Reasoning about action & change

- Frame problem and related problems
- Situation calculus, event calculus, fluent calculus
- Planning
- Relation with non-monotonic reasoning

158

Frame problem(s)

- Persistence problem
  - Assumption of inertia, minimal change
- Qualification problem
  - What are the preconditions of a successful performance of an action?
- Ramification problem
  - What *does* change as an (indirect) result of an action?

159

Persistence problem

- Assumption of *inertia / minimal change* is *epistemological* rather than *ontological:*
  - Not so much about the *nature* of the world but about our *description* of the world
  - It is a knowledge-technological device *to keep descriptions of (non)effects manageable*
    - 'Dynamic version' of *Closed World Assumption*

160

Situation calculus (McCarthy)

- Predicate calculus framework to reason about actions and change
- Situation: actual or hypothetical state of the world at a particular time
- Result(A, s): the new situation after action A has been performed in situation s
- Holds(p, s): the property p is true in situation s

161

Situation calculus, axiomatisation
Lin & Reiter (Herbrand situations)

- S0 ≠ Result(A, s)
- Result(A, s) = Result(A', s') □ A = A' □ s = s'

Induction rule:

□(S0)
□(s) □ □A[□(Result(A, s))]
____
□s[□(s)]

162

Situation calculus

- Some authors don't like the concept of a Herbrand situation, which amounts to a *situation as a sequence of actions performed in an initial situation*
- They prefer to view situations as states, thus reifying the abstract notion of state, including them in our conceptualisation of the world □ i.e. put states into *object language*

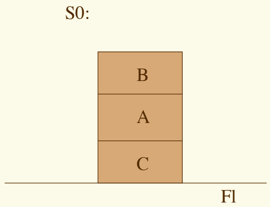
163

Situation calculus, state semantics

- *States* are mappings from properties to truth-values (or 'fluents' to their values)
- *States* are characterized by:
  - *Uniqueness* property
    ∀r∀r'[∀p[Holds(p,r)≡Holds(p,r')]⇒ r=r']
  - *Existence* property
    ∀r∀p∃r'[(Holds(p,r)≡¬Holds(p,r'))∧
    ∀p'[p≠p' ⇒ (Holds(p',r)≡Holds(p',r'))]]

Situation calculus, example

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



On(x, y, s) abbreviates Holds(On(x, y), s)
Clear(x, s) abbreviates Holds(Clear(x), s)

Sit. Calc., example (ctd)

- Some propositions true of all states:
  - (∀x,y,s)[On(x,y,s) ∧ ¬(y = Fl) ⇒ ¬Clear(y,s)]
  - (∀s) Clear(Fl,s)
- Derived assertions:
  - ¬Clear(A,S0)
  - Clear(Fl,S0)

Actions in the situation calculus

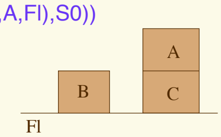
- Recipe for representing actions and their effects:
  - Reify the actions
  - Use function constant 'do' denoting a function mapping actions and states to states
  - Express the (positive and negative) effects of actions by wffs

Actions in situation calculus

- For example:
  - [On(x,y,s) ∧ Clear(x,s) ∧ Clear(z,s) ∧ (x≠z) ⇒ On(x,z,do(move(x,y,z),s))]
  - [On(x,y,s) ∧ Clear(x,s) ∧ Clear(z,s) ∧ (x≠z) ⇒ ¬On(x,y,do(move(x,y,z),s))]
  - [On(x,y,s) ∧ Clear(x,s) ∧ Clear(z,s) ∧ (x≠z) ∧ (y≠z) ⇒ Clear(y,do(move(x,y,z),s))]
  - [On(x,y,s) ∧ Clear(x,s) ∧ Clear(z,s) ∧ (x≠z) ∧ (z≠Fl) ⇒ ¬Clear(z,do(move(x,y,z),s))]

Actions in situation calculus

- Derived results: after applying move(B,A,Fl) it holds that:
  - 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))



Frame axioms in situation calculus

- Frame axioms pertain to non-effects
  - [On(x,y,s) □ (x≠u)] □ On(x,y,do(move(u,v,z),s))
  - [□On(x,y,s) □ [(x≠u) (y≠z)] □ □On(x,y,do(move(u,v,z),s))
  - Clear(u,s) □ (u≠z) □ Clear(u,do(move(x,y,z),s))
  - □Clear(u,s) □ (u≠y) □ □Clear(u,do(move(x,y,z),s))

170

The Frame Problem

- 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
- In principle a *pair* of frame axioms is needed for *every combination of fluent and action*
- ***This is unmanageable in practice!!***

171

Relation with non-monotonic reasoning

- Historically, the study of the frame problem gave rise to the field of non-monotonic reasoning (NMR)
- E.g., persistence in terms of *default rule*

$$\frac{\text{Holds}(p(x),s) : \neg\text{Ends}(p(x), \text{Result}(A,s))}{\text{Holds}(p(x), \text{Result}(A,s))}$$

172

Solution in situation calculus

- See slides Hölldobler & Thielscher

173

Deliberative / Planning Systems

- *Planning = knowing what to do, what action to perform*
- Origins in Newell & Simon's GPS
- STRIPS planning system (Fikes & Nilsson)
- *`planning from first principles'*: for every goal an entirely new plan / program

176

STRIPS-style Planning System

- Needed:
  - Symbolic model of agent's environment
  - Symbolic specification of actions available to the agent
  - Planning algorithm: *input*: representation environment, set of action specs, repr. goal state; *output*: plan / program to achieve goal
- Symbolic AI / logicist philosophy!

177

Planning: STRIPS

- Given a goal wff □, try to find a sequence of actions that produces a world state described by some state description □ such that □ ⊨ □
- *Forward* search methods
- *Backward* search methods

178

STRIPS Planning: forward search

- STRIPS operators are used to transform a before-action state description into an after-action state description
- A STRIPS operator consists of:
  - A set PC of ground literals, the *preconditions*
  - A set D of ground literals, the *delete list*
  - A set A of ground literals, the *add list*
- 'Built-in' persistence/inertia: *what is not deleted explicitly, stays in*

179

STRIPS Planning: forward, ctd.

For example:

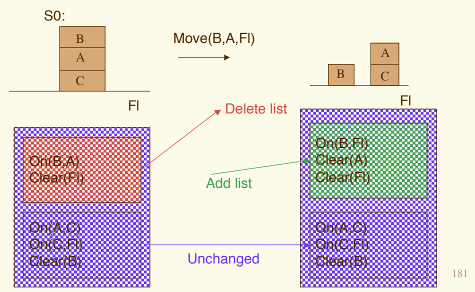
(for every ground instance of x, y, z:)

move(x,y,z):

- PC: On(x,y) □ Clear(x) □ Clear(z)
- D: Clear(z), On(x,y)
- A: On(x,z), Clear(y), Clear(Fl)

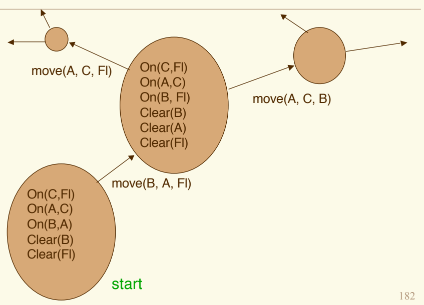
180

STRIPS planning: forward



181

STRIPS: forward search

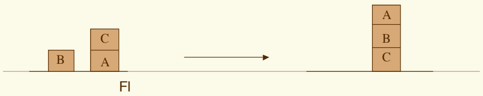


182

Problems with forward planning

- *Breadth-first* not practically feasible
- Alternative: if goal is conjunction of literals. try *divide-and-conquer* heuristic □ *'recursive STRIPS'*:
  - *Subsequently achieve conjuncts* in goal by forward search (a kind of search space 'island hopping')
  - *Kind of 'Depth-first'* approach (*satisfying subsequent goals completely in isolation*)
  - leads to problems like the *Sussman anomaly*

183



- Goal condition: On(B,C) $\wedge$ On(A,B)
- Suppose first On(B,C) is selected to be achieved: move(C,A,Fl) ; move(B,Fl,C)
- Then to achieve On(A,B) start with move(B,C,Fl) ; move(A,Fl,B)
- Next to obtain On(B,C) again, we do move(A,B,Fl) ; move(B,Fl,C): *repetition!!*

The Sussman Anomaly

184

Solution: going backward?

- \`Narrow' (*depth-first*) treatment of goals in isolation leads to problems such as the Sussman anomaly
- \`*Breadth-first*' solution would solve this, but is not practically feasible
- Possible solution: use *backward search, starting with the goal state* (comprising the *entire* goal as opposed to earlier):
  - *going backward breadth-first is (likely to be more) feasible than in forward direction:*
    - since *typically* many fewer conjuncts in goal wff than in initial state description!

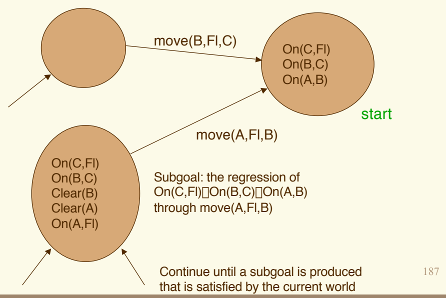
185

STRIPS: backward search

- *Regression* procedure:
  - Start with goal state
  - *Regress* goal wffs through STRIPS rules to produce *subgoal* wffs
  - *Regression of φ through α = weakest formula ψ s.t. if ψ holds in a state before α is applied then φ will hold afterwards*
    - Cf. Dijkstra's weakest precondition calculus, dynamic logic: $[\alpha]\varphi$ = wp(α, φ)

186

STRIPS: backward search



187