

# Part II: Methods of AI



4.1 State Space Planning

4.2 Partial Order Planning





# Partial Order Planning (POP)

See:  
Russell, Norvig, AI; chapter 11



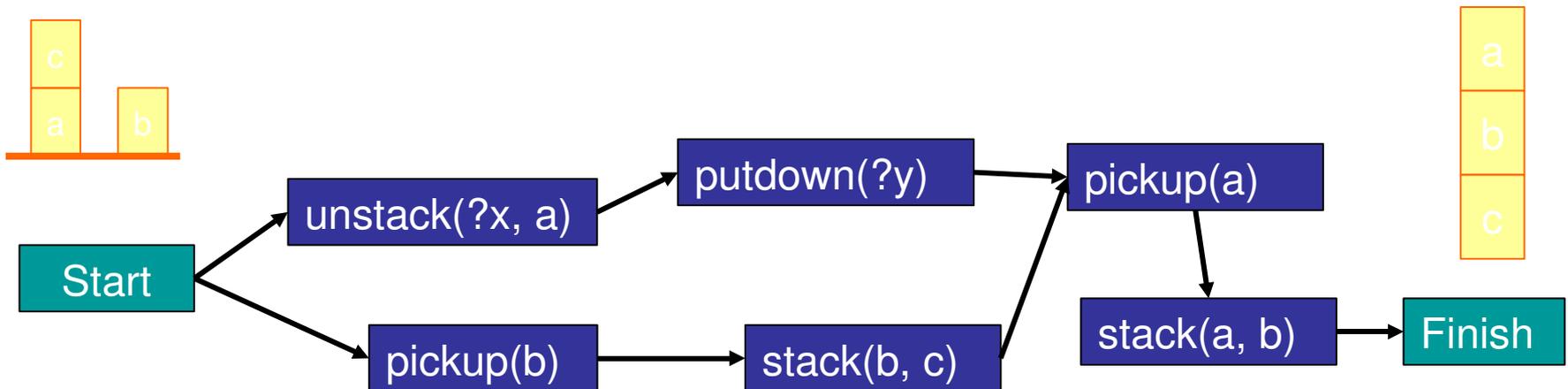
# Partial Plans



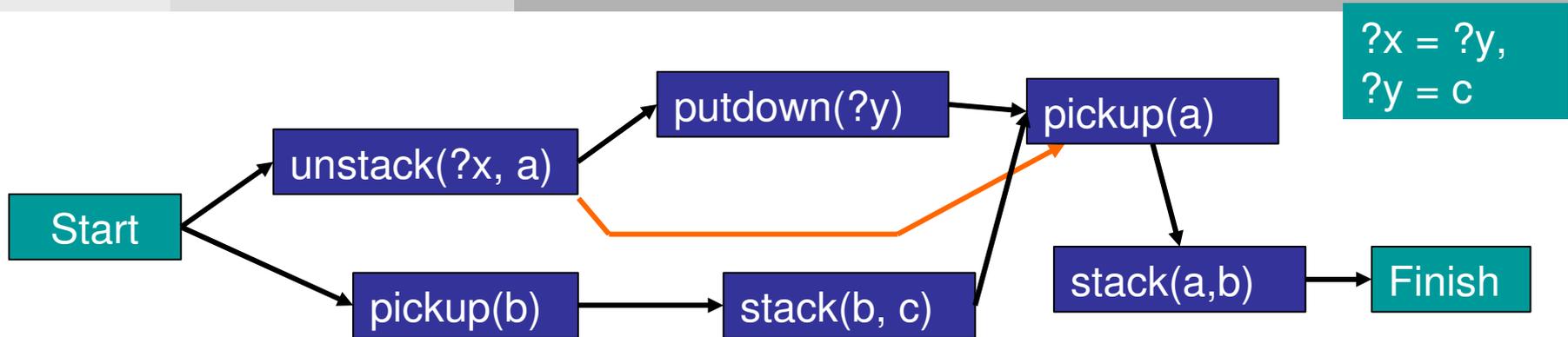
The partial order gives only some constraints on the order in which the operations have to be performed

**Start** a dummy operator

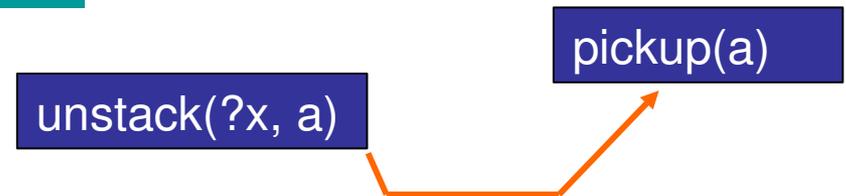
**Finish** another dummy operator



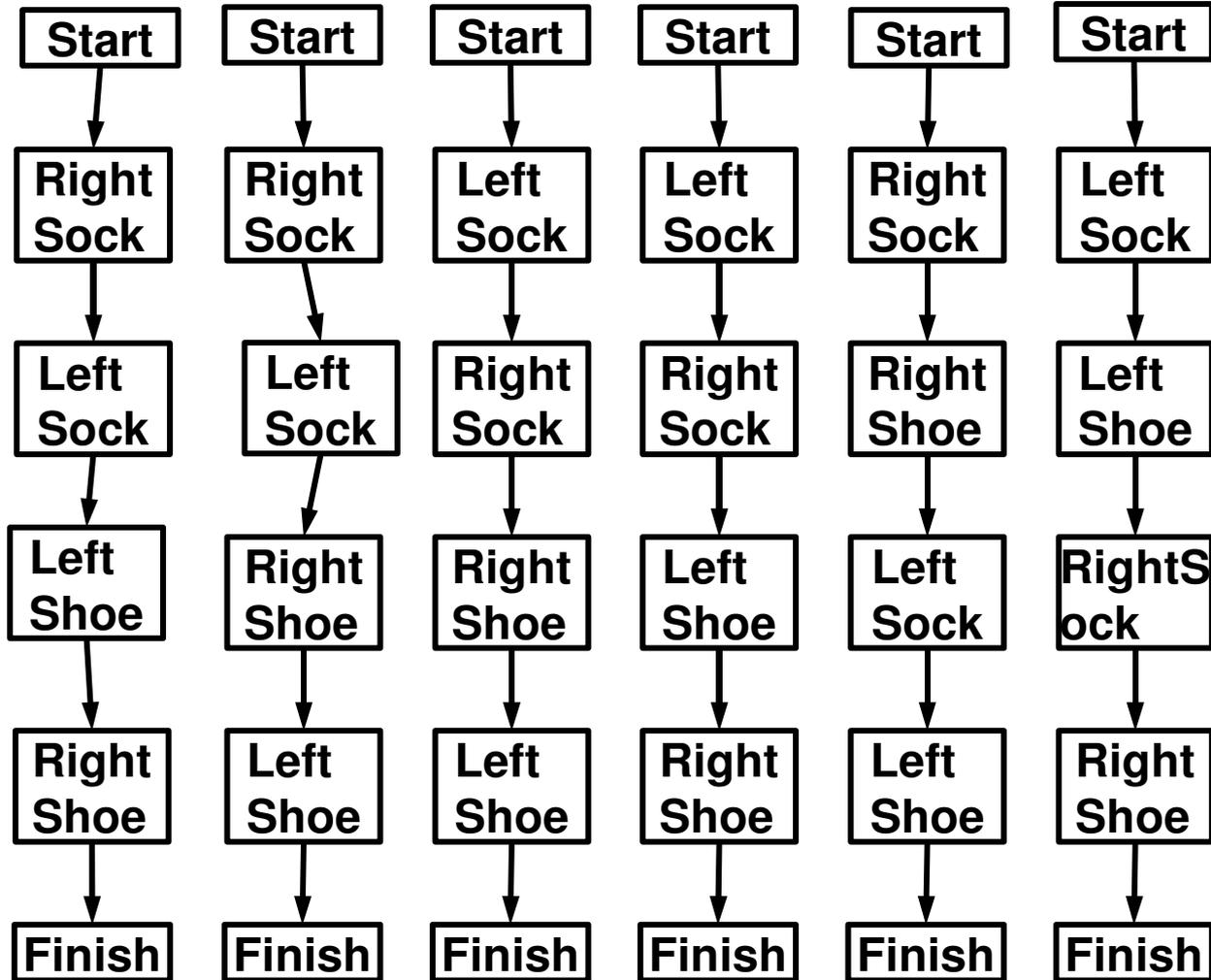
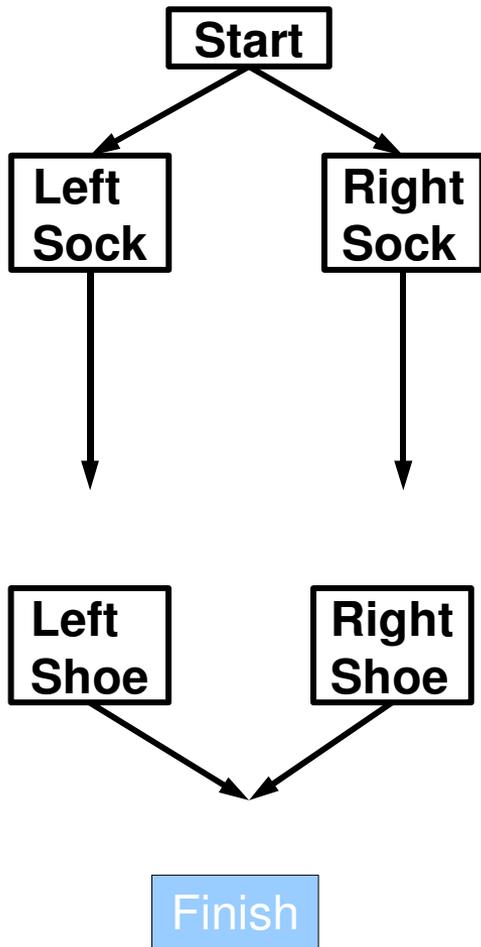
# Partial Plans



?x = ?y,  
?y = c



# Transform Partial Order Plan into Total-order Plan:



# The Poplan Algorithm

$A$  - actions  
 $O$  - order constraints  
 $L$  - links  
 $agenda$  - set of goals  
 $O$  - preconds of  $A$

POP ( $\langle A, O, L \rangle$ ,  $agenda$ )

**1. Termination:** If  $agenda$  is empty **return**  $\langle A, O, L \rangle$

**2. Goal selection:** Let  $\langle Q, A_{need} \rangle$  be a pair on the agenda

**3. Action selection:** Let  $A_{add}$  = choose an action that adds  $Q$   
**if** no such action exists, **then return failure**

**Let**  $L' = L \cup \{A_{add} \rightarrow A_{need}\}$ , and **let**  $O' = O \cup \{A_{add} < A_{need}\}$ .

**If**  $A_{add}$  is newly instantiated, **then**  $A' = A \cup \{A_{add}\}$  and

$O' = O \cup \{A_0 < A_{add} < A_{\infty}\}$  (otherwise,  $A' = A$ )

**4. Updating of goal set:** **Let**  $agenda' = agenda - \{\langle Q, A_{need} \rangle\}$ .

**If**  $A_{add}$  is newly instantiated, **then for each** conjunction,  $Q_i$ , of its precondition, add  $\langle Q_i, A_{need} \rangle$  to  $agenda'$

**5. Causal link protection:** For every action  $A_t$  that might threaten a causal link  $A_p \rightarrow A_c$ , add a consistent ordering constraint, either

(b) Demotion: Add  $A_t < A_p$  to  $O'$

(c) Promotion: Add  $A_c < A_t$  to  $O'$

(d) Inequality constraints

**If** neither constraint is consistent, **then return failure**

**6. Recursive invocation:** POP( $\langle A', O', L' \rangle$ ,  $agenda'$ ,  $\Lambda$ )



# THREATS

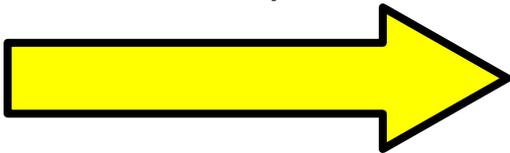


**DEFINITION:** Let  $\langle s \rightarrow t \rangle$  be a causal link with condition  $c$  in a partial plan  $P$ . Suppose  $P$  also contains a step  $r$  with an effect  $q$  such that

➤ **consistent <sub>$p$</sub> ( $s \rightarrow r$ )** and **consistent <sub>$p$</sub> ( $r \rightarrow t$ )**

i.e.,  $P$ 's ordering constraints allow putting  $r$  between  $s$  and  $t$

➤ Suppose  $q$  is unifiable with  $\sim c$ , and  $P$ 's variable-binding constraints do not prevent us from unifying it either



Then  $r$  is a *threat* to the causal link  $\langle s \rightarrow t \rangle$

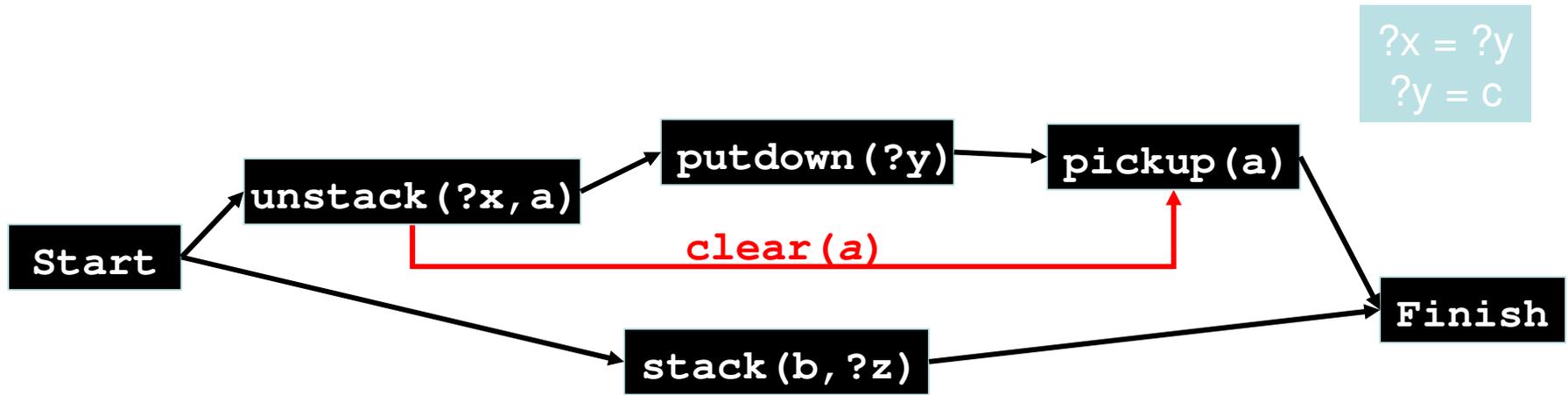


# Threats: Example



$r = \text{stack}(b, ?z)$

below is a threat to the causal link **clear(a)**

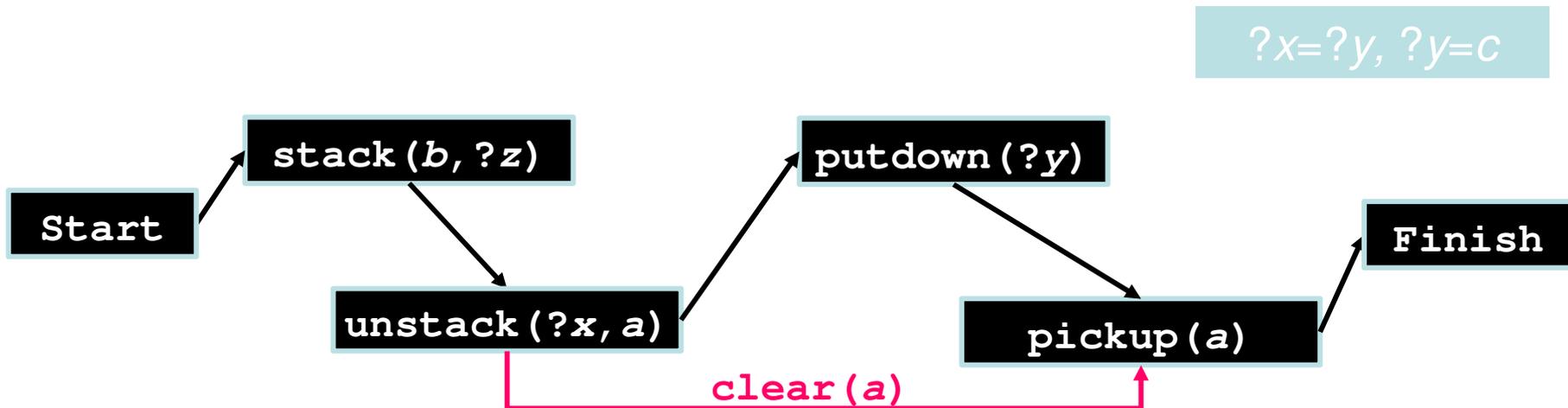


# Threat Detection and Removal (1)



- **Detection:** for each causal link  $\langle s \rightarrow t \rangle$  and for each step  $r$ , check positive and negative threats
- **Threat removal:**
  - (i) demotion, (ii) promotion and (iii) separation

(i) **Demotion:** order  $r$  before  $s$

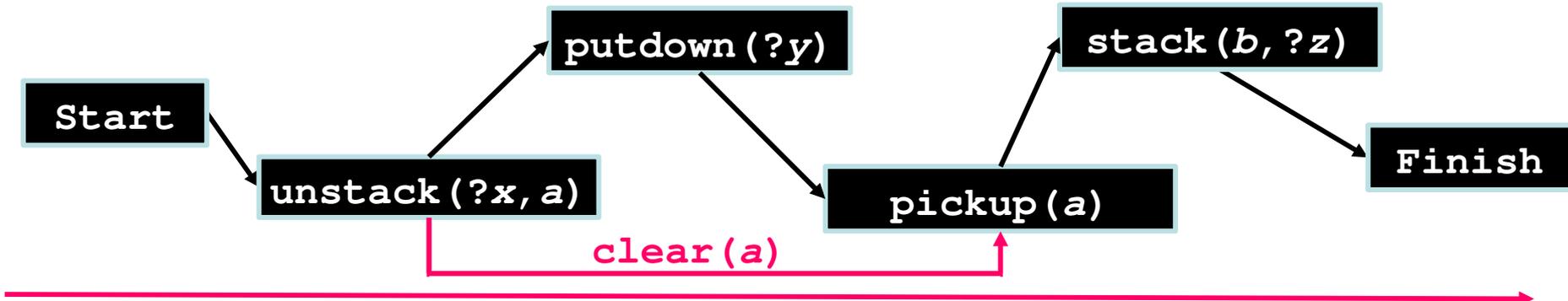


# Threat Detection and Removal

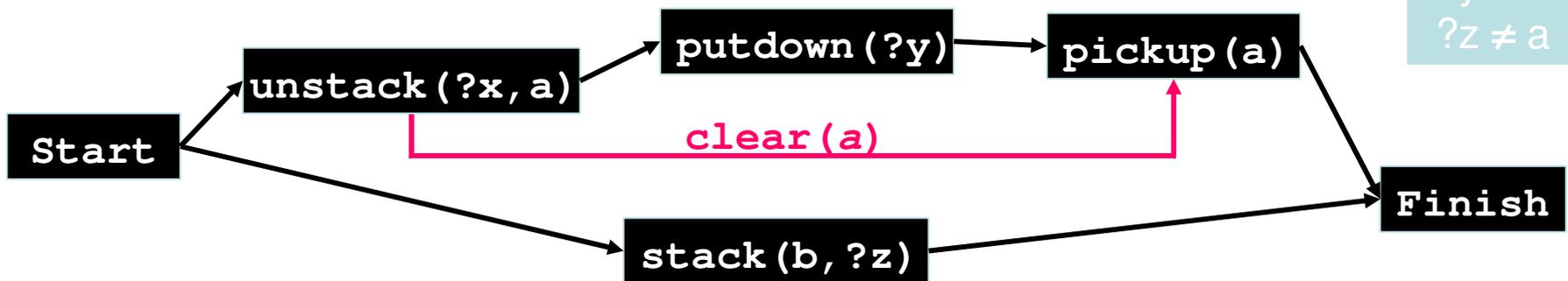


➤ (ii) **Promotion:** order  $r$  after  $t$

?x=?y, ?y=c



?x = ?y  
?y = c  
?z ≠ a

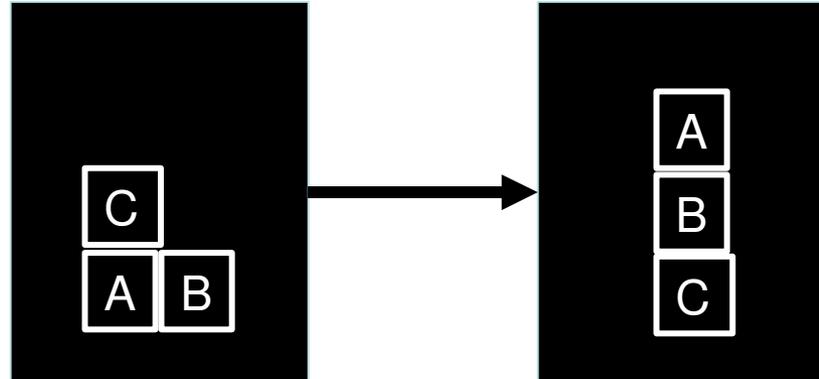


# POPLAN Search Space (1)



Start → Finish

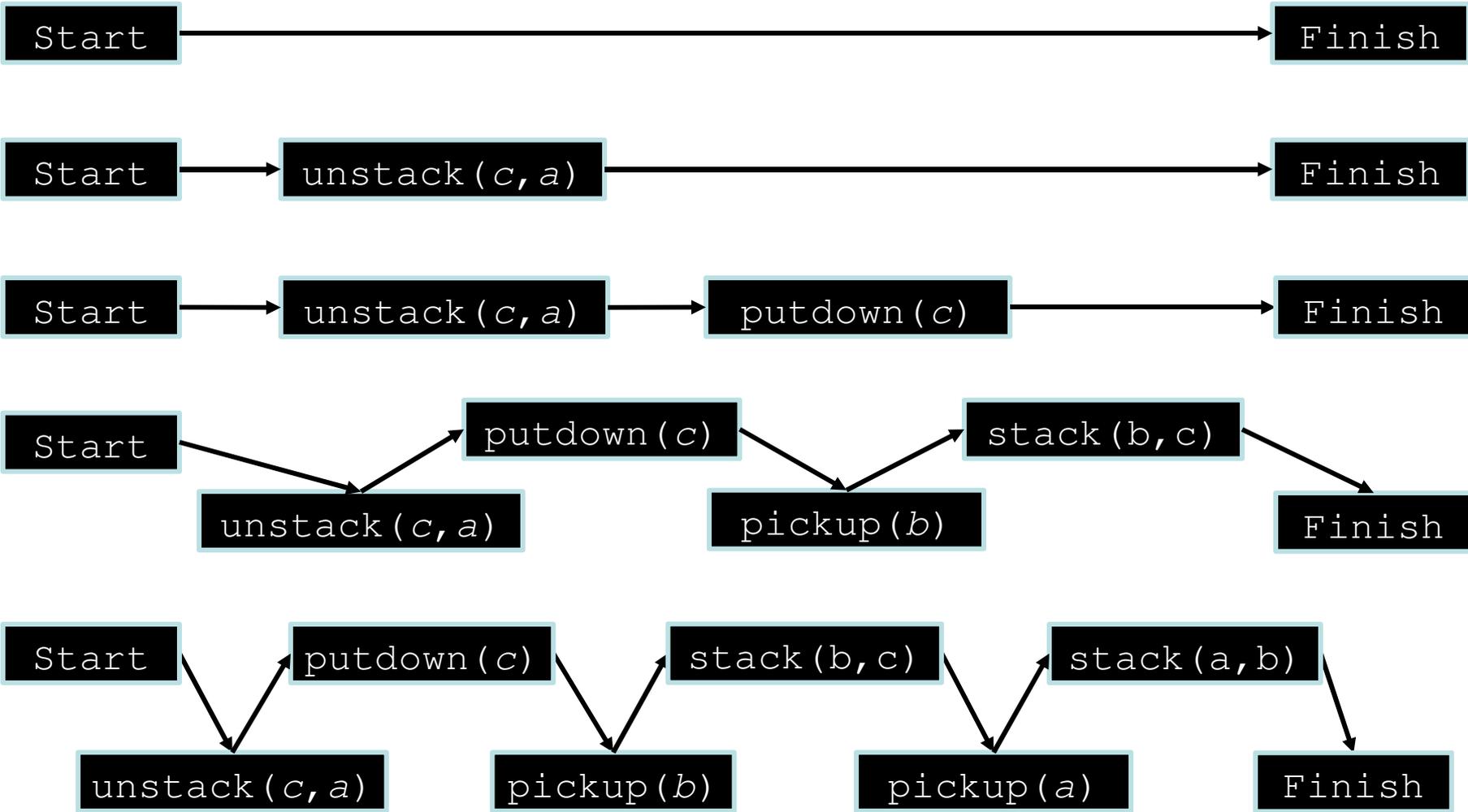
```
on(c, a)
ontable(a)
clear(c)
ontable(b)
clear(b)
handempty
```



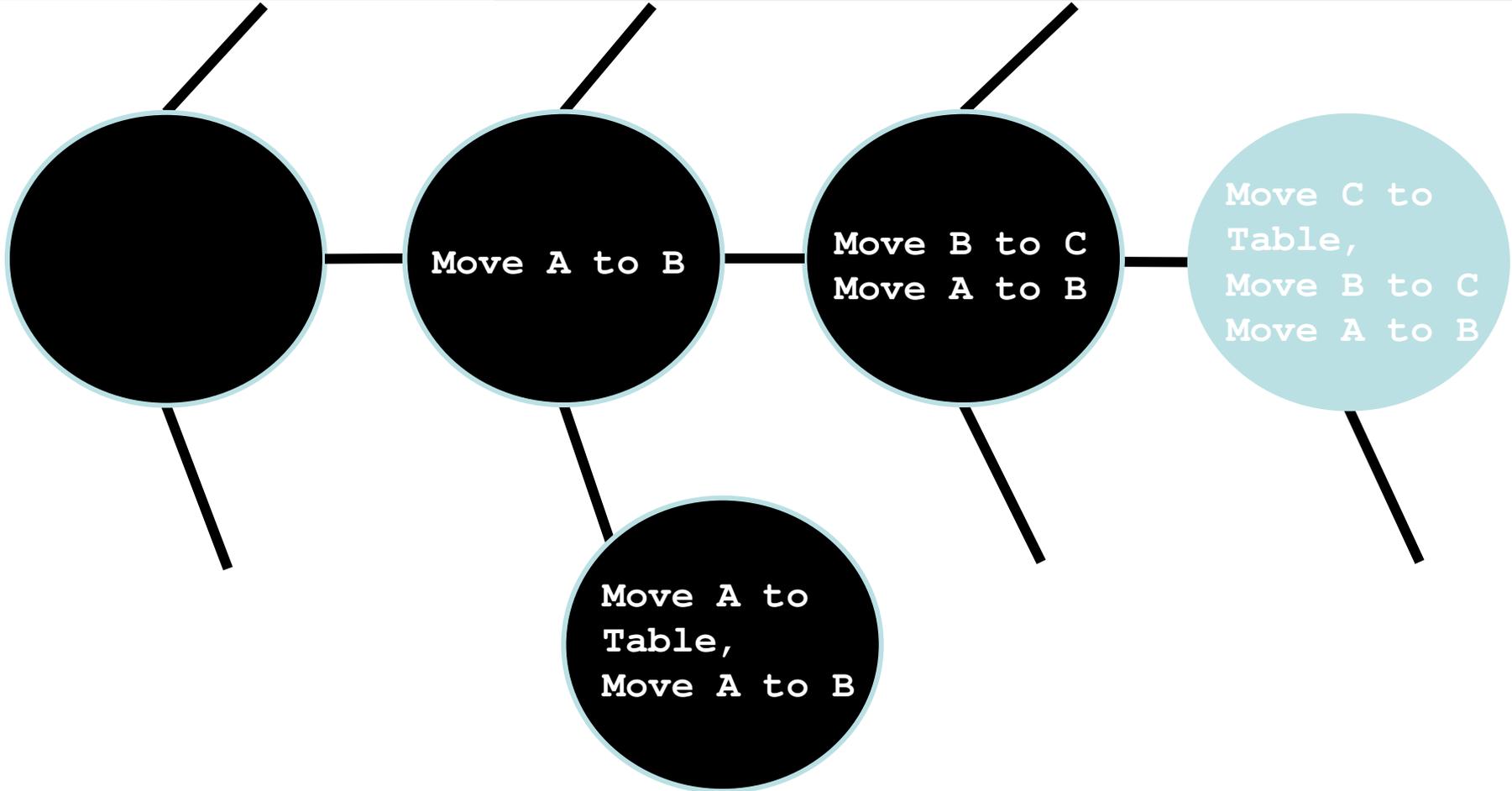
```
on(b, c)
on(a, b)
clear(a)
```



# Poplan Search Space (2)



# Plan Space Search



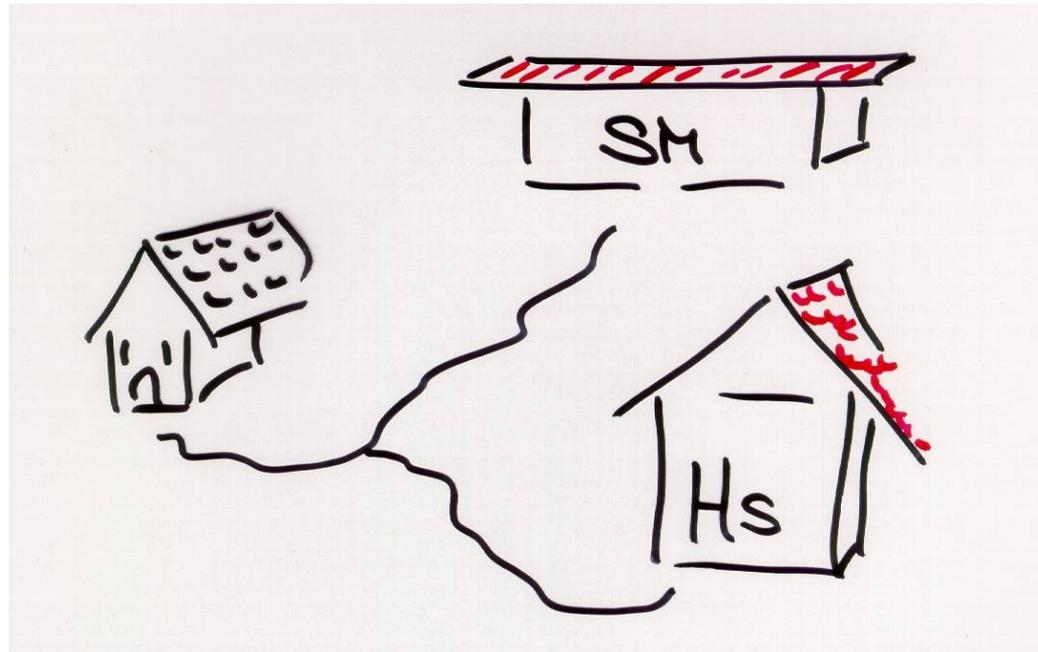


# Another Example



- At(Home)
- Sells(SM, Banana)
- Sells(HS, Drill)
- Have(Drill)
- Have(Milk)
- Have(Banana)

SM: Super Market  
HS: Hardware Store



# Partial Order Planning



Start At(Home)

Buy(Banana)

Buy(Milk)

Buy(Drill)

**Causal links**

Have(Banana)

Have(Milk)

Have(Drill)

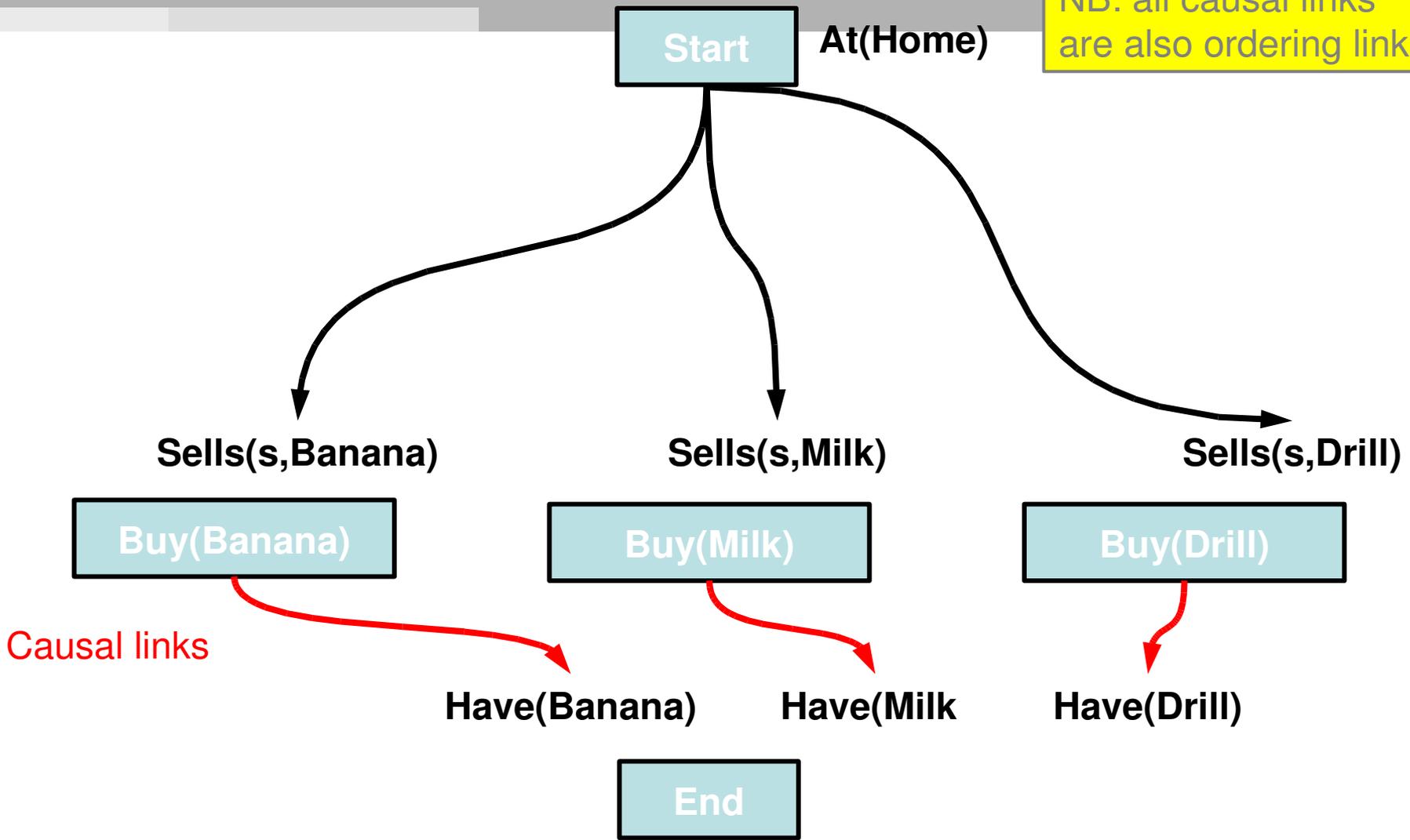
End



# Refining the Plan (1)



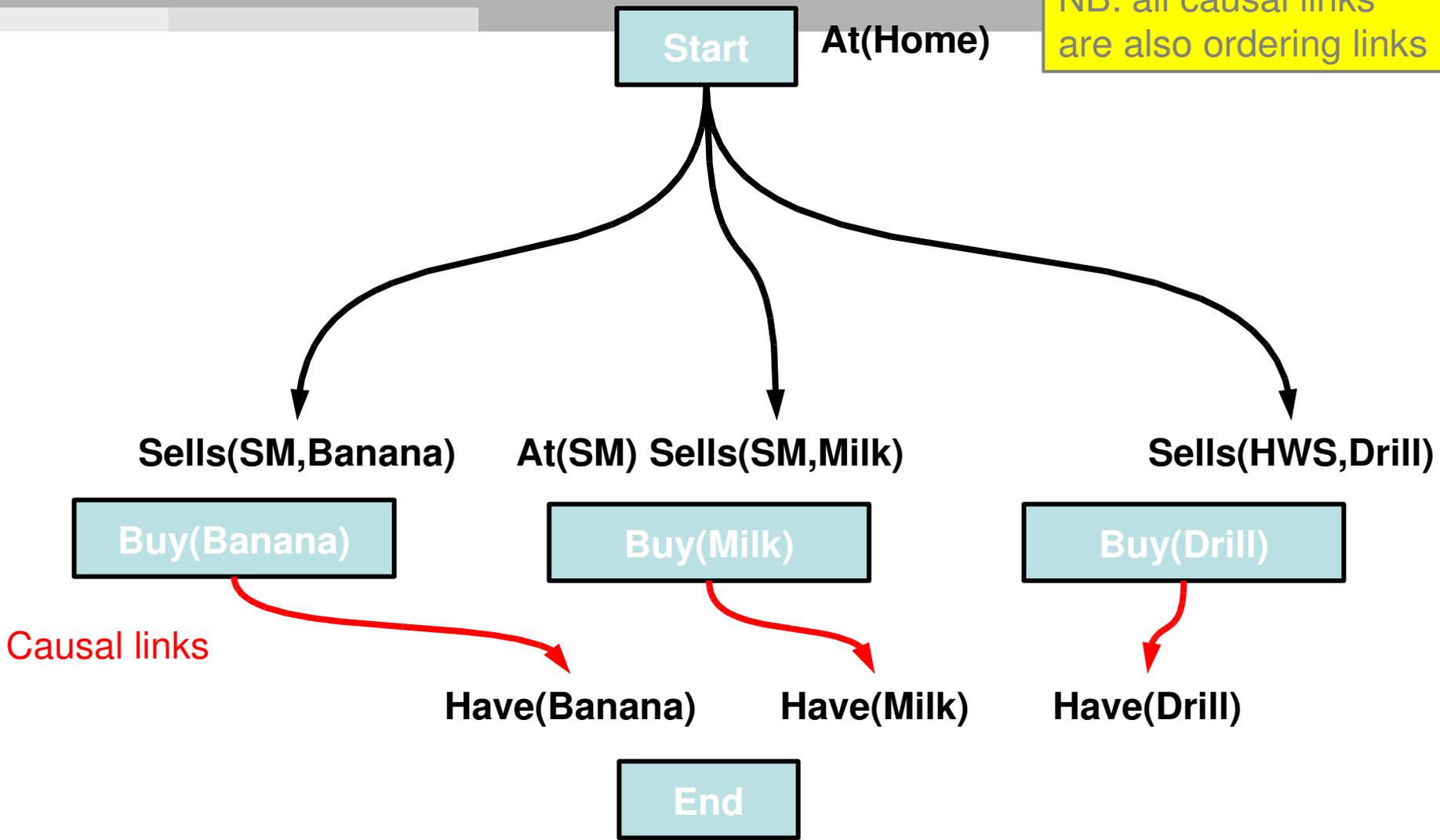
NB: all causal links are also ordering links



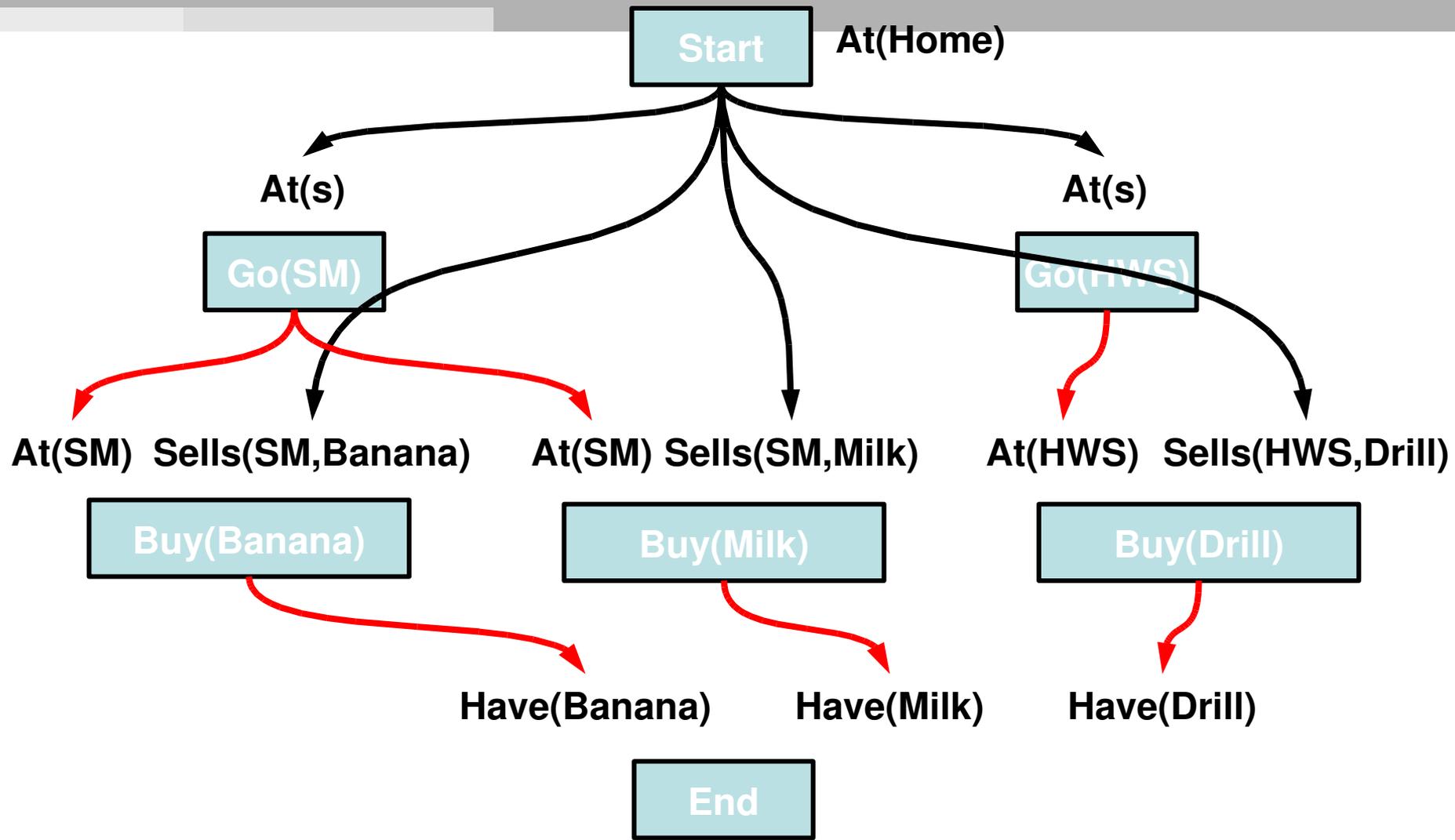
# Refining the Plan (2)



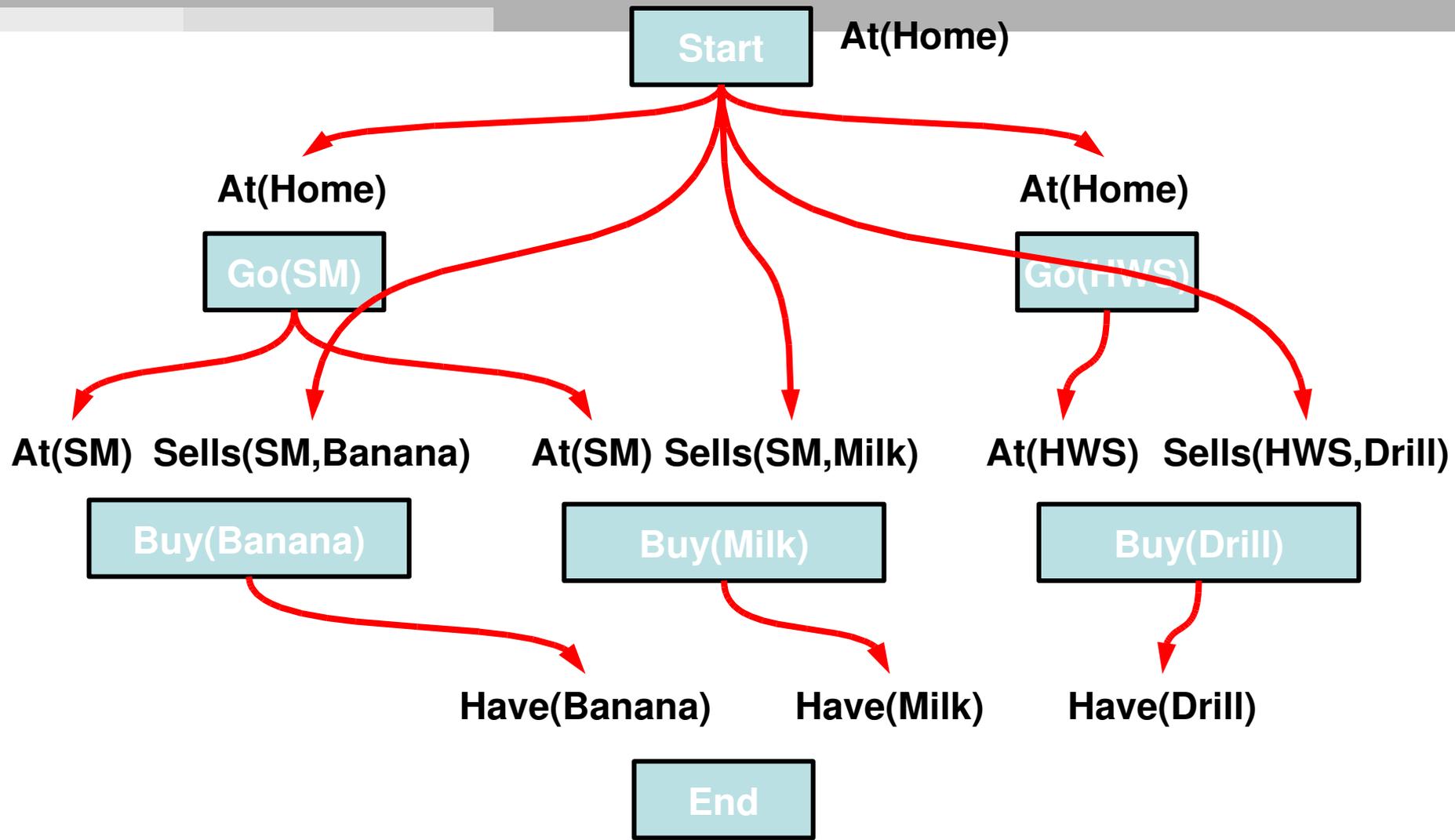
NB: all causal links are also ordering links



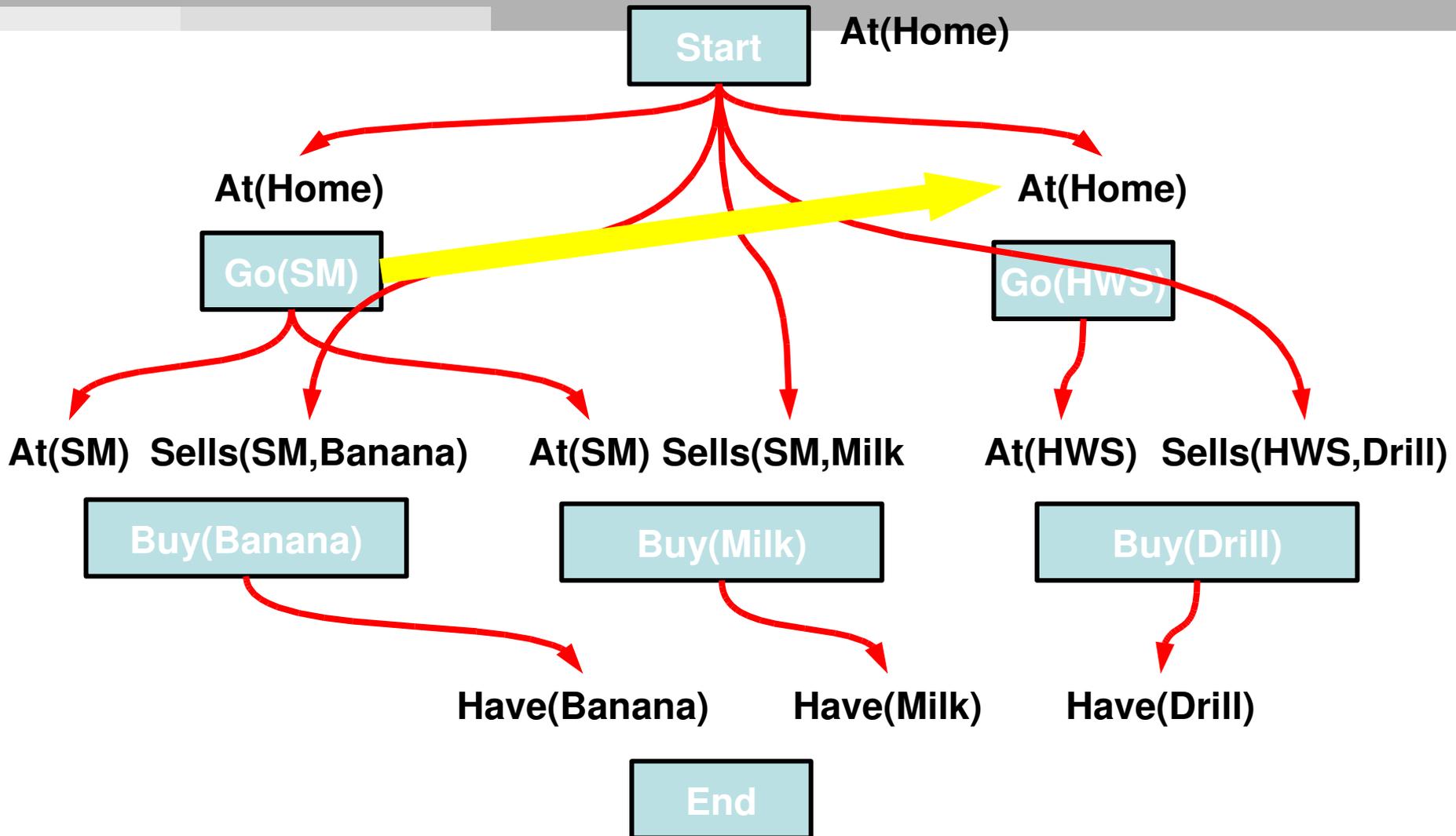
# Refining the Plan (3)



# Refining the Plan (4)



# Threats to Causal Links



# Threats to Causal Links: Demotion

