Artificial Intelligence
Encoding Planning Problems

Nilsson - Chapter 22

Russell and Norvig - Chapter 11

this starts our excursion into
“planning”

initial state

you are at home and don’t have milk, bananas, and a drill

goal state

you are at home and have milk, bananas, and a drill

first-order logic (= situation calculus)

initial state

At(Home, s0)
AND NOT Have(Milk, s0)
AND NOT Have(Bananas, s0)
AND NOT Have(Drill, s0)

goal state (= query)

EXISTS s
At(Home, s)
AND Have(Milk, s)
AND Have(Bananas, s)
AND Have(Drill, s)

operators

FORALL a, s
Have(Milk, Result(a,s))
EQUIV
a = Buy(Milk) AND At(Supermarket, s)
OR
Have(Milk, s) AND NOT a = Drop(Milk)

problems with first-order logic

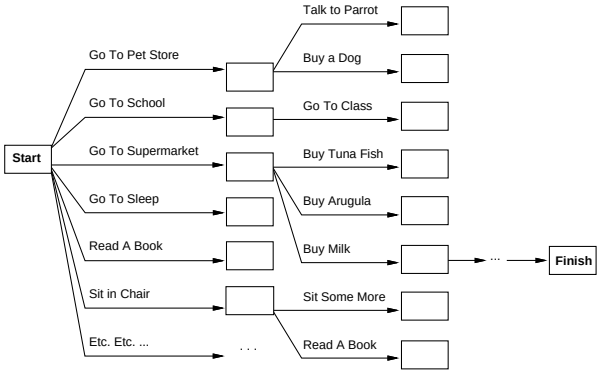
- inefficient
- does not necessarily generate a GOOD plan

- the frame problem
  problem of specifying what an action does NOT change

- the qualification problem
  problem of specifying when an action can be applied

- the ramification problem
  problem of specifying what changes an action makes implicitly

search



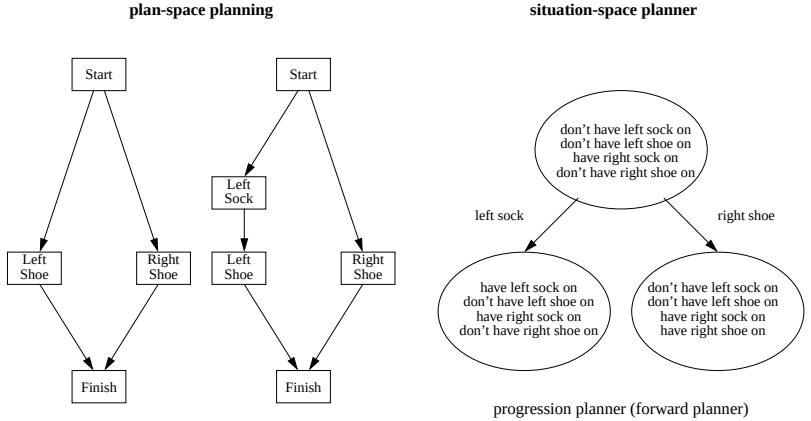
problems with search

- forward search
  1) decide that we need to buy milk, 2) decide how to do that
  -> means-ends planner

- states are not atomic
  looking at the start and goal states suggests that we need to buy milk

- most parts of the world are independent of the other parts
  1) plan how to buy the three items, 2) combine the plans

plan-space planner versus situation-space planner



refinement planner

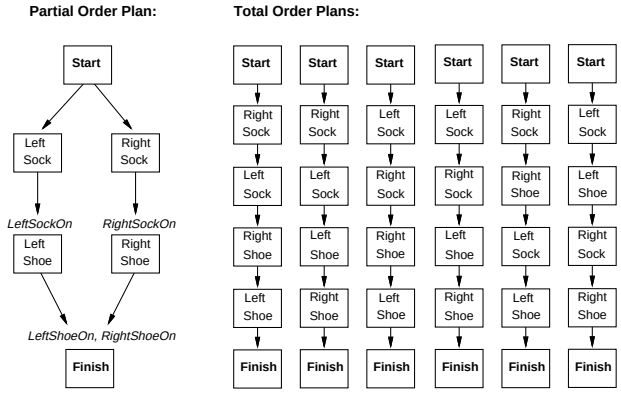
debugging planner

progression planner (forward planner)

regression planner (backward planner)

means-ends planner

partial-order planner versus total-order planner



least-commitment planner

STRIPS

start

At(Bob, Room1)
At(Box, Room2)
Pushable(Box)

goal

At(Box, Room1)

operators:

Go(x,y):
PRECOND: At(Bob,x)
EFFECT: At(Bob,y) AND NOT At(Bob,x)

Push(b,x,y):
PRECOND: At(b,x) AND At(Bob,x) AND Pushable(b)
EFFECT: At(b,y) AND At(Bob,y)
AND NOT At(b,x) AND NOT At(Bob,x)

example: the blocks world