Knowledge Representation and Reasoning

- Knowledge-Producing Actions
- Representing Knowledge
- The Frame Problem with Knowledge
- Reasoning

Sensing Actions

- Generally, agents do not have complete knowledge of the world.
  - formalism must distinguish between what is true in the world and what the agent knows
- Agents must reason about:
  - actions that produce knowledge
    * perception
    * reading
    * communicative acts
  - the knowledge prerequisites of actions

Knowledge and Action


- If John is at the same place as \(SF_1\) and he knows the combination of the safe, he can open the safe by dialing the combination.
- If John is at the same place as both \(SF_1\) and the piece of paper \(PPR_1\) and he knows that the combination of \(SF_1\) is the number written on \(PPR_1\), he can open \(SF_1\) by reading the piece of paper and dialing the combination.
- If \(C_1\) is the combination of \(SF_1\), and if John tries to open \(SF_1\) by dialing \(C_1\), he will then know that \(C_1\) is the combination of \(SF_1\).

Goal

- Account of Knowledge of Agent
- Knowledge-Producing Actions (Sensing)
- Ordinary Actions
- Solution to Frame Problem
- Reasoning Methods
The problem is to make a 3 egg omelet from a set of eggs some of which may be bad. None of the eggs in the omelet should be bad. We have two bowls; we can only see if an egg is bad if it is in a bowl. We can throw out the whole bowl. We can can assume a limited number of eggs (say 5), and simplify it even further by adding the statement that there are at least 3 good eggs. (Savage, Poole)

## The Axiomatization

- **Actions.**
  
  \[
  \begin{align*}
  \text{break\_into(bowl), fetch(container),} \\
  \text{pour(bowl1, bowl2), throw\_out(bowl)}
  \end{align*}
  \]

- **Fluents**
  
  \[
  \begin{align*}
  \text{in(egg, bowl, s), bad(egg, s),} \\
  \text{broken(egg, s), holding(egg, s),} \\
  \text{number\_eggs(bowl, s)}
  \end{align*}
  \]

- **Non-Fluents**
  
  \[
  \begin{align*}
  \text{egg(x), small\_bowl, large\_bowl, basket}
  \end{align*}
  \]

- **Effect Axioms (Causal Laws)**
  
  \[
  \begin{align*}
  \text{holding(e, s) \rightarrow} \\
  \text{in(e, b, do(break\_into(b), s))} \\
  \text{\neg in(e, b_1, do(pour(b_1, b_2), s))}
  \end{align*}
  \]

- **Actions**
  
  - Break an egg into a bowl.
  - Pour the contents of one bowl into another.
  - Throw out the contents of one bowl.
  - Inspect a bowl to see if there are any bad eggs in it.

- **Goal**
  
  - Have three eggs in a bowl that are not bad.

For each action \( A \), an action precondition axiom is needed.

\[
\begin{align*}
\text{Poss(break\_into(bowl), s) \leftrightarrow \exists egg} \\
\text{\neg broken(egg, s) \land} \\
\text{holding(egg, s)}
\end{align*}
\]

\[
\begin{align*}
\text{Poss(pour(b_1, b_2), s) \leftrightarrow} \\
\text{\neg \exists e holding(e, s) \land} \\
\text{number\_eggs(b_1, s) \geq 0}
\end{align*}
\]

\[
\begin{align*}
\text{Poss(fetch(e, con), s) \leftrightarrow} \\
\text{in(e, con, s) \land} \\
\text{\neg \exists e holding(e, s)}
\end{align*}
\]

- An axiomatization of \( S_0 \), the initial state.

\[
\begin{align*}
\text{\neg broken(egg1, S_0) \quad egg(egg1)}
\end{align*}
\]
Solving the Frame Problem
(Continued)

- For all fluents $F$, a successor state axiom is needed.

\[ \text{Broken}(e, do(a, s)) \leftrightarrow \\
(\text{Holding}(e, s) \land \exists b \ a = \text{Break} \_\text{Into}(b)) \lor \\
\text{Broken}(e, s) \]

\[ \text{In}(e, b_1, do(a, s)) \leftrightarrow \\
(\text{Holding}(e, s) \land \exists a = \text{Break} \_\text{Into}(b_1)) \lor \\
(\exists b_2 \ a = \text{Pour}(b_2, b) \land \text{In}(e, b_2, s)) \lor \\
\text{In}(e, b_1, s) \land \neg(\neg \text{Throw} \_\text{Out}(b) \lor \\
\exists b_2 \ a = \text{Pour}(b, b_2)) \]

An Epistemic Fluent

- We (following Moore) adapt the standard possible-world model of knowledge to the situation calculus.

- We introduce a binary relation $K(s', s)$, read as “$s'$ is accessible from $s$.”

- So something is known in $s$ if it is true in every $s'$ accessible from $s$, and conversely something is not known if it is false in some accessible situation.

- We can now introduce the notation $\text{Knows}(P, s)$ as an abbreviation for a formula that uses $K$.

\[ \text{Knows}(\text{Broken}(y), s) \overset{\text{def}}{=} \forall s' K(s', s) \rightarrow \text{Broken}(y, s'). \]

Solving the Frame Problem

- For non-knowledge-producing actions (e.g., $\text{Drop}(x)$), the specification (based on Moore) is as follows:

\[ \forall s, s'', K(s'', \text{do(} \text{Drop}(x), s)) \leftrightarrow \\
\exists s'(K(s', s) \land \text{Poss(} \text{Drop}(x), s') \land \\
(\text{do(} \text{Drop}(x), s')) \]

- The only change in knowledge that occurs in moving from $s$ to $\text{do(} \text{Drop}(x), s)$ is the knowledge that the action $\text{Drop}(x)$ has been performed.
Solving the Frame Problem

- Now consider the simple case of a knowledge-producing action (e.g., \text{INSPECT(bowl)}) that determines whether or not there is a bad egg in the bowl.

\[
P\text{oss}(\text{INSPECT}(b), s) \leftrightarrow \\
\exists e \: \text{Egg}(e) \land \text{IN}(e, b, s) \land \\
\text{Broken}(e)
\]

\[
[K(s'', \text{do} (\text{INSPECT}(b), s))] \leftrightarrow \\
\exists s' (K(s', s) \land \\
(s' = \text{do} (\text{INSPECT}(b), s')) \land \\
\text{Poss}(\text{INSPECT}(b)) \land \\
(\text{BAD}(b, s) \leftrightarrow \text{BAD}(b, s')))]
\]

- The idea here (based on Moore) is that in moving from \( s \) to \( \text{do}(\text{INSPECT}(b), s) \), the agent not only knows that the action \text{INSPECT}(b) has been performed (as before), but also the truth value of the predicate \text{BAD}(b).

The Solution

- In general, there may be many knowledge-producing actions. For each sensing action \( a \), we enter an axiom of the following form, where SF stands for sensed fluent:

\[
\text{SF}(a, s) \leftrightarrow \varphi_i(s)
\]

For the \text{INSPECT} example, the formula would be:

\[
\text{SF}(\text{INSPECT}, s) \leftrightarrow \text{BAD}(e, s)
\]

\[
\text{SF}(\text{DROP}, s) \leftrightarrow T
\]

- Successor State Axiom for K

\[
[K(s'', \text{do} (a, s))] \leftrightarrow \\
\exists s' (\text{Poss}(a, s') \land K(s', s) \land (s'' = \text{do} (a, s'))) \land \\
\text{SF}(a, s') \leftrightarrow \text{SF}(a, s)
\]

Properties of the Solution

- After performing an action, agents always know that the action has been performed.
- Agents know the effects of actions.
- The solution to the frame problem for ordinary actions is maintained.
- Knowledge only changes as appropriate.
- Memory.
Reasoning

- \[ \text{[FETCH, BREAK\_INTO, INSPECT]} \]
- If starting in the initial situation, first an egg is fetched from the basket, then it is broken into the small bowl, and then the bowl is inspected will it be known whether or not the egg in the bowl is bad.

\[
\text{Knows} (\text{BAD(SMALL\_BOWL)},
\begin{align*}
&\text{do} (\text{INSPECT(SMALL\_BOWL)}, \\
&\quad \text{do} (\text{BREAK\_INTO(SMALL\_BOWL)}, \\
&\quad \quad \text{do} (\text{FETCH(BASKET, } S_0)))
\end{align*}
)\\
\]

\[
\text{Knows}(\neg \text{BAD(SMALL\_BOWL)},
\begin{align*}
&\text{do} (\text{INSPECT(SMALL\_BOWL),} \\
&\quad \text{do} (\text{BREAK\_INTO(SMALL\_BOWL)}, \\
&\quad \quad \text{do} (\text{FETCH(BASKET, } S_0))))
\end{align*}
)\\
\]

What about Reading?

A READ action makes known the denotation of a term \( \tau \).

\[ \tau(s) = \tau(s') \]

\[
\text{Kref}(\tau, s) \overset{\text{def}}{=} \exists x \text{Knows}(\tau = x).
\]

Regression

- Regression operators transform something of the form

\[ G(\text{do}(A_1, \text{do}(\ldots, S_0) \ldots) \]

\[ R(S_0) \]

- Theorem

\[ \mathcal{F} \models G(\text{do}(A_1, \text{do}(\ldots, S_0) \ldots) \]

\[ \text{iff} \]

\[ \mathcal{F} - \mathcal{F}_s \models R(S_0) \]

- All reasoning about actions is done by regression.

- \( R(S_0) \) can be expressed in modal logic.

References


Chapter 11 from Reiter Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.