EECS 3401 — AI and Logic Prog. — Lecture 16

Adapted from slides of Brachman & Levesque (2005)

Vitaliy Batusov
vbatusov@cse.yorku.ca

York University

November 16, 2020

- Today: **Actions and Situations**
- Required reading: R & N Ch.11 (10 in 3-rd ed.)

Planning: more than search

Two strands in AI covered so far:

- Symbolic reasoning
  knowledge representation & reasoning using FOL
  Static knowledge, complex but unchanging state
- Search
  Path-finding in large graphs over complex states
  Multiple states implies dynamics of some sort

Last time: planning as heuristic search

- STRIPS is flawed, but contains the seeds of a great idea
- Knowledge base $\Rightarrow$ world state
- Logical theory changes from state to state???
- Can we describe this change within the theory itself?

Actually, the idea is much older than STRIPS.

Aside

Trouble with describing change in logic using implication

- Implication: $Cause \rightarrow Effect$
- Equiv. to $\neg Cause \vee Effect$
- Still holds when $Cause = false$ and $Effect = true$
- So, can have *Effect* without *Cause*
- Let's rule this out by disallowing this row in truth table
- Get $Cause \leftrightarrow Effect$
- Can no longer distinguish between cause and effect
- The whole thing loses meaning

Planning: more than search

- McCarthy, J. and Hayes, P.J., 1969. Some philosophical problems from the standpoint of AI, Machine Intelligence (Meltzer B. and Michie D., eds.), vol. 4.
  `http://www-formal.stanford.edu/jmc/mcchay69.pdf`

  Fundamental work on the philosoply of AI; introduces **situation calculus**. Recommended reading of general interest.

- Reiter, R., 2001. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press.

  Combines many many refinements to McCarthy's proposal into a robust formalism

Situation Calculus

The **situation calculus** is a dialect of FOL for representing dynamically changing worlds in which all changes are the result of named actions.

- Many-sorted logic: has several domains, one for each sort. Each term is interpreted only within its own sort.
- Situation calculus has sorts **action**, **situation**, and **object**

Actions and Situations

Actions: denoted by function terms of sort *action*, e.g.,

- $put(x, y)$ — put thing x on top of thing y
- $walk(location)$ — walk to location *location*
- $pickup(r, x)$ — robot r picks up thing x

Situations: world histories built using specialized symbols S_0 and $do(\cdot, \cdot)$

- S_0 — a constant, always denotes the initial situation
- $do(a, s)$ — a situation that results from doing action a in situation s

Example: $do(put(A, B), do(put(B, C), S_0))$

Fluents

- Already understand: predicates in FOL
- **Fluents**: predicates whose values may **vary from situation to situation**
- Syntactically, a fluent is a predicate whose last argument is of sort *situation*

  Example: $Holding(r, x, s)$ — robot r is holding thing x in situation s

  Can also say things like $\neg Holding(r, x, s) \wedge Holding(r, x, do(pickup(r, x,), s))$
- Note: there is no distinguished "current" situation. A sentence can talk about many different situations, past, present, and future.

Poss

Also:

- A distinguished predicate symbol $Poss(a, s)$ is used to state that action a is *legal to carry out* in situation s.

  $Poss(pickup(r, x), S_0)$ — it is possible for robot r to pick up thing x in the initial situation

Preconditions and effects

- It is necessary to include in a KB not only facts about the initial situation, but also about **world dynamics**
- I.e., a formal account of how and why things which are true (false) in one situation become false (true) in the next
- Action preconditions and action effects

Preconditions and effects

- Actions typically have **preconditions**: what needs to be true for the action to be performed

$$\begin{aligned} Poss(pickup(R, X), S) \leftrightarrow \forall Z\, [\neg Holding(R, Z, S)] \\ \wedge \neg Heavy(X) \wedge NextTo(R, X, S) \end{aligned}$$

  Free variables are assumed to be universally quantified from outside

$$Poss(repair(R, X), S) \leftrightarrow HasGlue(R, S) \wedge Broken(X, S)$$

These are called *precondition axioms*

Preconditions and effects

- Actions typically have **effects**: the fluents that change as the result of performing the action

$$Fragile(X) \rightarrow Broken(X, do(drop(R, X), S))$$

$$\neg Broken(X, do(repair(R, X), S))$$

These are called *effect axioms*

The Frame Problem

- Effect axioms only describe what *changes*. To fully describe how the world works, need to specify what fluents are *unaffected* by what actions.

$$Colour(X, C, S) \rightarrow Colour(X, C, do(drop(R, X), S))$$

$$\neg Broken(X, S) \wedge [X \neq Y \vee \neg Fragile(X)] \rightarrow \neg Broken(X, do(drop(R, Y), S))$$

These are called *frame axioms*

Frame Problem

- **Problem!** There will always be a **vast** number of frame axioms.
  An object's colour is unaffected by picking things up, opening a door, calling a friend, turning on a light, weather patterns in China, etc.
- In building a KB, need to include $2 \times |A| \times |F|$ facts about what doesn't change, and then reason efficiently with them

The Frame Problem

- Suppose we have a complete set of effect axioms (non-trivial dynamics, written down by a specialist)
- Can we maybe generate the frame axioms mechanically?
- And, hopefully, in some compact form
- Yes, under some assumptions (later)

The Projection Task

What is the situation calculus good for?

- **Projection task**: *Given a sequence of actions, determine what would be true in the situation that results from performing that sequence*
- More formally: Suppose $R(S)$ is a formula with a free situation variable S. To find out if $R(S)$ would be true after performing actions $\langle a_1, \ldots, a_n \rangle$ in the initial situation, we determine whether or not

$$KB \models R(do(a_n, do(a_{n-1}, \ldots, do(a_1, S_0) \ldots)))$$

- Example: using axioms above, it follows that $\neg Broken(b, S)$ would hold after executing the sequence

$$\langle pickup(a), pickup(B), drop(B), repair(B), drop(A) \rangle$$

Legality / Executability

- Projection does not test for whether the sequence of actions is legal (wrt precondition axioms)
- We call a situation *legal* is it is the initial situation or the result of performing an action whose preconditions are satisfied starting in a legal situation
- **Legality task**: task of determining whether a sequence of actions leads to a legal situation
- More formally: To find out if the sequence $\langle a_1, \ldots, a_n \rangle$ can be legally performed in the initial situation, we determine whether or not

$$KB \models Poss(a_i, do(a_{i-1}, \ldots, do(a_1, S_0) \ldots))$$

for all $1 \leq i \leq n$.

Limitations of Situation Calculus

(as presented)

- No time: cannot talk about how long actions take, or when they occur
- Only known actions: no hidden exogenous actions, no unnamed events
- No concurrency
- Only discrete situations: no continuous actions, like pushing an object from A to B
- Only hypotheticals: cannot say that an action has occurred or will occur
- Only primitive actions: no internal structure to actions, conditional actions, iterative actions, etc.

A Solution to Frame Problem

- Suppose there are two positive effect axioms for the fluent *Broken*:

$$Fragile(X) \rightarrow Broken(X, do(drop(R, X), S))$$
$$NextTo(B, X, S) \rightarrow Broken(X, do(explode(B), S))$$

- Can equivalently rewrite these as

$$\begin{aligned} &\exists R\{A = drop(R, X) \wedge Fragile(X)\} \\ \vee\, &\exists B\{A = explode(B) \wedge NextTo(B, X, S)\} \\ &\rightarrow Broken(X, do(A, S)) \end{aligned}$$

A Solution to Frame Problem

- Similarly for the negative effect axiom:

$$\neg Broken(X, do(repair(R, X), S))$$

can be rewritten as

$$\exists R\{A = repair(R, X)\} \rightarrow \neg Broken(X, do(A, S))$$

Note how nice the right-hand sides are starting to look. This is called the *normal form for effect axioms*. (One positive NF axiom, one negative NF axiom.)

A Solution to Frame Problem

- In general, for any fluent F, we can rewrite **all** the effect axioms as two formulas of the form

$$P_F(\bar{X}, A, S) \rightarrow F(\bar{X}, do(A, S)) \tag{1}$$

$$N_F(\bar{X}, A, S) \rightarrow \neg F(\bar{X}, do(A, S)) \tag{2}$$

- Next, make a *completeness assumption* regarding these:
  Assume that (1) and (2) characterize ALL the conditions under which an action A changes the value of fluent F
- Formally, this is captured by *explanation closure axioms*:

$$\neg F(\bar{X}, S) \wedge F(\bar{X}, do(A, S)) \rightarrow P_F(\bar{X}, A, S) \tag{3}$$

$$F(\bar{X}, S) \wedge \neg F(\bar{X}, do(A, S)) \rightarrow N_F(\bar{X}, A, S) \tag{4}$$

A Solution to Frame Problem

- In fact, axioms (3) and (4) are, in fact, disguised versions of the frame axioms!

$$\neg F(\bar{X}, S) \land \neg P_F(\bar{X}, A, S) \rightarrow \neg F(\bar{X}, do(A, S))$$

$$F(\bar{X}, S) \land \neg N_F(\bar{X}, A, S) \rightarrow F(\bar{X}, do(A, S))$$

A Solution to Frame Problem

- Need some additional assumptions:
  - Integrity of effect axioms: can't have $P_F(\bar{X}, A, S)$ and $N_F(\bar{X}, A, S)$ hold at the same time—this must be provable from KB
  - Unique names for actions—some standard axioms

- With these and some effort, it can be shown that, in the models of the KB, the axioms (1)–(4) are logically equivalent to

$$F(\bar{X}, do(A, S)) \leftrightarrow P_F(\bar{X}, A, S) \vee F(\bar{X}, S) \wedge \neg N_F(\bar{X}, A, S)$$

This is called the **successor state axiom** for F.

Example

Example of a SSA:

$$
\begin{aligned}
Broken(X, do(A, S)) \leftrightarrow\ & \exists R\{A = drop(R, X) \land Fragile(X)\} \\
& \lor \exists B\{A = explode(B) \land NextTo(B, X, S)\} \\
& \lor Broken(X, S) \land \neg\exists R\{A = repair(R, X)\}
\end{aligned}
$$

A simple solution to the frame problem

- This simple solution is due to Raymond Reiter yields the following axioms:
  - one SSA per fluent
  - one precondition axiom per action
  - unique name axioms for actions
- Note: the length of a SSA is roughly proportional to the number of actions which affect the truth value of the fluent
- The conciseness of the solution relies on quantification over actions, the assumption that relatively few actions affect each fluent, and the completeness assumption (also, assumption of determinism)

End of Lecture

- Next time: **GOLOG and Planning in Situation Calculus**