Introduction to the Situation Calculus

Maurice Pagnucco
School of Computer Sc. & Eng.
University of New South Wales
NSW 2052, AUSTRALIA

morri@cse.unsw.edu.au
http://www.cse.unsw.edu.au/~morri/

NB: Many examples from: R. Brachman and H. J. Levesque, Knowledge Representation, 2004.

Overview

- Situation Calculus
- States/Situations
- Domain Constraints
- Actions
- The Frame Problem
- Solving the Frame Problem
- Summary

Reasoning About Action

- One method to reason about action is to simply change the agent's knowledge base
- Erase some sentence(s) that should no longer be true and add sentences that will now be true (i.e., after performing action)
- However, we can only answer questions about the current state
- It will not be possible to reason about past or future states
- On the other hand, if all we want to do is reason about which actions to perform, this may be a viable approach

Modelling Domains and Actions

- Aspects we need to consider:
  - The state of the world
  - Actions that change state of the world and what changes they effect
  - Constraints on legal scenarios (won't deal much with these in this lecture)
  - Can you think of anything else?

Situation Calculus

- The situation calculus is a way of describing change in first-order logic
- In simple terms it may be viewed as a dialect of FOL
- Terms
  - actions
  - situations
- Fluents—predicates or functions whose values may vary

State of the World

- Method 1:

  $on(C, A, S_1)$
  $on(A, Table, S_1)$
  $on(B, Table, S_1)$
  $clear(B, S_1)$
  $clear(C, S_1)$
- **Note:** we reify states (i.e., make them entities in our formalisation)
- Another common way using the situation calculus is as follows
- Method 2:

  $holds(on(C, A), S_1)$
  $holds(on(A, Table), S_1)$
  $holds(on(B, Table), S_1)$
  $holds(clear(B), S_1)$
  $holds(clear(C), S_1)$

Actions

- Actions are named
  - $put(x, y)$ — put object x on top of object y
  - $move(x, y, z)$ — move block x from y to z
  - $clear(x)$ — clear x

Situations

- Situation — a snapshot of the world at a particular point in time
- Alternate view — world histories
-
  - $S_0/init$ — initial situation (no actions have been performed)
  - $do(a, s)$ — situation resulting from performing action a in situation s
- For example, $do(put(A,B), do(put(B,C), S_0))$
  Situation resulting from putting block B on block C in the initial situation and then placing block A on block B

Fluents

- Predicates and functions whose values may vary from situation to situation
- For example, $\neg Broken(x, s) \wedge Broken(x, do(drop(r, x), s))$

Preconditions

- Special predicate $Poss(a, s)$ denotes that action a may be performed in state s
- For example, $Poss(pickup(r, x), s) \equiv \forall z \neg Holding(r, z, s) \wedge \neg Heavy(x) \wedge NextTo(r, x, s)$

Effects

- Actions can have positive effects
  $Fragile(x) \supset Broken(x, do(drop(r, x), s))$
- and negative effects
  $\neg Broken(x, do(repair(r, x), s))$

Domain Constraints

- Also known as state constraints
- True at all (legal) states even though they involve state-dependent relations

x is on the table iff it is not on top of another block

$$on(x, Table, s) \equiv \neg \exists y (on(x, y, s) \wedge y \neq Table)$$

x is clear iff there is no block on top of it

$$clear(x, s) \equiv \neg \exists y\, on(y, x, s)$$

If y is a block and there is another block on it, then y is not clear

$$on(x, y, s) \wedge \neg (y = Table) \supset \neg clear(y, s)$$

etc.

The Frame Problem

- Action descriptions are not complete:
  - They describe what changes BUT do **not** specify what stays the same!
- The (famous) Frame Problem:

  The problem of characterising those aspects of the state description that are not changed by an action
- One solution — Frame Axioms

  Moving an object does not change its colour
  $Colour(x, c, s) \supset Colour(x, c, do(put(x, y), s))$

  Fragile things do not break
  $\neg Broken(x, s) \wedge (x \neq y \vee \neg Fragile(x)) \supset \neg Broken(x, do(drop(r, y), s)$
- Since actions often leave most fluents unchanged, many frame axioms may be required

Ramification Problem

- What are the ramifications (direct and indirect effects) of performing an action
$$\neg clear(b,\ do(move(c,\ a,\ b),\ S_0))$$
- Recent approaches have investigated the use of explicit notions of causality in an attempt to solve this problem efficiently

Qualification Problem

- What qualifications (preconditions) do we require in specifying actions and their effects
- Trying to specify exactly under which conditions an action has a particular effect is very difficult (in principle, the list of preconditions can be vast)

What counts as a solution to the frame problem?

- Once we have described the actions of a system, we would like a systematic method for automatically generating frame axioms
- Preferably, the representation should be concise
- Reasons:
  - Require frame axioms for reasoning
  - They are not entailed by other axioms
  - Reduce possibility of errors in determining frame axioms
  - Can easily update frame axioms if additional effects are specified

Projection

- Determining what is true in the situation resulting from the performing of a sequence of actions $a_1, \ldots, a_n$
- Suppose we gather all the axioms above in a sentence F. To determine whether a formula ϕ is true after performing the sequence of actions $a_1, \ldots, a_n$, we need to determine
$$\Gamma \models \phi(do(a_n,\ do(a_{n-1},\ \ldots,\ do(a_1,\ S_0)\ldots)))$$

Legality

- However, we don't know whether the sequence of actions $a_1, \ldots, a_n$ can be performed
- A situation is legal iff:
  - $Legal(S_0)$ — it is the initial situation
  - $Legal(do(a,\ s)) \equiv Legal(s) \wedge Poss(a,\ s)$ — it results from performing the action in a legal situation where its precondition is satisfied
- Adding these axioms to Γ, we can determine whether a sequence of actions can be performed by showing that they lead to a legal situation
$$\Gamma \models Legal(do(a_n,\ do(a_{n-1},\ \ldots,\ do(a_1,\ S_0)\ldots)))$$

Normal form for effect axioms

- Given positive effect axioms for fluent *Broken*:
  $Fragile(x) \supset Broken(x,\ do(drop(r,\ x),\ s))$
  $NextTo(b,\ x,\ s) \supset Broken(x,\ do(explode(b),\ s))$
- Rewrite them:
  $\exists r\{a = drop(r,\ x) \land Fragile(x)\}\lor$
  $\exists b\{a = explode(b) \land NextTo(b,\ x,\ s)\} \supset$
  $Broken(x,\ do(a,\ s))$
- Negative effect axiom:
  $\neg Broken(x,\ do(repair(r,\ x),\ s))$
- Rewrite as:
  $\exists r\{a = repair(r,\ x)\} \supset \neg Broken(x,\ do(a,\ s))$
- These formulae or of the form:
  $P_F(x_1,\dots,x_n,\ a,s) \supset F(x_1,\dots,x_n,do(a,\ s))$
  $N_F(x_1,\dots,x_n,\ a,s) \supset \neg F(x_1,\dots,x_n,do(a,\ s))$

Explanation Closure

- **Assumption:** The previous two formulae characterise the only way in which a fluent may change
- Explanation Closure Axioms
  - $\neg F(\mathbf{x},\ s) \land F(\mathbf{x},\ do(a,\ s)) \supset P_F(\mathbf{x},\ a,\ s)$
  - $F(\mathbf{x},\ s) \land \neg F(\mathbf{x},\ do(a,\ s)) \supset N_F(\mathbf{x},\ a,\ s)$
- Disguised frame axioms:
  - $\neg F(\mathbf{x},\ s) \land \neg P_F(\mathbf{x},\ a,\ s) \supset \neg F(\mathbf{x},\ do(a,\ s))$
  - $F(\mathbf{x},\ s) \land \neg N_F(\mathbf{x},\ a,\ s) \supset F(\mathbf{x},\ do(a,\ s))$

Successor State Axioms

- Additional axioms:
  - Integrity of effect axioms
    $\neg\exists \mathbf{x},\ a,\ s\ P_F(\mathbf{x},\ a,\ s) \land N_F(\mathbf{x},\ a,\ s)$
  - Unique names for actions
    $A(x_1,\dots,x_n) = A(y_1,\dots,y_n) \supset (x_1 = y_1) \land \dots \land (x_n = y_n)$
    $A(x_1,\dots,x_n) \neq B(y_1,\dots,y_m)$ for distinct A and B
- Together, axioms on last three slides equivalent to successor state axiom for F:
  $F(\mathbf{x},\ do(a,\ s)) \equiv P_F(\mathbf{x},\ a,\ s) \lor (F(\mathbf{x},\ s) \land \neg N_F(\mathbf{x},\ a,\ s))$
- $Broken(x,\ do(a,\ s)) \equiv$
  $\exists r\{a = drop(r,\ x) \land Fragile(x)\}\lor$
  $\exists b\{a = explode(b) \land NextTo(b,\ x,s)\}\lor$
  $Broken(x,\ s) \land \neg\exists r\{a = repair(r,\ x)\}$

What we cannot do

- Explicit time
- Exogenous actions
- Concurrent actions
- Continuous actions
- Complex actions
- ...

Summary

- Reasoning about actions is a very interesting area of artificial intelligence and often makes use of nonmonotonic reasoning techniques
- We have seen that a number of challenging problems arise that we must deal with in order to reason effectively
- One of the problems, however, is the possible proliferation of axioms
- The search continues for a concise solution to the frame problem (and associated problems)
- Other formalisms include the event calculus, $\mathcal{A}$ languages, features and fluents, fluent calculus
- Current research: causal approaches, cognitive robotics, planning (an area in its own right)