Planning

Given:

- A description of the effects and preconditions of the actions
- A description of the initial state
- A goal to achieve

find a sequence of actions that is possible and will result in a state satisfying the goal.

Forward Planning

Idea: search in the state-space graph.

- The nodes represent the states
- The arcs correspond to the actions: The arcs from a state s represent all of the actions that are legal in state s.
- A plan is a path from the state representing the initial state to a state that satisfies the goal.

Example state-space graph

Actions

mc: move clockwise
mac: move anticlockwise
nm: no move
puc: pick up coffee
dc: deliver coffee
pum: pick up mail
dm: deliver mail

$$\begin{tikzcd}
 & & \boxed{\langle cs,\overline{rhc},swc,mw,\overline{rhm}\rangle} \arrow[dl, "puc"'] \arrow[d, "mc"] \arrow[dr, "mac"] & \\
 & \boxed{\langle cs,rhc,swc,mw,\overline{rhm}\rangle} \arrow[dl, "mc"'] \arrow[d, "mac"] & \boxed{\langle off,\overline{rhc},swc,mw,\overline{rhm}\rangle} \arrow[d, "mc"'] \arrow[dr, "mac"] & \boxed{\langle mr,\overline{rhc},swc,mw,\overline{rhm}\rangle} \\
\boxed{\langle off,rhc,swc,mw,\overline{rhm}\rangle} \arrow[d, "dc"] \arrow[dr, "mc"] \arrow[drr, "mac"] & \boxed{\langle mr,rhc,swc,mw,\overline{rhm}\rangle} & \boxed{\langle lab,\overline{rhc},swc,mw,\overline{rhm}\rangle} & \boxed{\langle cs,\overline{rhc},swc,mw,\overline{rhm}\rangle} \\
\boxed{\langle off,\overline{rhc},\overline{swc},mw,\overline{rhm}\rangle} & \boxed{\langle lab,rhc,swc,mw,\overline{rhm}\rangle} & \boxed{\langle mr,rhc,swc,mw,\overline{rhm}\rangle} &
\end{tikzcd}$$

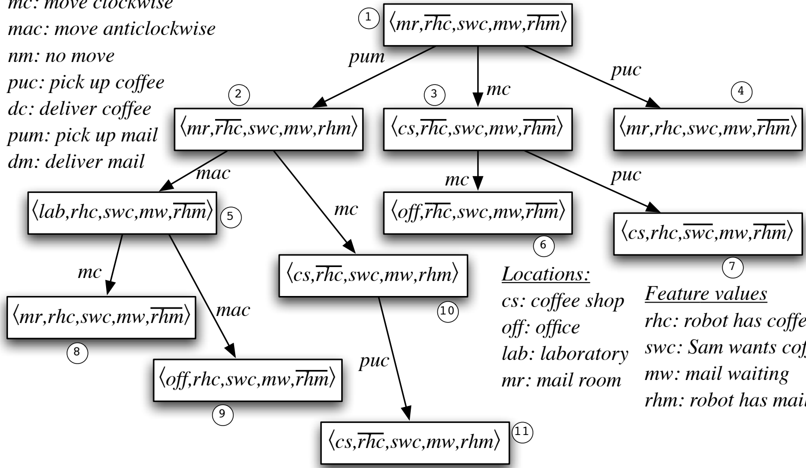
Locations:

cs: coffee shop
off: office
lab: laboratory
mr: mail room

Feature values

rhc: robot has coffee
swc: Sam wants coffee
mw: mail waiting
rhm: robot has mail

What are the errors?



1 $\langle mr,\overline{rhc},swc,mw,\overline{rhm}\rangle$
pum
mc
puc
2 $\langle mr,\overline{rhc},swc,mw,rhm\rangle$
3 $\langle cs,\overline{rhc},swc,mw,\overline{rhm}\rangle$
4 $\langle mr,rhc,swc,mw,\overline{rhm}\rangle$
mac
mc
mc
puc
5 $\langle lab,rhc,swc,mw,\overline{rhm}\rangle$
6 $\langle off,\overline{rhc},swc,mw,\overline{rhm}\rangle$
7 $\langle cs,rhc,\overline{swc},mw,\overline{rhm}\rangle$
10 $\langle cs,\overline{rhc},swc,mw,rhm\rangle$
mc
mac
puc
8 $\langle mr,rhc,swc,mw,\overline{rhm}\rangle$
9 $\langle off,rhc,swc,mw,\overline{rhm}\rangle$
11 $\langle cs,\overline{rhc},swc,mw,rhm\rangle$

Actions

mc: move clockwise
mac: move anticlockwise
nm: no move
puc: pick up coffee
dc: deliver coffee
pum: pick up mail
dm: deliver mail

Locations:

cs: coffee shop
off: office
lab: laboratory
mr: mail room

Feature values

rhc: robot has coffee
swc: Sam wants coffee
mw: mail waiting
rhm: robot has mail

Forward planning representation

- The search graph can be constructed on demand: you only construct reachable states.
- If you want a cycle check or multiple path-pruning, you need to be able to find repeated states.
- There are a number of ways to represent states:
  - As a specification of the value of every feature
  - As a path from the start state

Improving Search Efficiency

Forward search can use domain-specific knowledge specified as:

- a heuristic function that estimates the number of steps to the goal
- domain-specific pruning of neighbors:
  - don't go to the coffee shop unless "Sam wants coffee" is part of the goal and Rob doesn't have coffee
  - don't pick-up coffee unless Sam wants coffee
  - unless the goal involves time constraints, don't do the "no move" action.