

A solution is a complete, consistent plan. ~~that starts~~  
~~goal conditions~~  $\uparrow$  every precondition at each step met  $\uparrow$  no ordering constraint violations, no variables bound to two different things  $\uparrow$  Start w/ plan

Start  $\downarrow$  Finish

must get this kind of plan

Ex)

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

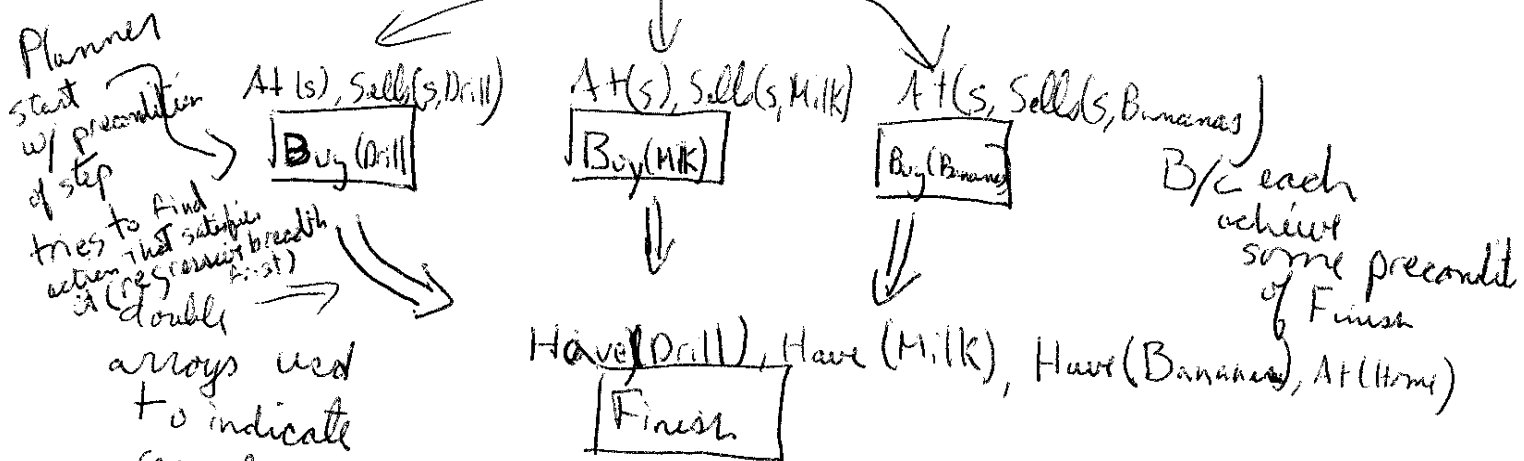
Op(Action: Finish, Precond: Have(Drill)  $\wedge$  Have(Milk)  $\wedge$  Have(Banana)  $\wedge$  At(Home))

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

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

Planner only considers adding steps that achieve a precondition that has not yet been achieved.

So in above Start

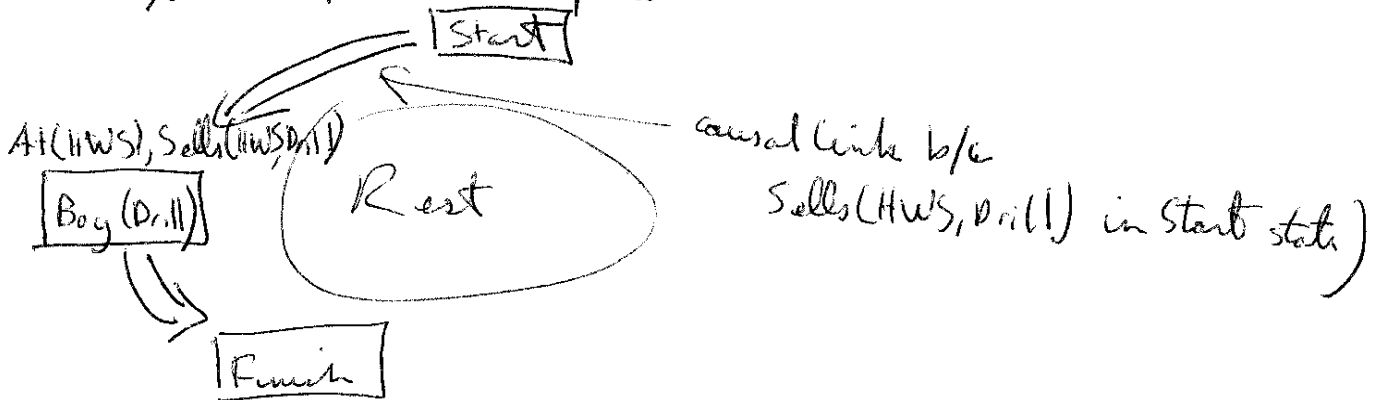


Planner start w/ precondition of step tries to find action that satisfies it (regress until double)

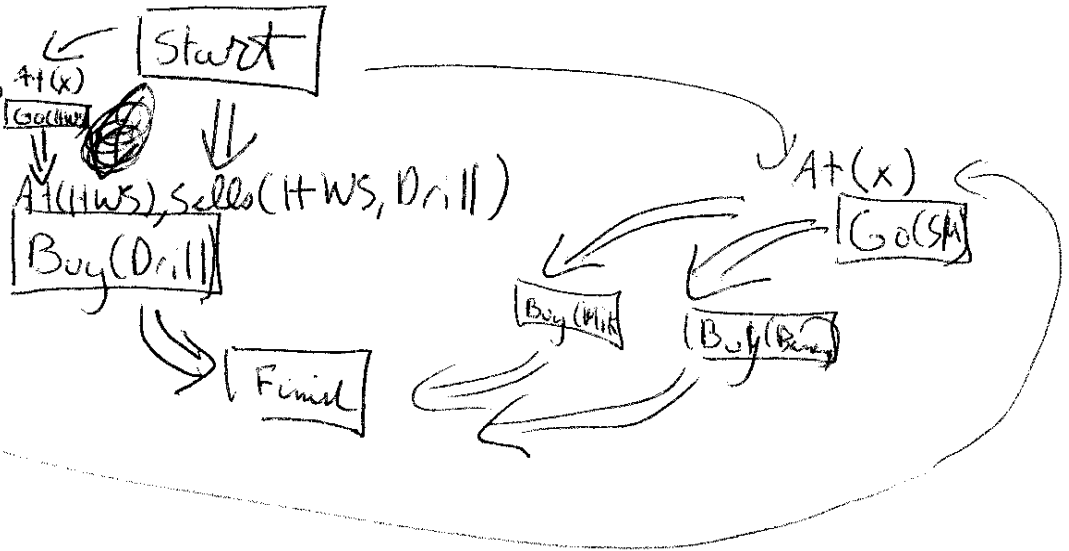
B/c each achieved some precond of Finish

arrows used to indicate causal links planner must protect if insert another thing into state

Next kind is each precondition



Try to satisfy other preconditions

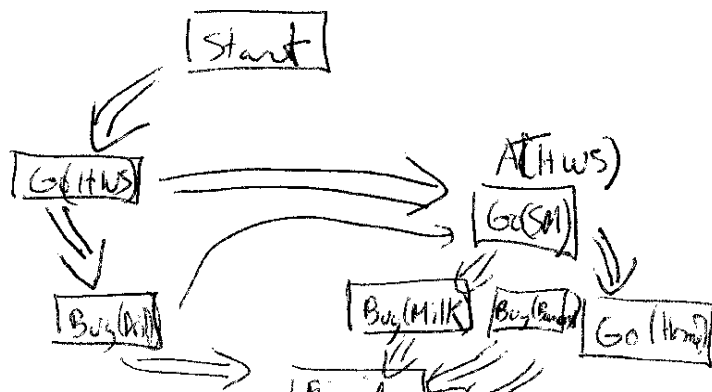


Initially might both kind to home

but then have problem b/c if go to HWS no longer at Home so couldn't go (SA).

So two actions are threats to each other.

Solution must put one after other (promotion) or before other (demotion)



Top Sort Graph to get final Plan