PLANNING II
Partial Order Planning

• The answer to the problem we ended the last lecture with is to use partial order planning.

• Basically this gives us a way of checking before adding an action to the plan that it doesn’t mess up the rest of the plan.

• The problem is that in this recursive process, we don’t know what the rest of the plan is.

• Need a new representation *partially ordered plans*. 
Partial Order Plan:

- Start
- Left Sock
  - Right Sock
    - Right SockOn
      - LeftShoeOn, RightShoeOn
        - Finish
      - Right SockOn
    - Right Sock
  - Left Sock
    - Left Sock
      - Left SockOn, RightShoeOn
        - Finish
      - Right Shoe
    - Left Shoe
      - Right Shoe
        - LeftShoeOn, RightShoeOn
          - Finish
      - Left Shoe
        - Left Sock
          - Start
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left Sock
                - Left SockOn
                  - LeftShoeOn, RightShoeOn
                    - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Start
                - Right Sock
                  - Right SockOn
                    - LeftShoeOn, RightShoeOn
                      - Finish
                  - Right SockOn
                    - Right Sock
                      - Finish
                - Left SockOn
                  - LeftShoeOn, RightShoeOn
                    - Finish
                  - Right SockOn
                    - Right Sock
                      - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Right Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
            - Left Sock
              - Right SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Left SockOn
                - LeftShoeOn, RightShoeOn
                  - Finish
              - Right SockOn
                - Right Sock
                  - Finish
Partially ordered plans

- Partially ordered collection of steps with
  - *Start* step has the initial state description as its effect
  - *Finish* step has the goal description as its precondition
  - *causal links* from outcome of one step to precondition of another
  - *temporal ordering* between pairs of steps
- *Open condition* = precondition of a step not yet causally linked
- A plan is *complete* iff every precondition is achieved
- A precondition is *achieved* iff it is the effect of an earlier step and no possibly intervening step undoes it
Plan construction

Start

At(Home) Sells(HWS, Drill) Sells(SM, Milk) Sells(SM, Ban.)

Have(Milk) At(Home) Have(Ban.) Have(Drill)

Finish
Plan construction (2)

Start

At(Home)  Sells(HWS,Drill)  Sells(SM,Milk)  Sells(SM,Ban.)

At(HWS)  Sells(HWS,Drill)

Buy(Drill)

At(x)

Go(SM)

At(SM)  Sells(SM,Milk)

Buy(Milk)

Have(Milk)  At(Home)  Have(Ban.)  Have(Drill)

Finish

cis716-fall2003-parsons-lect12
Plan construction (3)

Start
- At(Home)
- Go(HWS)
- Go(SM)

Buy(Drill)
- At(HWS) Sells(HWS,Drill)
- Buy(Milk)
- Sells(SM,Milk)
- At(SM) Sells(SM,Ban.)

Buy(Ban.)
- At(SM)
- At(HWS)
- Go(HWS)
- Go(SM)
- At(Home)

Have(Milk) Have(Ban.) Have(Home) Have(Drill)

Finish
Planning process

- Operators on partial plans:
  - *add a link* from an existing action to an open condition
  - *add a step* to fulfill an open condition
  - *order* one step wrt another to remove possible conflicts
- Gradually move from incomplete/vague plans to complete, correct plans
- Backtrack if an open condition is unachievable or if a conflict is unresolvable
**POP algorithm**

**function** POP(initial, goal, operators) **returns** plan

\[
\text{plan} \leftarrow \text{MAKE-MINIMAL-PLAN}(\text{initial}, \text{goal})
\]

**loop** do

  **if** SOLUTION?(plan) **then return** plan

  \[ S_{\text{need}}, c \leftarrow \text{SELECT-SUBGOAL}(\text{plan}) \]

  \[ \text{CHOOSE-OPERATOR}(\text{plan}, \text{operators}, S_{\text{need}}, c) \]

  \[ \text{RESOLVE-THREATS}(\text{plan}) \]

**end**

**function** SELECT-SUBGOAL(plan) **returns** S_{\text{need}}, c

\[
\text{pick a plan step } S_{\text{need}} \text{ from STEPS(plan)}
\]

\[
\text{with a precondition } c \text{ that has not been achieved}
\]

**return** S_{\text{need}}, c
procedure CHOOSE-OPERATOR\( (plan, \text{operators}, S_{\text{need}}, c) \)

choose a step \( S_{\text{add}} \) from \( \text{operators} \) or \( \text{STEPS}(plan) \) that has \( c \) as an effect

if there is no such step then fail
add the causal link \( S_{\text{add}} \xrightarrow{c} S_{\text{need}} \) to \( \text{LINKS}(plan) \)
add the ordering constraint \( S_{\text{add}} \prec S_{\text{need}} \) to \( \text{ORDERINGS}(plan) \)

if \( S_{add} \) is a newly added step from \( \text{operators} \) then
add \( S_{add} \) to \( \text{STEPS}(plan) \)
add \( \text{Start} \prec S_{add} \prec \text{Finish} \) to \( \text{ORDERINGS}(plan) \)

procedure RESOLVE-THREATS\( (plan) \)

for each \( S_{\text{threat}} \) that threatens a link \( S_i \xrightarrow{c} S_j \) in \( \text{LINKS}(plan) \)
do
choose either

Demotion: Add \( S_{\text{threat}} \prec S_i \) to \( \text{ORDERINGS}(plan) \)
Promotion: Add \( S_j \prec S_{\text{threat}} \) to \( \text{ORDERINGS}(plan) \)

if not CONSISTENT\( (plan) \) then fail
end
A clobberer is a potentially intervening step that destroys the condition achieved by a causal link. E.g., \(Go(\text{Home})\) clobbers \(At(\text{Supermarket})\):

- **Demotion**: put before \(Go(\text{Supermarket})\)

- **Promotion**: put after \(Buy(Milk)\)
Properties of POP

- Nondeterministic algorithm: backtracks at choice points on failure:
  - choice of $S_{add}$ to achieve $S_{need}$
  - choice of demotion or promotion for clobberer
  - selection of $S_{need}$ is irrevocable
- POP is sound, complete, and systematic (no repetition)
- Extensions for disjunction, universals, negation, conditionals
- Can be made efficient with good heuristics derived from problem description
- Particularly good for problems with many loosely related subgoals
Example

"Sussman anomaly" problem

Start State

\[ \text{Clear}(x) \land \text{On}(x,z) \land \text{Clear}(y) \]

\[ \text{PutOn}(x,y) \]

\[ \neg\text{On}(x,z) \land \neg\text{Clear}(y) \]

\[ \text{Clear}(z) \land \text{On}(x,y) \]

Goal State

\[ \text{Clear}(x) \land \text{On}(x,z) \]

\[ \text{PutOnTable}(x) \]

\[ \neg\text{On}(x,z) \land \text{Clear}(z) \land \text{On}(x,\text{Table}) \]

+ several inequality constraints
Example (2)

On(C, A) On(A, Table) Cl(B) On(B, Table) Cl(C)

On(A, B) On(B, C)

FINISH
Example (3)

- On(C, A) On(A, Table) Cl(B) On(B, Table) Cl(C)
- PutOn(B, C)
- On(A, B) On(B, C)

START

FINISH
Example (4)

On(A,B)  On(B,C)

PutOn(A,B)  clutters Cl(B) => order after  PutOn(B,C)

On(A,z)  Cl(B)

Cl(A)  On(A,Table)  Cl(B)  On(B,Table)  Cl(C)

PutOn(A,B)

On(B,z)  Cl(C)

Cl(B)

PutOn(B,C)

On(B,C)

FINISH

PutOn(A,B)

On(A,Table)  Cl(B)  On(B,Table)  Cl(C)

C

B

A

On(C,A)  On(A,Table)  Cl(B)  On(B,Table)  Cl(C)

Start

B

A

On(C,A)  On(A,Table)  Cl(B)  On(B,Table)  Cl(C)

PutOn(A,B)

On(A,B)  On(B,C)

FINISH

B

A

C

A

B

G

 cis716-fall2003-parsons-lect12
Example (5)

On(A,B)     On(B,C)

Start

On(C,A) On(A,Table) Cl(B) On(B,Table) Cl(C)

PutOn(A,B)

Cl(A) On(A,z) Cl(B)

PutOn(A,B)

Cl(B) On(B,z) Cl(C)

PutOn(B,C)

PutOnTable(C)

On(C,z) Cl(C)

PutOnTable(C)

FINISH

PutOn(B,C) clobbers Cl(B) => order after PutOn(B,C)

PutOn(B,C) clobbers Cl(C) => order after PutOnTable(C)
Summary

- This lecture has looked at a more advanced approach to planning.
  - Partial order planning
- This requires a new way of looking at the world, but the payoff is a more robust approach.
- We also looked at the POP algorithm, …
- … and saw how it could solve the Sussman anomaly.