NON LINEAR PLANNING: EXERCISE

Given the following initial state robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b),

We have to reach the goal: coloured(plate, blue), coloured(cube,red)

Actions are modelled as follows:

colour(Object, Location, Colour)

PRECOND: object_at(Object, Location), robot_at(Location),

robot_has(Colour)

ADD: coloured(Object, Colour)

go(X,Y)

PRECOND: robot_at(X)

DELETE: robot_at(X)

ADD: robot at(Y)

pickColour(C)

PRECOND: handempty

DELETE: handempty

ADD: robot_has(C)

releaseColour(C)

PRECOND: robot_has(C)

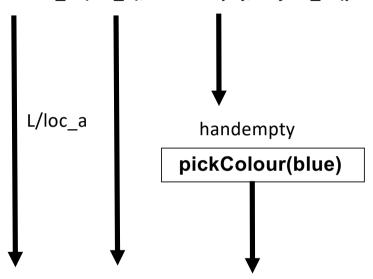
DELETE: robot_has(C)

ADD: handempty

Solve the problem with the POP algorithm, identifying threats and their solution during the process

start

robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b)



object_at(plate, L), robot_at(L), robot_has(blue)

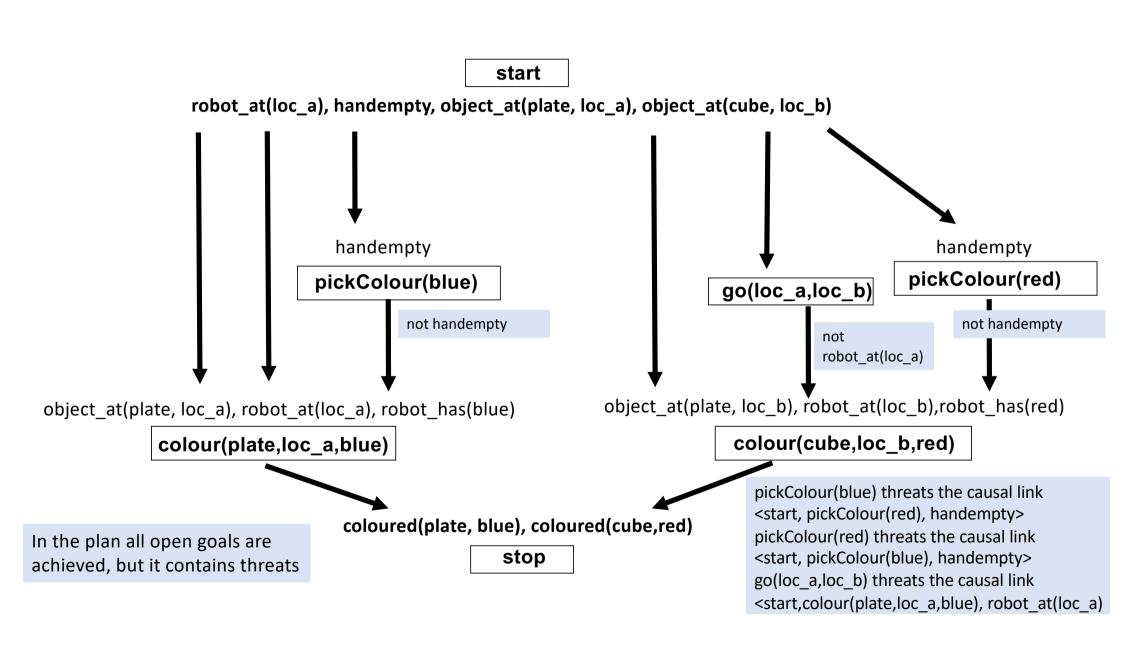
colour(plate,L,blue)

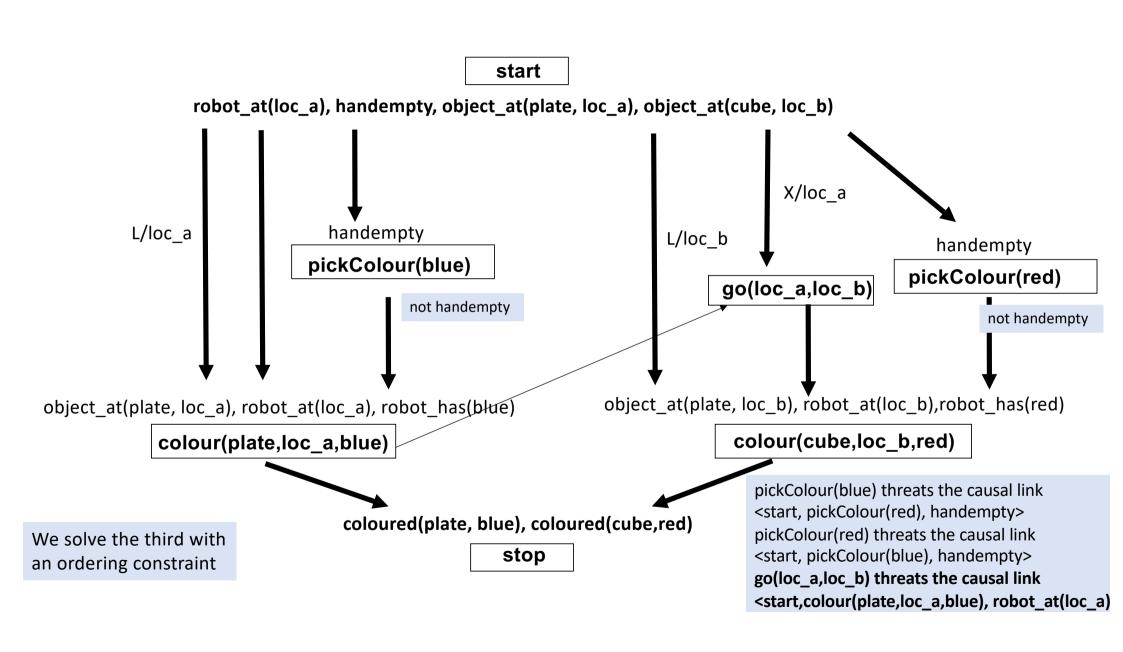
coloured(plate, blue), coloured(cube,red)

stop

start robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b) L/loc_a L1/loc_b handempty pickColour(blue) object_at(plate, loc_a), robot_at(loc_a), robot_has(blue) object_at(cube, L1), robot_at(L1), robot_has(red) colour(cube,L1,red) colour(plate,loc_a,blue) coloured(plate, blue), coloured(cube,red) stop

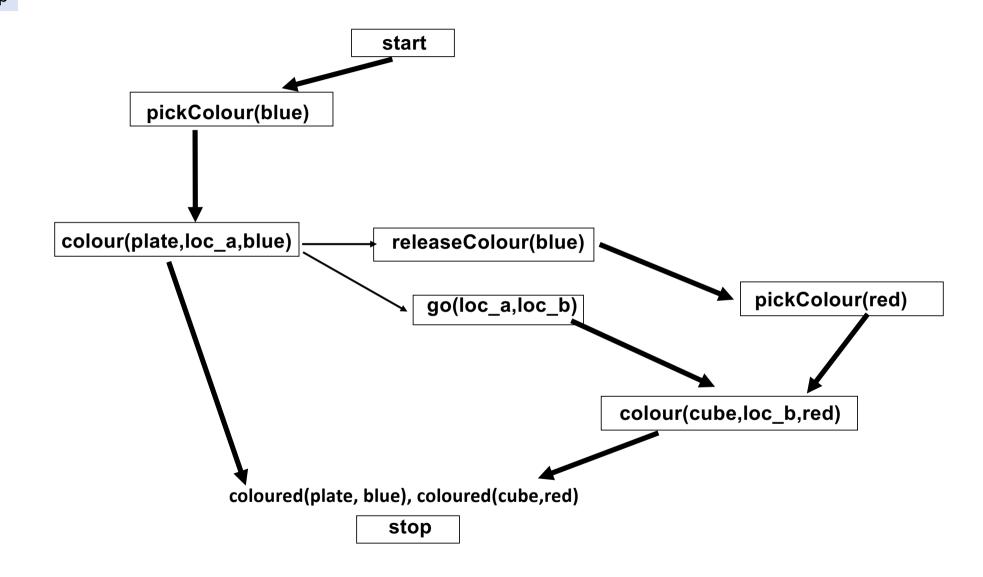
start robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b) X/loc_a L/loc_a L1/loc_b handempty handempty pickColour(blue) pickColour(red) go(X,loc_b) object_at(cube, loc_b), robot_at(loc_b),robot_has(red) object_at(plate, loc_a), robot_at(loc_a), robot_has(blue) colour(cube,loc_b,red) colour(plate,loc_a,blue) coloured(plate, blue), coloured(cube,red) stop





start robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b) releaseColour(blue) handempty handempty pickColour(blue) pickColour(red) go(loc_a,loc_b) Not handempty Not handempty object_at(plate, loc_b), robot_at(loc_b), robot_has(red) object_at(plate, loc_a), robot_at(loc_a), robot_has(blue) colour(cube,loc_b,red) colour(plate,loc_a,blue) pickColour(blue) threats the causal link coloured(plate, blue), coloured(cube,red) <start, pickColour(red), handempty> We order pickColour(blue) stop pickColour(red) threats the causal link and pickColour(red) and we have to introduce a <start, pickColour(blue), handempty> white knight

start robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b) releaseColour(blue) handempty handempty pickColour(blue) pickColour(red) go(loc_a,loc_b) Not handempty Not handempty object_at(plate, loc_a), robot_at(loc_a), robot_has(blue) object_at(plate, loc_b), robot_at(loc_b),robot_has(red) colour(cube,loc_b,red) colour(plate,loc_a,blue) Now releaseColour(blue) threats the coloured(plate, blue), coloured(cube,red) Causal link <pickColour(blue), colour(plate,loca,blue),</pre> stop We order colour(plate,loc a,blue) robot has(blue)> and releaseColour(blue)



GRAPHPLAN: EXERCISE

Given the following initial state robot_at(loc_a), handempty, object_at(plate, loc_a), object_at(cube, loc_b),

We have to reach the goal: coloured(plate, blue), coloured(cube,red)

Actions are modelled as follows:

colour(Object, Location, Colour)

PRECOND: object_at(Object, Location), robot_at(Location),

robot_has(Colour)

ADD: coloured(Object, Colour)

go(X,Y)

PRECOND: robot_at(X)

DELETE: robot_at(X)

ADD: robot_at(Y)

pickColour(C)

PRECOND: handempty

DELETE: handempty

ADD: robot_has(C)

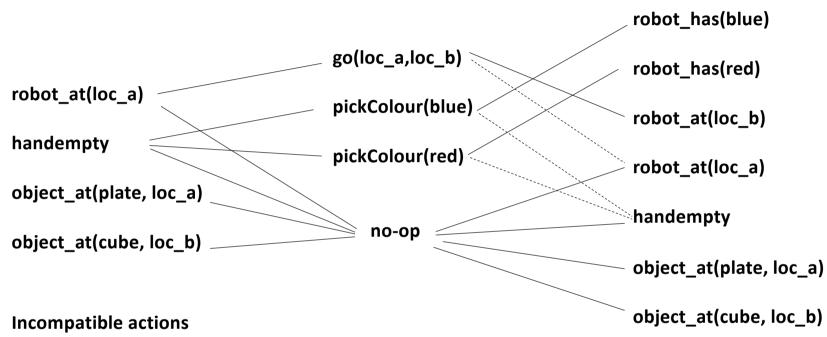
releaseColour(C)

PRECOND: robot_has(C)

DELETE: robot_has(C)

ADD: handempty

Show two levels of graphplan for this problem showing incompatible actions and incompatible propositions



Interference:

pickColour(blue) and pickColour(red) pickColour(blue) and no op on handempty pickColour(red) and no op on handempty go(loc a,loc b) and no op on robot at(loc a)

Incompatible propositions

robot_at(loc_a) and robot_at(loc_b)
robot_has(blue) and handempty
robot_has(red) and handempty