UCPOP

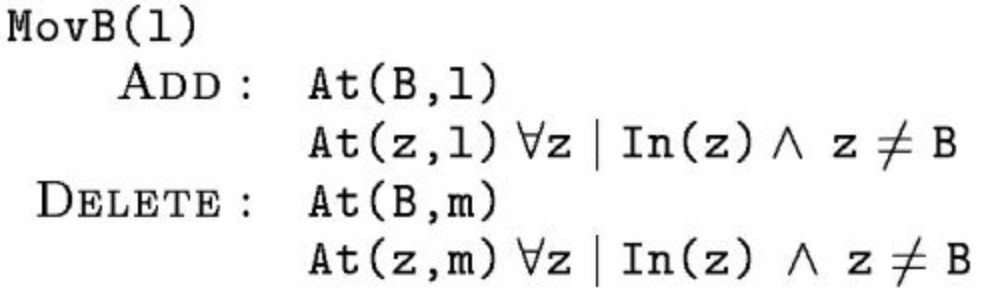
Gage Birchmeier

Background

- Most planners at the time had one of two limitations:
  - Only using STRIPS notation
  - Output is a totally ordered plan
- Learned previously about POP which uses STRIPS to give a partially ordered plan

Action Description Language (ADL)

- Expands on STRIPS to allow more features
- Existential (∃) and Universal (∀) Quantifiers
- Equality (=) and inequality (≠)



Introducing UCPOP

- Universal Conditional Partial Order Planner
- Uses ADL instead of STRIPS
- Produces a nonlinear plan
- Can solve more problems than POP because it uses ADL
- Allows for conditionals and quantifiers

How it works

- Very similar to POP algorithm from last time
- We must search through ground clauses to find goals, subgoals, and protect causal links

1. Terminate if there's no more goals
2. Select a goal
3. Find an operator that gives the effect we're looking for
4. Generate new subgoals
5. Protect causal links (Promotion, Demotion, *Separation*)
6. Recurse

Expanding quantifiers to get ground clauses

- When finding operators, subgoals, we need ground clauses rather than quantifiers
- The universal base Υ gives us a way to expand quantifiers

$$\{\exists x Above(x, A), \forall \exists On(y, w)\}$$

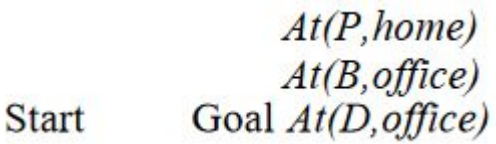
- We can apply Υ to this to receive the following:

$$\{\exists x Above(x, A),\ \exists w' On(A, w'),\ \exists w' On(B, w')\}$$

Separation

- Another way besides promotion and demotion to protect causal links
- Given causal link $S_i \xrightarrow{e_i,q} S_j$ with threat S_k, add $\{S_i < S_k < S_j\}$ as a constraint
- Two different ways to separate
  - Add more constraints on existential variables that cause the variables to not interfere thus removing the threat
  - Create a new goal that will separate the conflicting variables when resolved

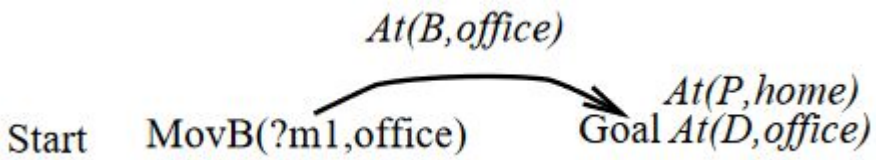
Briefcase Problem



- Two locations, Home (H) and Office (W)
- Three items, a Paycheck (P) and a dictionary (D), and a Briefcase (B)
- All three items start at home, with the Paycheck inside the Briefcase
- Operators:
  - MovB(M, L) which moves the briefcase and everything in it from location M to the location L
  - PutIn(I, L) which puts item I in the briefcase at the location L
  - TakeOut(I) which takes item I out of the briefcase
- Goal: leave the paycheck at home and take the briefcase and dictionary to the office

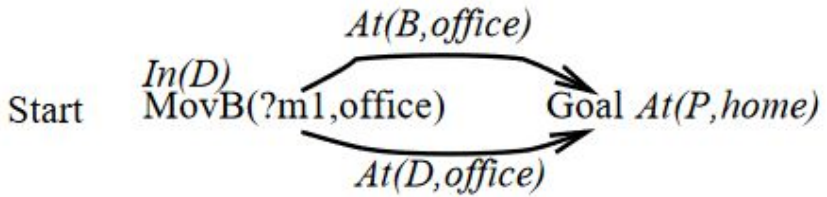
Briefcase Problem (Cont.)

- We select At(B, Office) as the first goal
- Then use MovB(?m1, office) to move the briefcase to the office



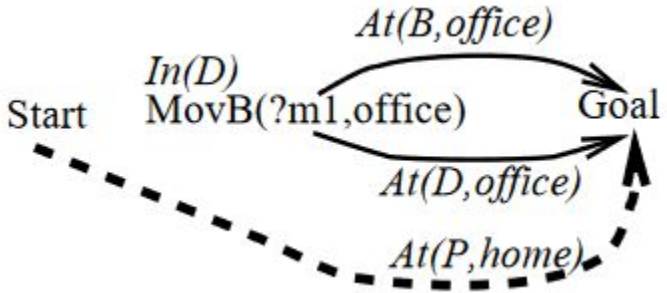
Briefcase Problem (Cont.)

- Next we select At(D, office) as a goal
- We recognize we can solve this with MovB, assuming the dictionary is in the briefcase
- In(D) gets added as a subgoal before MovB



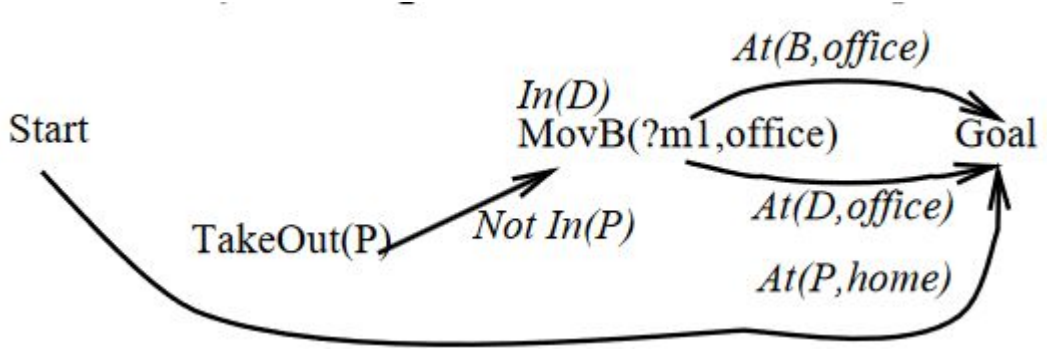
Briefcase Problem (Cont.)

- Next we select At(P, home) as a goal
- This is part of the initial state of the system
- However this causal link is threatened by the MovB step since the paycheck is in the briefcase
- *All items within the briefcase are moved*



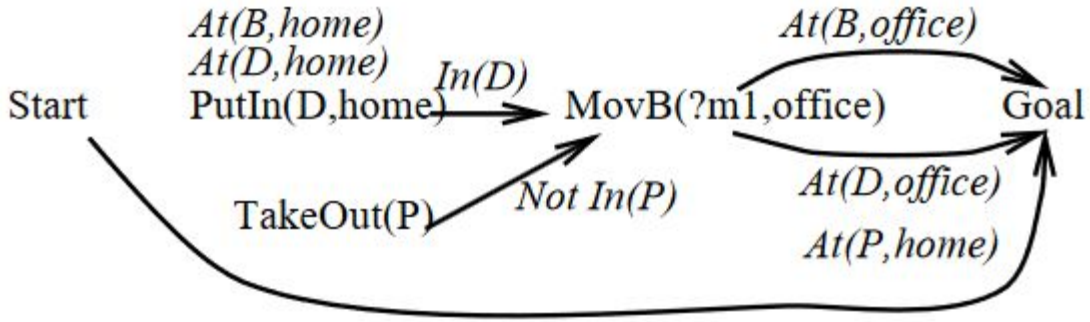
Briefcase Problem (Cont.)

- We add a new subgoal, ~In(P) to remove the paycheck from the briefcase
- This is solved by taking the paycheck out of the briefcase before moving it



Briefcase Problem (Cont.)

- We now select In(D) as a goal
- This is solved by putting the dictionary in the briefcase before moving it
- We now have two new goals, B and D must be at home



Briefcase Problem (Cont.)

- We can resolve these last two goals from the initial state of the problem
- ?m1 can be assigned *home*
- The plan is complete!