline per unit time as
  $drive_{0,t}(v)$ **contributes** $-v * tp/20$ **to** gas_in_tank **from** 0 **to** t.

tp — time elapsed from the start of the action

Example

The initial state: $gas\_in\_tank = 10$, $distance = 0$, $moving = false$, and $v = 15$ then the level of gasoline in the tank at the time $0, 1, \ldots, 4, 5$ will be as follows, assuming that $drive_{0,4}(15)$ is executed at the time 0:

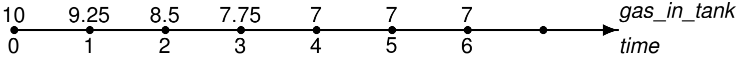

Figure: Gasoline in tank and time – $drive_{0,4}(15)$ at 0

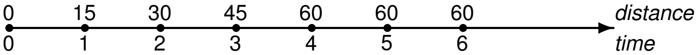

Figure: Distance and time – $drive_{0,4}(15)$ at 0

We still need the static causal laws. Examples:

$$\begin{array}{l} at(elpaso) \textbf{ if } distance = 60 \\ \neg at(elpaso) \textbf{ if } distance \neq 60 \\ at(lc) \textbf{ if } distance = 0 \\ \neg at(lc) \textbf{ if } distance \neq 60 \end{array}$$

$$\textbf{executable } a \textbf{ if } c_1, \ldots, c_k \tag{1}$$
$$a \textbf{ needs } r_1, \ldots, r_m \tag{2}$$
$$a \textbf{ causes } f = valf(f, f_1, \ldots, f_n, t) \textbf{ from } t_1 \textbf{ to } t_2 \tag{3}$$
$$a \textbf{ contributes } valf(f, f_1, \ldots, f_n, t) \textbf{ to } f \textbf{ from } t_1 \textbf{ to } t_2 \tag{4}$$
$$a \textbf{ initiates } p \textbf{ from } t_s \tag{5}$$
$$a \textbf{ terminates } p \textbf{ at } t_s \tag{6}$$
$$p \textbf{ is\_associated\_with } f = valf(f, f_1, \ldots, f_n, t) \tag{7}$$
$$p \textbf{ is\_associated\_with } f \leftarrow valf(f, f_1, \ldots, f_n, t) \tag{8}$$

where

- *a* is an action name, the *f*'s are fluents, the *c*'s are atoms,

- the r's are atoms of the specific form $f = \phi$ where f is a numeric-type fluent and ϕ is an evaluable expression,
- t_s, t_1, t_2 ($t_1 \leq t_2$) are non-negative real numbers, representing time units relative to the time point where a's execution is started,
- $valf(f, f_1, \ldots, f_n, t)$ is a function that takes the value of the fluents $f, f_1, \ldots, f_n$ when a started its execution and the elapsed time t in $[0, t_2 - t_1]$ and returns a value from $dom(f) \cup \{undefined\}$, and
- p is a process name and (7) states that p is associated with the *fluent definition* "$f = valf(f, f_1, \ldots, f_n, t)$" whereas (8) says that it is associated with the *update expression* "$f \leftarrow valf(f, f_1, \ldots, f_n, t)$", where f and $valf(f, f_1, \ldots, f_n, t)$ have the same meaning as in the above item.

Processes

- $turn\_on$ **initiates** $start\_light\_on$ **from** 0 and $start\_light\_on$ **is_associated_with** $light\_on$ (or $light\_on = true$ (turn on the light causes the light to be on)
- $turn\_off$ **terminates** $stop\_light\_on$ **at** 0 and $stop\_light\_on$ **is_associated_with** $\neg light\_on$

Semantics—Ideas

1. *State*: $\langle I, O \rangle$ where I is an interpretation of state variables and O is a set of future effects
2. *Plan*: $\langle A_1[t_1]; A_2[t_2]; \ldots, A_k[t_k] \rangle$ where t_i's are relative time to the current state and $t_1 < t_2 < \ldots < t_k$.
3. *Query*: c **after** $\langle A_1[t_1]; \ldots ; A_n[t_n] \rangle$ where $c = \phi[t_1, t_2]$ (whether the formula ϕ holds from t_1 to t_2.
4. *Transition function*: $\Phi(s, \alpha, \epsilon)$, where $\alpha = \langle A_1[t_1]; A_2[t_2]; \ldots, A_k[t_k] \rangle$ is a plan, s is state, and ϵ is a non-negative number indicating the units of time from *now*, the current state.

Semantics—Example: $drive_{0,4}(15)$

S_0

$\langle\{distance = 0, gas\_in\_tank = 10, \neg moving\}, \emptyset\rangle$

S_{0^+}

$\langle\{distance = 0, gas\_in\_tank = 10, moving\}, \{(1, distance = distance + 15 * tp, 0.4)(2, gas\_in\_tank- = 15 * tp/20, 0, 4)\}\rangle$

S_1 ($tp = 1$)

$\langle\{distance = 15, gas\_in\_tank = 9.25, moving\}, \{(1, distance = distance + 15 * tp, 0, 3), (2, gas\_in\_tank- = 15 * tp/20, 0, 3)\}\rangle$

Answering Queries and Planning

$$\Phi(s, \alpha, \epsilon + \Delta) = \Phi(\Phi(s, \alpha, \epsilon), \beta, \Delta)$$

where β is obtained from α by (i) removing all $A[t]$ where $t < \Delta$; and (ii) replacing each remaining $A[t]$ by $A[t - \Delta]$.

So, Φ is like the transition function for us