Artificial Intelligence
The Situation Calculus

Outline

- Reasoning about States and Actions
- Some Difficulties
- Generating Plans
- Additional Readings and Discussion

21.1 Reasoning about States and Actions

- To investigate feature-based planning methods much more thoroughly, richer language to describe features and the constraints among them will be introduced.

- Generally, a goal condition can be described by any wff in the predicate calculus, and we can determine if a goal is satisfied in a world state described by formulas by attempting to prove the goal wff from those formulas.

Situation calculus (1/3)

- A predicate calculus formalization of states, actions, and the effects of actions on states.

- Our knowledge about states and actions as formulas in the first-order predicate calculus
- Then use a deduction system to ask questions such as “Does there exist a state to satisfy certain properties, and if so, how can the present state be transformed into that state by actions”

  $\Rightarrow$ A plan for achieving the desired state.

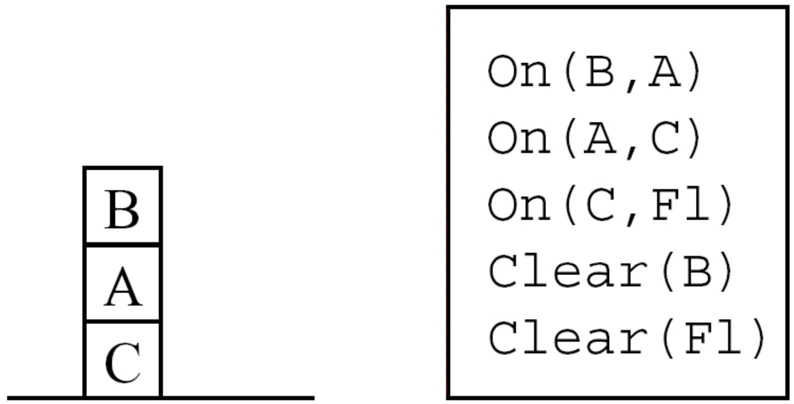
Situation calculus (2/3)

- The situation calculus was used in some early AI planning systems. $\Rightarrow$ it does not used nowadays.
- However, the formalism remains important for exposing and helping to clarify conceptual problems.

Situation calculus (3/3)

- In order to describe states in the situation calculus, we *reify* states.
- States can be denoted by constant symbols (S0, S1, S2, …), by variables, or by functional expressions.
- *Fluents*: the atomic wff can denote relations over states.

Example (Figure 21.1)



- First-order predicate calculus

$$On(B,A) \wedge On(A,C) \wedge On(C,F1) \wedge Clear(B) \wedge \ldots$$

- True statement

$$On(B,A,S0) \wedge On(A,C,S0) \wedge On(C,F1,S0) \wedge Clear(B, S0)$$

- Prepositions true of all states

$$(\forall x,y,s)[On(x,y,s) \wedge \neg(y=F1) \supset \neg Clear(y,s)]$$

$$\text{And} \quad (\forall s) Clear(F1,s)$$

To represent actions and the effects (1/2)

1. Reify the action
   – Actions can be denoted by constant symbols, by variables, or by functional expressions
   – Generally, we can represent a family of move actions by the *schema*, move(x,y,z), where x, y, and z are *schema variables*.
2. Imagine a function constant, do, that denotes a function that maps actions and states into states.
   – do(α,σ) denotes a function that maps the state-action pair into the state obtained by performing the action denoted by α in the state denoted by σ.

To represent actions and the effects (2/2)

3. Express the effects of actions by wffs.
   - There are two such wffs for each action-fluent pair.
   - For the pair {On, move}.

$$[On(x,y,s) \wedge Clear(x,s) \wedge Clear(z,s) \wedge (x \neq z) \supset On(x,z,do(move(x,y,z),s))]$$

positive effect axiom

preconditions

And $$[On(x,y,s) \wedge Clear(x,s) \wedge Clear(z,s) \wedge (x \neq z) \supset \neg On(x,z,do(move(x,y,z),s))]$$

negative effect axiom

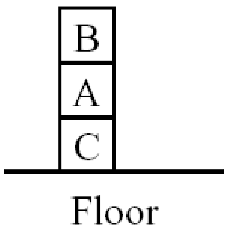
consequent

Effect axioms

- We can also write effect axioms for the {Clear, move} pair.

$$[On(x,y,s) \wedge Clear(x,s) \wedge Clear(z,s) \wedge (x \neq z) \wedge (y \neq z)$$
$$\supset Clear(y,do(move(x,y,z),s))]$$
$$[On(x,y,s) \wedge Clear(x,s) \wedge Clear(z,s) \wedge (x \neq z) \wedge (z \neq F1)$$
$$\supset \neg Clear(z,do(move(x,y,z),s))]$$

- The antecedents consist of two parts
  - One part expresses the preconditions under which the action can be executed
  - The other part expresses the condition under which the action will have the effect expressed in the consequent of the axiom.

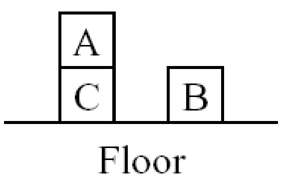



```
On(B,A,S0)
On(A,C,S0)
On(C,Fl,S0)
Clear(B,S0)
Clear(Fl,S0)

S0

move(B,A,Fl)

S1 = do(move(B,A,Fl),S0)
```



Inferred using effect axioms:

```
On(B,Fl,do(move(B,A,Fl),S0))
¬On(B,A,do(move(B,A,Fl),S0))
Clear(A,do(move(B,A,Fl),S0))
```

Inferred using frame axioms:

```
On(A,C,do(move(B,A,Fl),S0))
On(C,Fl,do(move(B,A,Fl),S0))
Clear(B,do(move(B,A,Fl),S0))
```

True in all states:

```
(∀s)Clear(Fl,s)
```

Figure 21.2 Mapping a State-Action Pair into a State

21.2.1 Frame Axioms (1/3)

- Not all of the statements true about state do(move(B,A,F1),S0) can be inferred by the effects axioms.
  - Before the move, such as that C was on the floor and that B was clear are also true of the state after the move.
- In order to make inferences about these constancies, we need *frame axioms* for each action and for each fluent that doesn’t change as a result of the action.

21.2.1 Frame Axioms (2/3)

- The frame axioms for the pair, {(move, On)}

$$[On(x,y,s) \wedge (x \neq u)] \supset On(x,y,do(move(u,y,z),s))$$

$$(\neg On(x,y,s) \wedge [(x \neq u) \vee (y \neq z)] \supset \neg On(x,y,do(move(u,v,z),s))$$

positive frame axioms

negative frame axioms

- The frame axioms for the pair, {move, Clear}

$$Clear(u,s) \wedge (u \neq z)] \supset Clear(u,do(move(x,y,z),s))$$

$$\neg Clear(u,s) \wedge (u \neq y) \supset \neg Clear(u,do(move(x,y,z),s))$$

21.2.1 Frame Axioms (3/3)

- Frame axioms are used to prove that a property of a state remains true if the state is changed by an action that doesn't affect that property.

- *Frame problem*
  - The various difficulties associated with dealing with fluents that are not affected by actions.

21.2.2 Qualifications

- The antecedent of the transition formula describing an action such as move gives the preconditions for a rather idealized case.
  - To be more precise, adding other qualification such as $\neg$Too_heavy(x,s), $\neg$Glued_down(x,s), $\neg$Armbroken(s), …
- *Qualification problem*
  - The difficulty of specifying all of the important qualifications.

21.2.3 Ramifications

- *Ramification problem*
  - Keeping track of which derived formulas survive subsequent state transitions.

21.3 Generating Plans

- To generate a plan that achieves some goal, $\gamma(s)$,
  - We attempt to prove $(\exists s)\ \gamma(s)$
  - And use the answer predicate to extract the state as a function of the nested actions that produce it.

Example (Figure 21.1) (1/2)

- We want a plan that gets block *B* on the floor from the initial state, *S0*, given in Figure 21.1.
- Prove $(\exists s)$ On(B, F1, s)
- We will prove by resolution refutation that the negation of $(\exists s)$ On(B, F1, s), together with the formulas that describe *S0* and the effects of move are inconsistent.
  – Use an answer predicate to capture the substitutions made during the proof.

Example (Figure 21.1) (2/2)

On(A,C,S0)

On(C,F1,S0)

Clear(B,S0)

Clear(F1,S0)

$[On(x,y,s) \wedge Clear(x,s) \wedge Clear(z,s) \wedge (x \neq z) \supset$
$On(x,z,do(move(x,y,z),s))]$

Difficulties

- If several actions are required to achieve a goal, the action functions would be nested.
- The proof effort is too large for even simple plan.

Additional Readings

- [Reiter 1991]
  - Successor-state axiom
- [Shanahan 1997]
  - Frame problem
- [Levesque, et al. 1977]
  - GOLOG(alGol in LOGic)
- [Scherl & Levesque 1993]
  - Robot study in Perception Robot Group