Title: Planning II

AIMA: Chapter 11 (Sections 11.5-11.6)

Introduction to Artificial Intelligence

CSCE 476-876, Spring 2000

www.cse.unl.edu/~cse476

Berthe Y. Choueiry

choueiry@cse.unl.edu, (402)472-5444

Partial-Order Planning: Example (I)

PO-planner that:

- searches through the plan space
- starts with an initial plan (initial state + goal state)
- at each iteration, adds one more step
- remains focused: only considers adding steps that serve to achieve a precondition that has not yet been achieved
  $\rightarrow$ causal links allow keeping track of this

Problem: milk-banana-drill simple problem

Assumptions:

- *Go* action used to travel between any two locations
- *Buy* action ignores money
- HWS = hardware store, SM = supermarket

PO Planning: Example (II)

- **Initial state**:

$Op\ (Action : Start,$
$\quad Effect : At(Home) \wedge Sells(HWS, Drill)$
$\quad\quad \wedge Sells(SM, Milk)$
$\quad\quad \wedge Sells(SM, Banana))$

- **Goal state**:

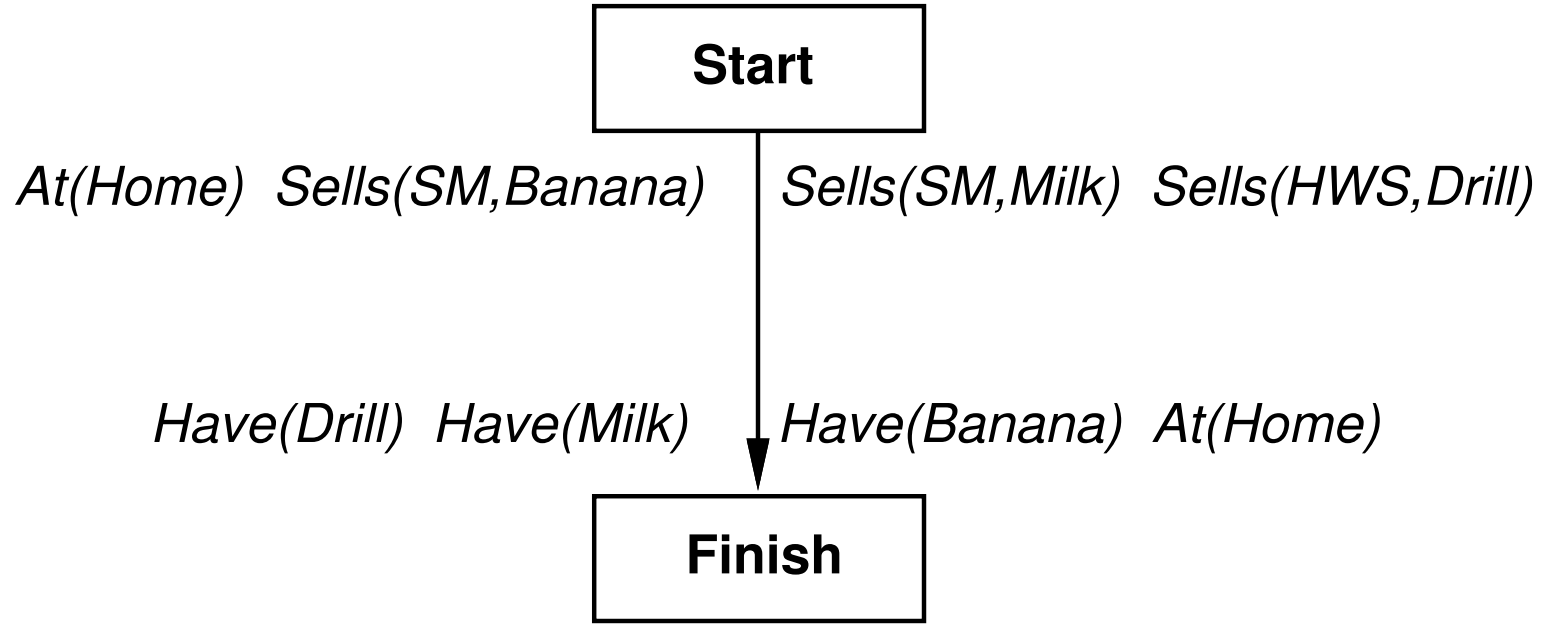
$Op\ (Action : Finish,$
$\quad Precond : Have(Drill) \wedge Have(Bananas)$
$\quad\quad \wedge Have(Milk) \wedge At(Home))$

- **Actions**:

$Op\ (Action : Go(there),$
$\quad Precond : At(here),$
$\quad Effect : At(there) \wedge \neg At(here))$

$Op\ (Action : Buy(x),$
$\quad Precond : At(store) \wedge Sells(store, x),$
$\quad Effect : Have(x)$

Initial plan:



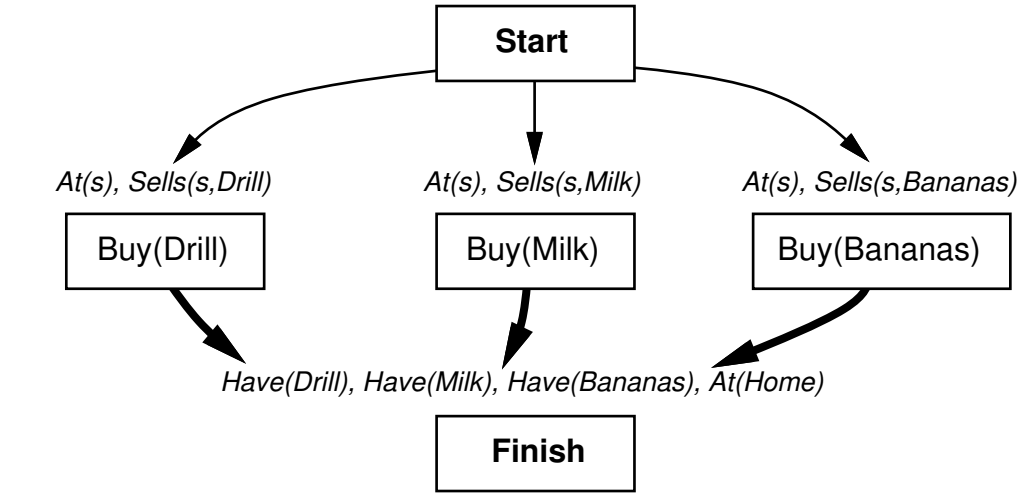
→ choose actions that achieve *Precond* of goal state

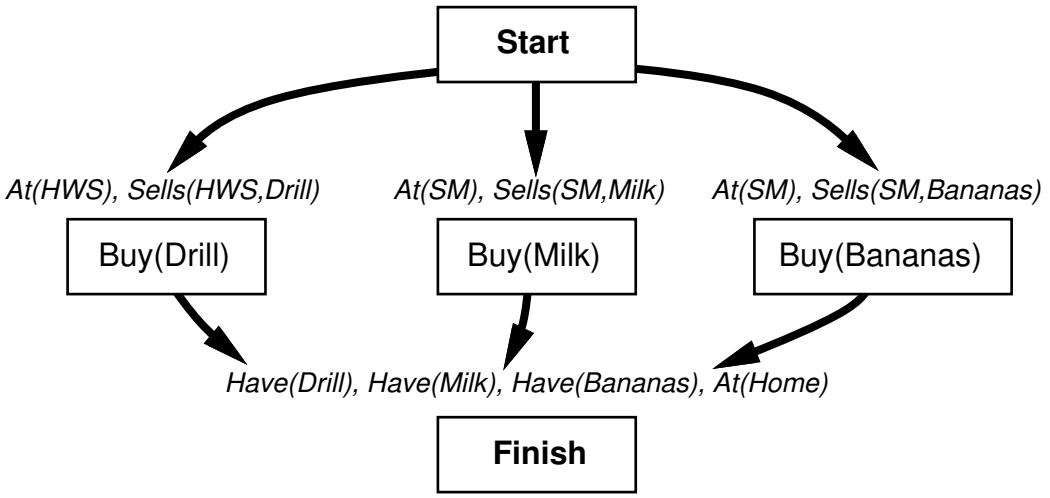
→ In general, there are correct and incorrect choices of actions.

→ In this case only one action: *Buy*

$\longrightarrow$ Bold arrow denotes a causal link

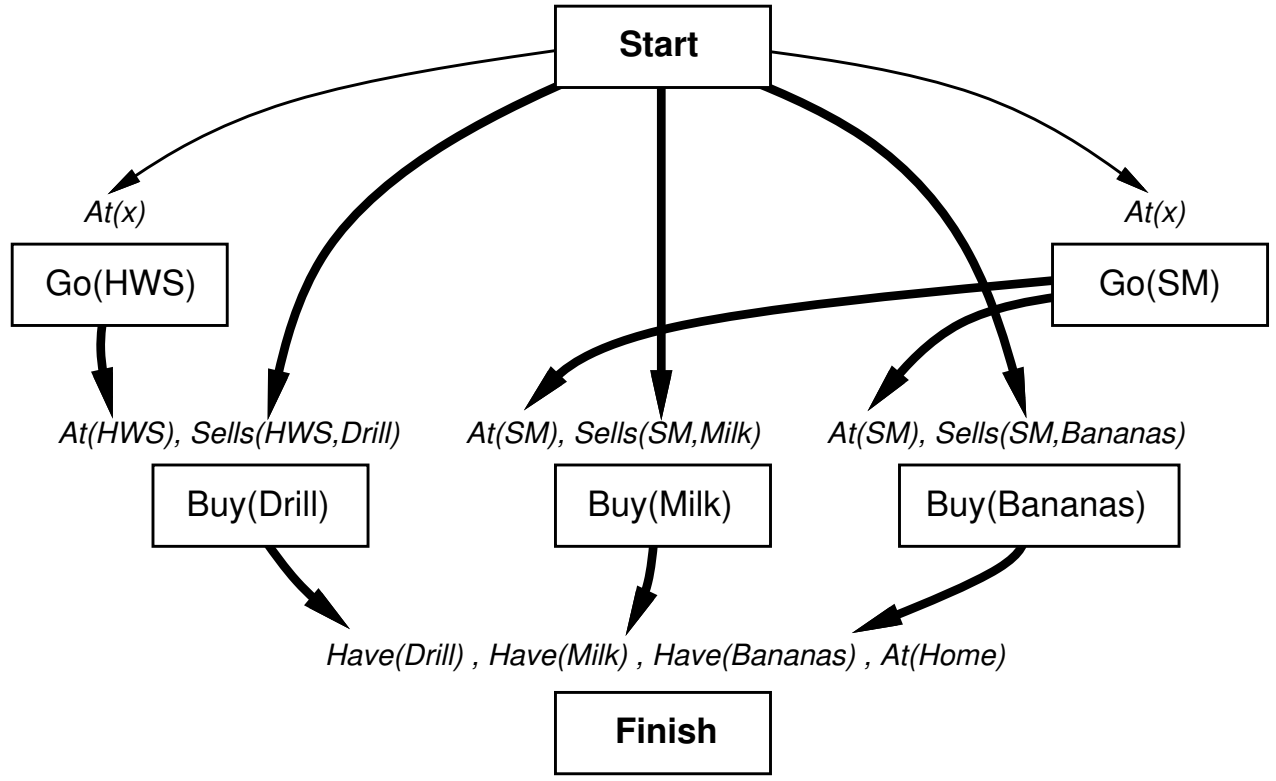
causal linked are protected: no step that deletes $Have(Drill)$ may be inserted between $Buy(Drill)$ and $Finish$





$\rightarrow$ No action to achieve $Sell$, achieved by linking them to $Start$

$\longrightarrow$ Action to achieve $At(x)$ is $Go(x)$



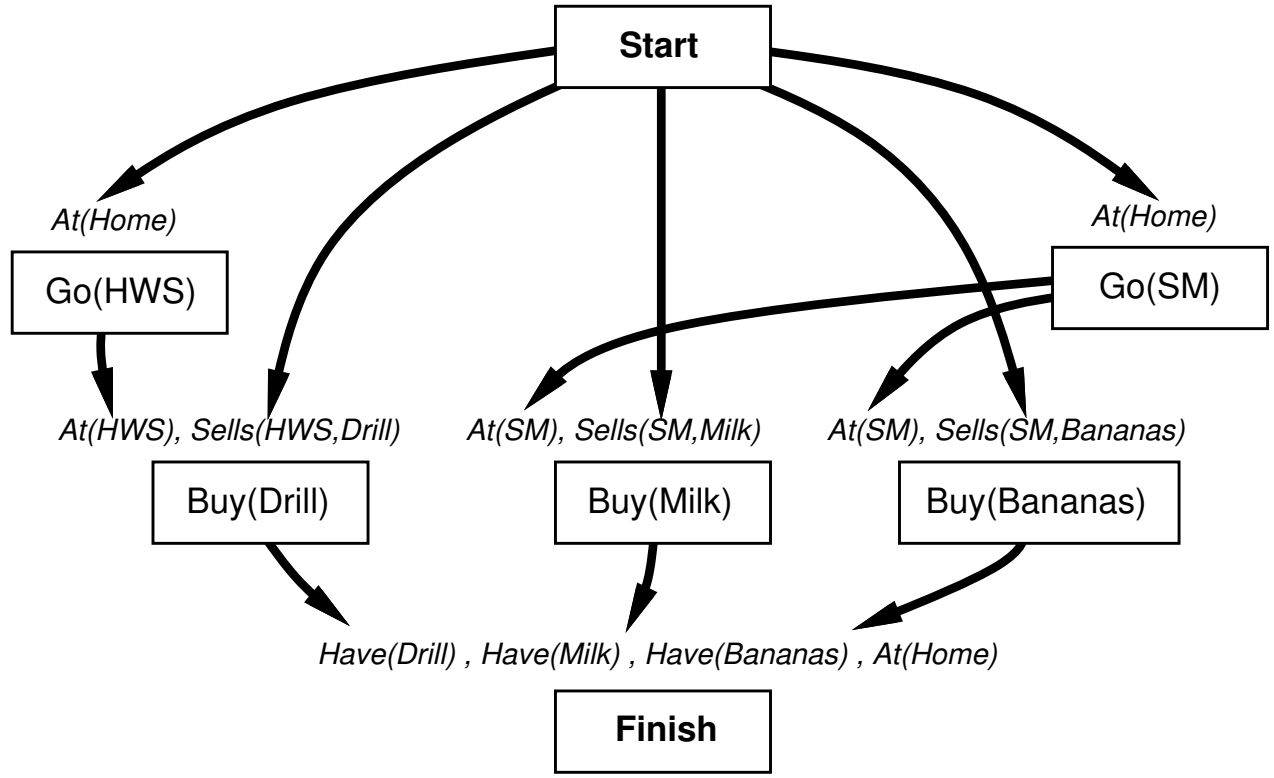
$\longrightarrow$ Try to achieve $At(x)$ with by linking to $Start$

$\longrightarrow$ Problem:

$Go(HWS)$ adds $At(HWS)$ and deletes $At(home)$

$Go(SM)$ is no longer feasible

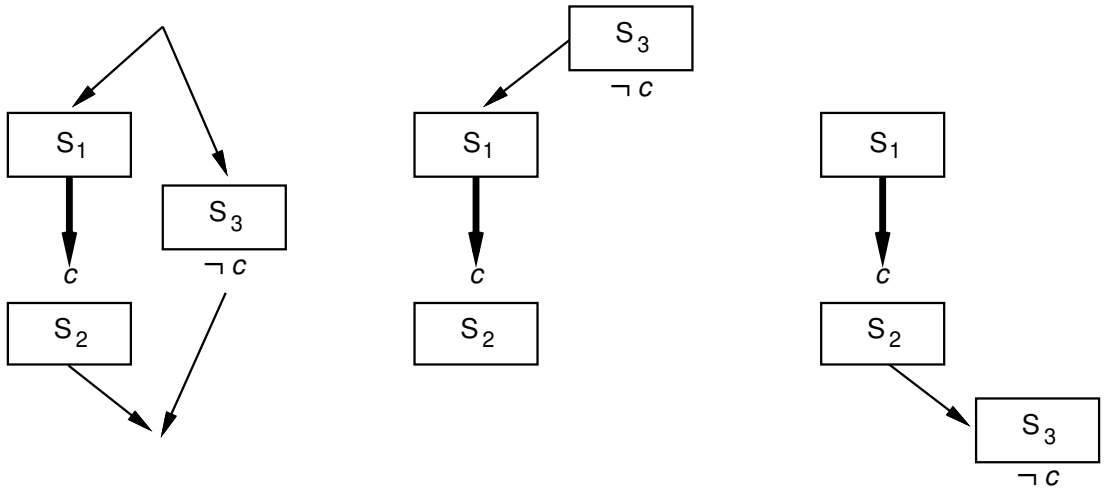
$\longrightarrow$ Solution: POP protects causal links



Protection of causal links:

Threats: steps that might delete (clobber) protected condition

Solved by adding ordering constraints: { demotion (S_3 placed before) ; promotion (S_3 placed before)

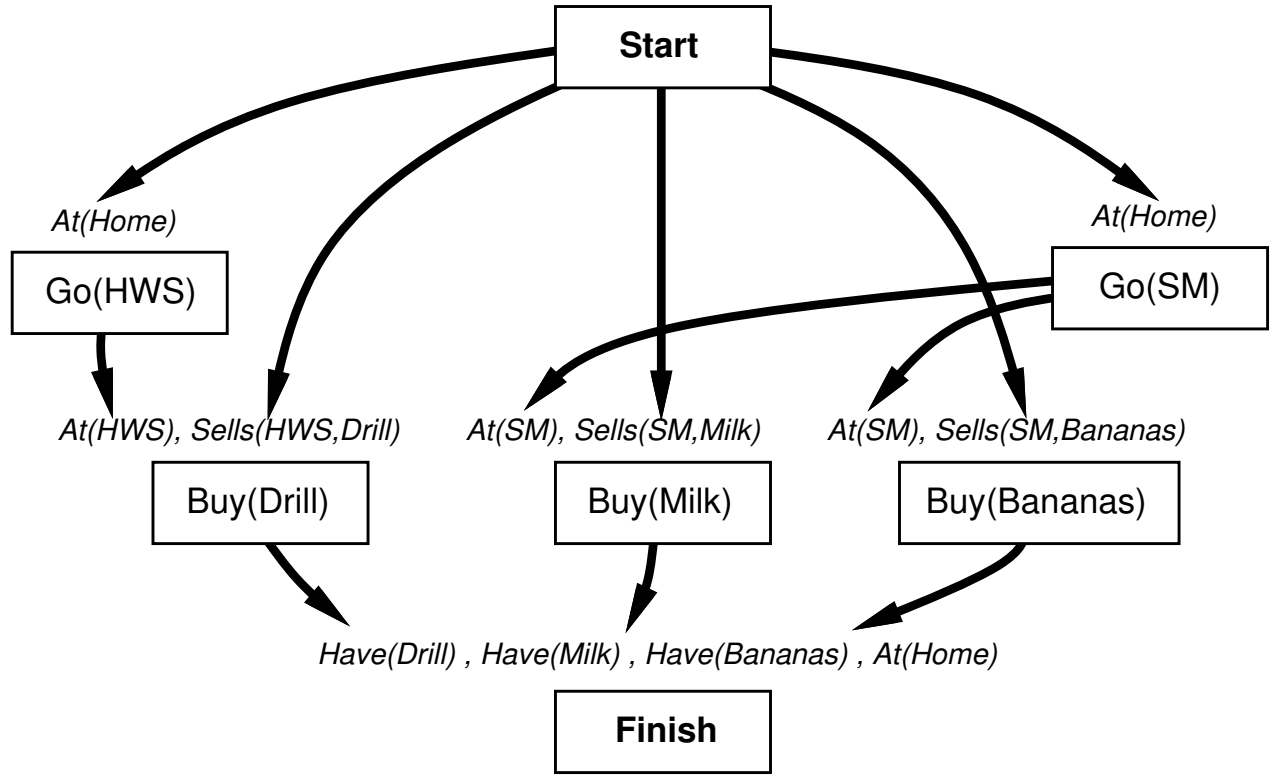


$\longrightarrow$ Promotion? Demotion?... none works?

backup to a different choice point in the plan

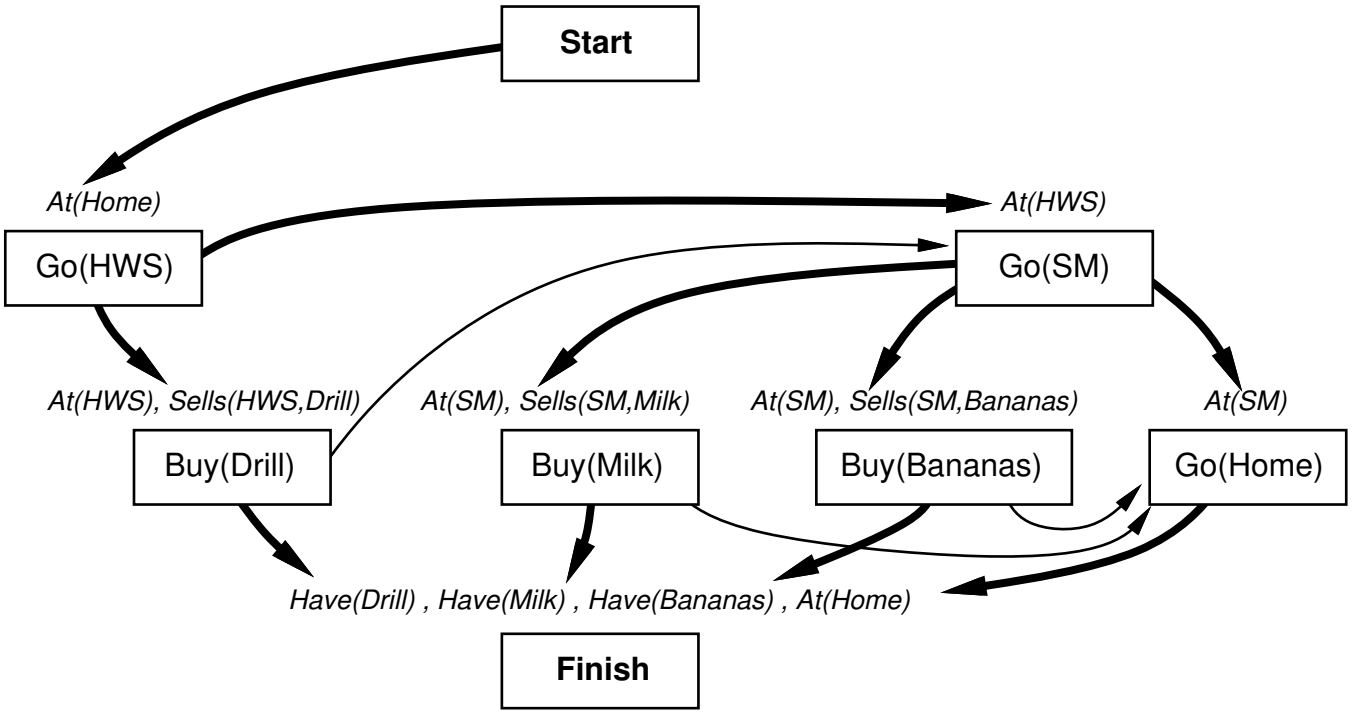
$\longrightarrow$ Try to achieve $At(x)$ of $Go(SM)$

by adding causal link from $Go(HWS)$ to $Go(SM)$



$\longrightarrow$ $Go(SM)$ threatens $At(HWS)$, precond for $Buy(Drill)$

Promote $Go(SM)$



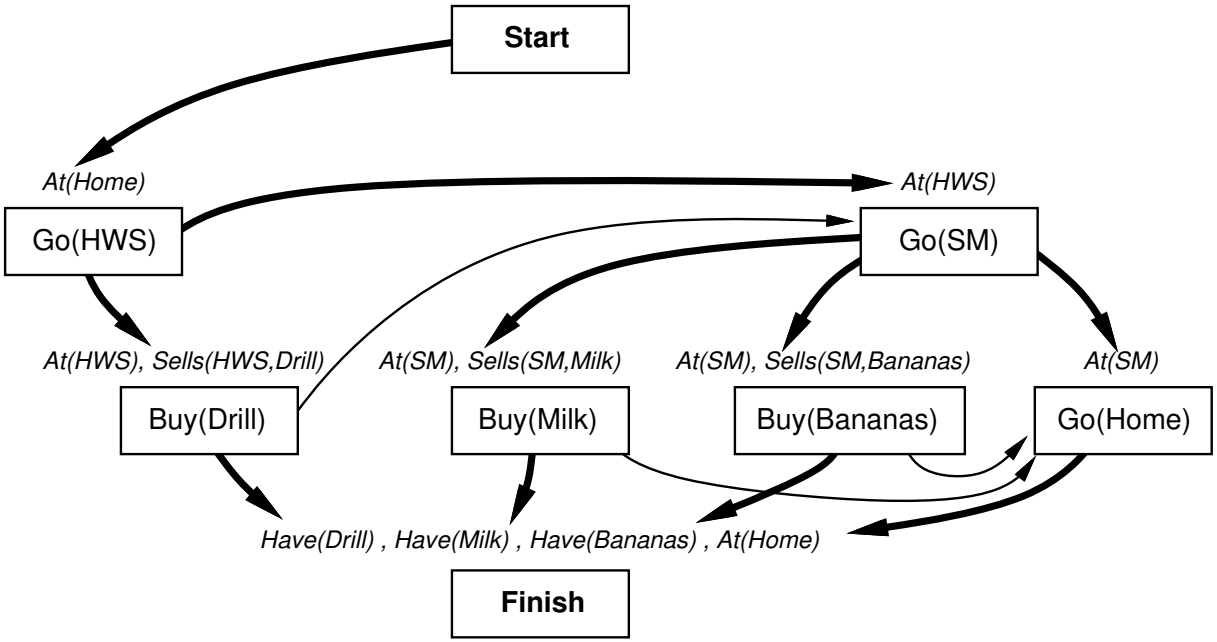
$\longrightarrow$ All precond achieved except $At(Home)$ of $Finish$

Add step $Go(home)$

– link to $Start$? No way to resolve threats $Go(SM)$, $Go(HWS)$

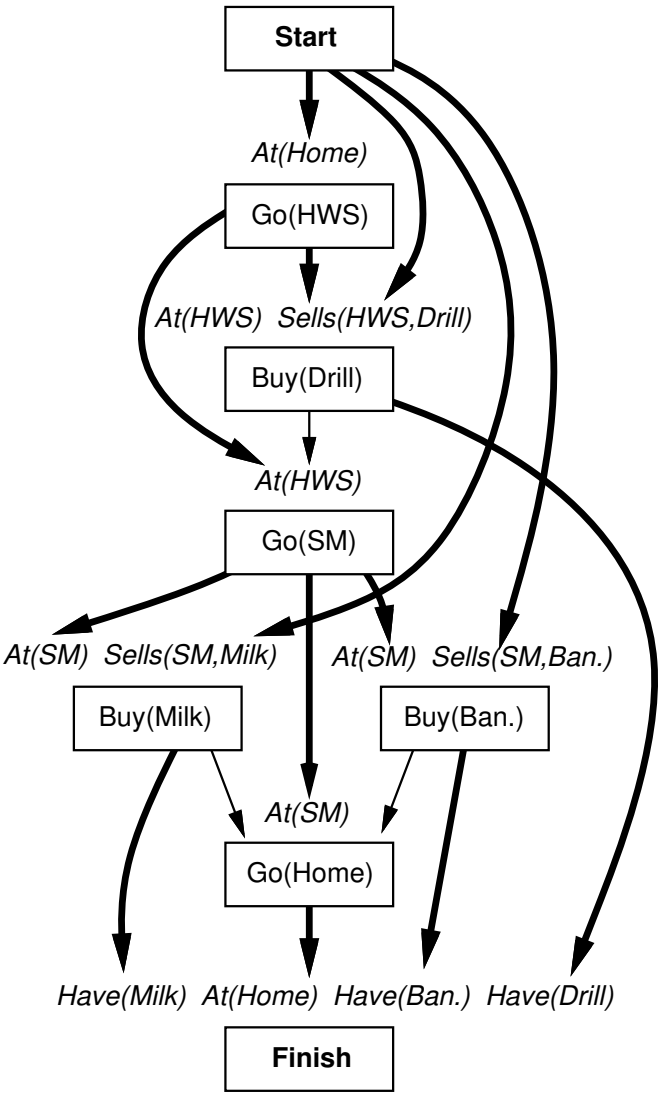
– link to $At(x)$ to $Go(HWS)$? No way to resolve threat $Go(SM)$

– link to $At(x)$ to $Go(SM)$? threat to $Buy(Milk)$ and $Buy(Bananas)$ solved by promoting $Go(home)$



Finally...

nearly a totally ordered plan (except for buy bananas and buy milk)



POPlanner

- solves in few steps what search solves in many many steps..
- least commitment: search is required only when subplans interact
- causal links recognize when to abandon a doomed plan without wasting time expanding irrelevant parts of the plan..

POPlanner algorithm

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

  plan ← MAKE-MINIMAL-PLAN(*initial, goal*)
  loop do
    if SOLUTION?(*plan*) **then return** *plan*
    S_{need}, c ← SELECT-SUBGOAL(*plan*)
    CHOOSE-OPERATOR(*plan, operators*, S_{need}, c)
    RESOLVE-THREATS(*plan*)
  end

---

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

  pick a plan step S_{need} from STEPS(*plan*)
    with a precondition c that has not been achieved
  return S_{need}, c

---

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

  choose a step S_{add} from *operators* or STEPS(*plan*) that has c as an effect
  if there is no such step **then fail**
  add the causal link $S_{add} \xrightarrow{c} S_{need}$ to LINKS(*plan*)
  add the ordering constraint $S_{add} \prec S_{need}$ to ORDERINGS(*plan*)
  if S_{add} is a newly added step from *operators* **then**
    add S_{add} to STEPS(*plan*)
    add $Start \prec S_{add} \prec Finish$ to ORDERINGS(*plan*)

---

procedure RESOLVE-THREATS(*plan*)

  for each S_{threat} that threatens a link $S_i \xrightarrow{c} S_j$ in LINKS(*plan*) **do**
    choose either
      Promotion: Add $S_{threat} \prec S_i$ to ORDERINGS(*plan*)
      Demotion: Add $S_j \prec S_{threat}$ to ORDERINGS(*plan*)
    if not CONSISTENT(*plan*) **then fail**
  end

→ Backtrack: when fail to resolve a threat or find operator

→ Select-subgoal is <u>not</u> subject to backtracking

POPlanner algorithm

- is a regression planner
- is sound, complete, and systematic (no repetition)
- ...
- go back to shopping example and apply mechanically

List of terms for glossary:

Due date: Wed, April 5, 2000

Words in margins in AIMA + ...