Knowledge, action, and the frame problem

Reasoning about Actions

Francesco Petri, Flavio Maiorana and Charlotte Primiceri

May 29, 2024

SAPIENZA UNIVERSITÀ DI ROMA

Table of Contents

Outline

Situation calculus provides a framework for reasoning about actions.
This work presents an expansion to handle the *knowledge* possessed or acquired by the agent, and allow it to shape the agent's decisions.

- Knowledge is represented by one additional fluent
- Uniform axiomatization with the rest of sitcalc
- Ordinary actions and knowledge-producing ones are strictly separated
- Easy expansion of regression as defined in [Reiter2001]
- Desirable properties are direct consequences of the axiomatization (e.g. knowledge persistence / memory)

...

Opzionale
Un paio di azioni ordinarie e un paio di azioni di conoscenza di esempio, giusto per inquadrare il discorso

The K fluent

$$K(s', s)$$

Defines an accessibility relation between situations.

(Informal) definition

$K(s', s)$ is true if an agent in situation s could mistake the current situation for the other s', given its current knowledge.

Knowledge

Definition of knowledge

A fluent is known to be true (false) in a situation s if and only if it is true (false) in all situations accessible from s.

Shorthand notation: $\textbf{Knows}(\phi, s) \stackrel{\text{def}}{=} \forall s' \, \mathrm{K}(s', s) \rightarrow \phi(s')$

Knowledge-producing actions

Actions that have an effect on the agent's knowledge

SENSE actions

Learn the truth value of a formula. Example: check if a door is open or closed.

$$\textbf{Knows}(\mathrm{P}, \textsc{do}(\textsc{sense}_{\mathrm{P}}, s)) \lor \textbf{Knows}(\neg\mathrm{P}, \textsc{do}(\textsc{sense}_{\mathrm{P}}, s))$$

READ actions

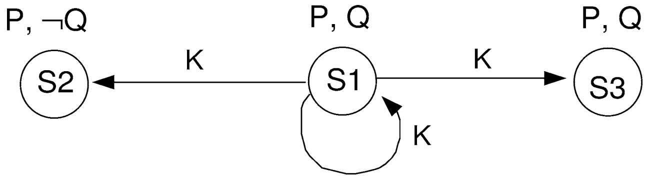
Learn the value of a term. Example: read a number on a sheet of paper.

$$\exists x\, \textbf{Knows}(\tau = x, \textsc{do}(\textsc{read}_{\tau}, s))$$

Assumption: ordinary and knowledge-producing actions are strictly separated.

Knowledge effects

In order to complete the specification of the K fluent, we need to define its successor state axiom, determining how ordinary actions and knowledge-producing actions affect it.
Consider this case study with three accessible situations. The agent is in S1.



$$\textbf{Knows}(\mathrm{P}, S1) \wedge \neg\textbf{Knows}(\mathrm{Q}, S1)$$

Ordinary actions

Assume the agent performs a DROP action.

Informal idea

The agent cannot distinguish the current situation s from all the other s' accessible from it. Therefore, after executing the action, the agent may believe to be in any situation resulting from any s' after executing DROP.

Axiomatization

$$\mathrm{K}(s'', \textsc{do}(\textsc{drop}, s)) \equiv \exists s' \, (\textsc{Poss}(\textsc{drop}, s') \wedge \mathrm{K}(s', s) \wedge s'' = \textsc{do}(\textsc{drop}, s'))$$

Ordinary actions

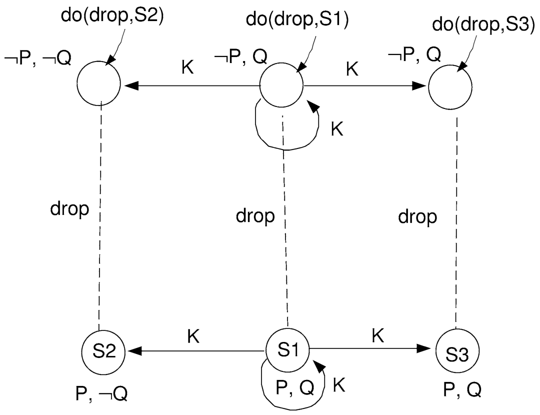
The only knowledge gained is that the DROP action has been performed, as well as anything that can be derived from the action effects.
For example, if DROP makes Pfalse:

$$\mathrm{P}(\mathrm{DO}(a, s)) \equiv a \neq \mathrm{DROP} \wedge \mathrm{P}(s)$$

then

$$\textbf{Knows}(\neg \mathrm{P}, \mathrm{DO}(\mathrm{DROP}, S1))$$

but no extra knowledge is gained about Q.



Knowledge-producing actions

Consider an action $\textsc{sense}_{\textsc{Q}}$

The successor state axiom for K

Pippo

<varie ed eventuali>

Pippo

Table of Contents

The Problem

- A robot has to manage **n plants** in a garden.
- The robot performs an action on one plant at a time and only if it's near the plant.
- A plant can be watered only if it is dry and the temperature is known.
- The robot has access to a watering can that is full and has unlimited capacity.
- The robot can hold only one object at a time.

(Non)Fluents

Fluents

- $\textsc{Near}(x, s) \rightarrow$ Robot is near object x in situation s
- $\textsc{Holding}(x, s) \rightarrow$ Robot is holding object x in situation s
- $\textsc{Moist}(p, s) \rightarrow$ Plant p is moist in situation s
- $\textsc{Temperature}(p) \rightarrow$ Temperature value of the spot near plant p
- $\textsc{HealthyPlants}(s) \rightarrow$ Number of healthy plants

Non-Fluents

- $\textsc{WateringCan}(x) \rightarrow$ Object x is a watering can
- $\textsc{Thermometer}(x) \rightarrow$ Object x is a thermometer
- $\textsc{Moisturemeter}(x) \rightarrow$ Object x is a moisturemeter

Actions

Normal

- $\text{GOTO}(x) \rightarrow$ Go to object x
- $\text{WATER}(p) \rightarrow$ Water plant p
- $\text{PICKUP}(x) \rightarrow$ Pick up object x
- $\text{PUTDOWN}(x) \rightarrow$ Put down object x

Knowledge

- $\text{CHECKMOISTURE}(p) \rightarrow$ Check moisture of plant p
- $\text{CHECKTEMPERATURE}(p) \rightarrow$ Check the temperature of the spot near plant p

Effects

Sensing Result Axioms

- $\text{SR}(\text{GOTO}(x), s) = r \equiv r = \text{"}OK\text{"}$
- $\text{SR}(\text{CHECKMOISTURE}(p), s) = r \equiv (r = \text{"}YES\text{"} \wedge \text{Moist}(p, s)) \vee (r = \text{"}NO\text{"} \wedge \neg\text{Moist}(p, s))$
- $\text{SR}(\text{CHECKTEMPERATURE}(p), s) = r \equiv r = \text{Temperature}(p, s)$

Knowledge Action Effects

- $\textbf{Kwhether}(\text{Moist}(p, s), \text{DO}(\text{CHECKMOISTURE}(p), s))$
- $\textbf{Kref}(\text{Temperature}(p), \text{DO}(\text{CHECKTEMPERATURE}(p), s))$

Preconditions

- $\textsc{Poss}\,(\textsc{water}(p), s) \equiv$
  $\textsc{Near}(p, s) \wedge \textsc{Holding}(x, s) \wedge \textsc{WateringCan}(x) \wedge \neg\textsc{Moist}(x, p) \wedge$
  $\textbf{Kref}\,(\textsc{temperature}(p), s)$
- $\textsc{Poss}\,(\textsc{pickup}(x), s) \equiv \textsc{Near}(x, s) \wedge \neg\exists y.\textsc{Holding}(y, s)$
- $\textsc{Poss}\,(\textsc{putdown}(x), s) \equiv \textsc{Holding}(x, s)$
- $\textsc{Poss}\,(\textsc{checkmoisture}(p), s) \equiv$
  $\textsc{Near}(p, s) \wedge \textsc{Holding}(x, s) \wedge \textsc{MoistureMeter}(x)$

Successor State Axioms

In general $F(x, \text{DO}(\alpha, s)) \equiv \Phi_F^+(x, a, s) \vee (F(x, s) \wedge \neg\Phi_F^-(x, a, s))$

- $\text{NEAR}(x, \text{DO}(\alpha, s)) \equiv \alpha = \text{GOTO}(x) \vee (\text{NEAR}(x, s) \wedge \neg\exists y.\alpha = \text{GOTO}(y))$
- $\text{HOLDING}(x, \text{DO}(\alpha, s)) \equiv \text{PICKUP}(x) \vee (Holding(x, s) \wedge \neg\exists r.\alpha = \text{PUTDOWN}(x))$
- $\text{MOIST}(p, \text{DO}(\alpha, s)) \equiv (\text{MOIST}(p, s) \wedge \neg\exists r.\alpha = \text{WATER}(p)) \vee$
  $(\alpha = \text{CHECKHUMIDITY}(p) \wedge \text{SR}(\alpha(p), s) = h)$
- $\text{TEMPERATURE}(p, \text{DO}(\alpha, s)) \equiv \text{TEMPERATURE}(p, s)$
- $\text{HEALTHYPLANTS}(p, \text{DO}(\alpha, s)) = n \equiv$
  $(\text{HEALTHYPLANTS}(p, s) = n - 1 \wedge \alpha = \text{WATER}(p))$

Initial Situation

- $\textsc{WateringCan}(c)$
- $\textsc{Thermometer}(t)$
- $\textsc{MoistureMeter}(m)$
- $\textsc{Plant}(p_1)$

  $\vdots$

- $\textsc{Plant}(p_n)$
- $\textsc{Near}(c)$

Encoding

while $\neg\mathbf{Knows}(\textsc{HealthyPlants}(s) = n)$ **do** $\textsc{goto}(\mathsf{p})$ Ok
end while

Example Run

Table of Contents

Knowledge, action, and the frame problem

Thank you for listening!
Any questions?