16-662 Autonomy Homework 3

David Hill

9 April 2024

1 New Action Code Snippets

Figure 1 is a code snippet of the three new actions. The first action is for moving the robot from the kitchen to the pantry. The second action is for moving the robot from the pantry to the kitchen. The third action is for slicing fruit in the kitchen.

```
### Move to Pantry
Precond=np.zeros([nrObjects, nrPredicates])
Precond[0][1]=1  # Robot in the kitchen
Precond[0][5]=-1 # Robot not in the pantry

Effect=np.zeros([nrObjects, nrPredicates])
Effect[0][1]=-2. # Robot not in the kitchen
Effect[0][5]=2.  # Robot in the the pantry

ActionPre.append(Precond)
ActionEff.append(Effect)
ActionDesc.append("Move to InPantry from InKitchen")

### Move from Pantry
Precond=np.zeros([nrObjects, nrPredicates])
Precond[0][5]=1  # Robot in the pantry
Precond[0][1]=-1 # Robot not in the kitchen

Effect=np.zeros([nrObjects, nrPredicates])
Effect[0][5]=-2. # Robot not in the pantry
Effect[0][1]=2.  # Robot in the the kitchen

ActionPre.append(Precond)
ActionEff.append(Effect)
ActionDesc.append("Move to InKitchen from InPantry")

###Cut fruit in kitchen
for j in [1,2]:
    Precond=np.zeros([nrObjects, nrPredicates])
    Precond[0][1]=1  # Robot in the kitchen
    Precond[j][1]=1  # Fruit j in the kitchen
    Precond[4][1]=1  # Knife in the kitchen
    Precond[j][6]=-1  # Fruit is not chopped

    Effect=np.zeros([nrObjects, nrPredicates])
    Effect[j][6]=2  # Fruit is chopped

    ActionPre.append(Precond)
    ActionEff.append(Effect)
    ActionDesc.append("Cut "+Objects[j]+" in the kitchen")
```

Figure 1: Code snippet of the three new actions.

2 Path Planning Results

Figure 2 shows the results of planning a path using both Dijkstra's algorithm and the A* algorithm. Dijkstra's takes much longer and must explore 4545 states in order to find a path with 16 actions. This is guaranteed to be optimal. A* is faster, only exploring 1835 states before finding a solution with 16 steps. The solution A* produces is optimal in this case, but it is not guaranteed to be optimal in general.

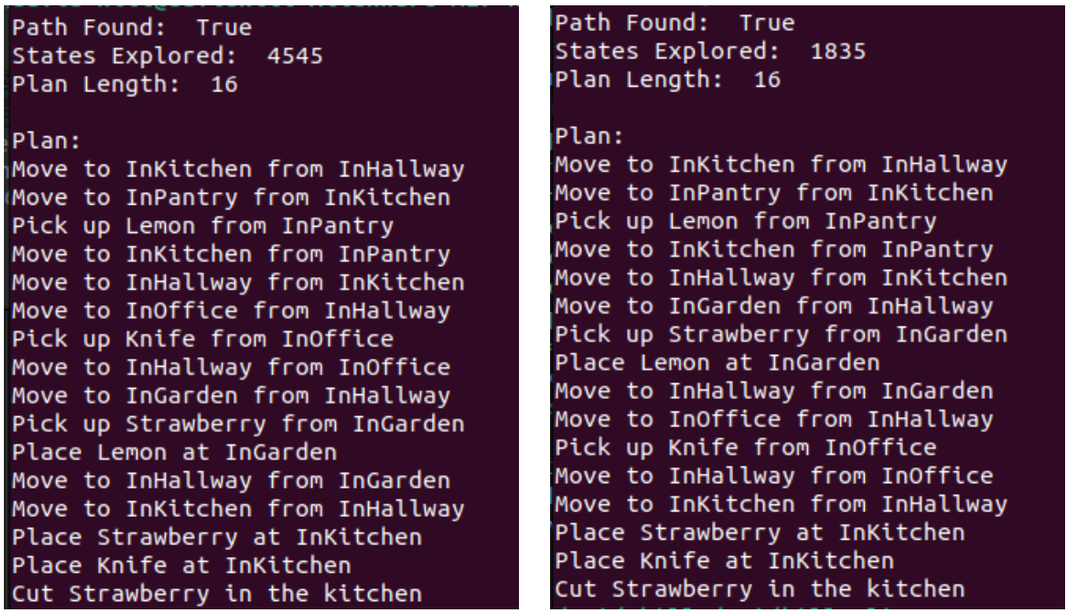

Figure 2: Results after running Dijkstra's (left) and A* (right) algorithms.