Partially Ordered Plan

- Plan
  - Steps
  - Ordering constraints
  - Variable binding constraints
  - Causal links
- POP Algorithm
  - Make initial plan
    - Loop until plan is a complete
      - Select a subgoal
      - Choose an operator
      - Resolve threats

Choose Operator

- Choose operator(c, S_{new})
  - Choose a step S from the plan or a new step S by instantiating an operator that has c as an effect
  - If there's no such step, Fail
  - Add causal link S \rightarrow S_{new}
  - Add ordering constraint S < S_{new}
  - Add variable binding constraints if necessary
  - Add S to steps if necessary

Nondeterministic choice

- Choose - pick one of the options arbitrarily
- Fail - go back to most recent non-deterministic choice and try a different one that has not been tried before

Resolve Threats

- A step S threatens a causal link S_i \rightarrow S_j iff \neg c \in \text{effects}(S) and it's possible that S_i < S < S_j

- For each threat
  - Choose
    - Promote S : S < S_i < S_j
    - Demote S : S_i < S_j < S
  - If resulting plan is inconsistent, then Fail

Threats with Variables

If c has variables in it, things are kind of tricky.

- S is a threat if there is any instantiation of the variables that makes \neg c \in \text{effects}(S)
- We could possibly resolve the threat by adding a negative variable binding constraint, saying that two variables or a variable and a constant cannot be bound to one another
- Another strategy is to ignore such threats until the very end, hoping that the variables will become bound and make things easier to deal with

Shopping Domain

- Actions
  - Buy(x, store)
    - Pre: At(store), Sells(store, x)
    - Eff: Have(x)
  - Go(x, y)
    - Pre: At(x)
    - Eff: At(y), \neg At(x)
- Goal
  - Have(Milk) \land Have(Banana) \land Have(Drill)
- Start
  - At(Home) \land Sells(SM, Milk) \land Sells(SM, Banana) \land Sells(HW, Drill)
Shop 'til You Drop!

start
At (Home)

At(HDW) Sells(HDW,D) At(SM) Sells(SM,M) At(SM) Sells(SM,B)
Buy (Drill) Buy (Milk) Buy (Bananas)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)

GO (HDW) GO (SM)

At(x2) GO (SM) At(x2) GO (HDW)

At(x1) ¬ At(x1)

start
At (Home) Buy (Bananas) Buy (Drill) Sells (HDW,D) At(HDW)
Buy (Milk) At (SM) Sells(SM,M) Sells(SM,B)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)

GO (HDW) GO (SM)

At(x2) GO (SM) At(x2) GO (HDW)

At(x1) ¬ At(x1)

start
At (Home) Buy (Bananas) Buy (Drill) Sells (HDW,D) At(HDW)
Buy (Milk) At (SM) Sells(SM,M) Sells(SM,B)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)

GO (HDW) GO (SM)

At(x2) GO (SM) At(x2) GO (HDW)

At(x1) ¬ At(x1)

start
At (Home) Buy (Bananas) Buy (Drill) Sells (HDW,D) At(HDW)
Buy (Milk) At (SM) Sells(SM,M) Sells(SM,B)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)

GO (HDW) GO (SM)

At(x2) GO (SM) At(x2) GO (HDW)

At(x1) ¬ At(x1)

start
At (Home) Buy (Bananas) Buy (Drill) Sells (HDW,D) At(HDW)
Buy (Milk) At (SM) Sells(SM,M) Sells(SM,B)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)

GO (HDW) GO (SM)

At(x2) GO (SM) At(x2) GO (HDW)

At(x1) ¬ At(x1)

start
At (Home) Buy (Bananas) Buy (Drill) Sells (HDW,D) At(HDW)
Buy (Milk) At (SM) Sells(SM,M) Sells(SM,B)

NB: Causal links imply ordering of steps

finish
Have(D) Have(B) Have(M)

At(SM) Sells(SM,B)
**Shop 'til You Drop!**

- **start**
  - At (Home)
  - Buy (Drill)
  - Sells (HDW, D) At (HDW)
  - At (x2)
  - GO (SM)
  - At (x2)
  - Buy (Milk)
  - At (SM) Sells (SM, M)
  - finish
  - Have (D)
  - Have (B)
  - Have (M)
  - At (SM) Sells (SM, B)
  - GO (HDW)
  - At (x1)
  - ¬ At (x1)

- **x1 = Home**
- **x2 = Home**
- **x2 = HDW**

---

**Subgoal Dependence**

- **objects**: A, B, C, Table
- **on (x, y)**
- **clear (x)**
- **Goal**: on (A, B) ∧ on (B, C)

- **Put A on B first**
- **Put B on C first**
Lecture 11  

Sussman Anomaly

Start:
- clear(T)
- on(C,A)
- on(A,T)
- on(B,T)
- clear(C)
- clear(B)

Move (A,x1,B):
- on(A,B)
- on(B,C)

Finish:
- on(A,B)
- on(B,C)

Move (A,x1,B):
- on(A,B)
- on(B,C)
- clear(A)
- clear(B)

Finish:
- on(A,B)
- on(B,C)

MoveT(y,A)
Lecture 11

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)

finish

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
on(y,A) clear(y)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)

finish

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C

on(y,A) clear(y)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)

finish

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C

on(y,A) clear(y)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)

finish

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

y=C

on(y,A) clear(y)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)

finish

Sussman Anomaly

start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

y=C

on(y,A) clear(y)

MoveT(y,A)
on(y,A) clear(A) ¬on(y,A)
on(A,x₁,B)

on(A) clear(A) ¬on(A)
on(A,B) ¬on(A,B) clear(x₁) ¬clear(B)
on(A,B) on(B,C)
Recitation Problems

- Russell & Norvig
  - 11.2
  - 11.7 a,b,c
  - 11.9