

HW4 - Robot Autonomy Plus

Professor: Oliver Kroemer, TA's: Steven Lee, Ami Sawhney

Due: 2/15/21 at 11:59pm Eastern Daylight Time (GMT-4))

1 Task Planning/Discrete Search

For this portion of the homework, you will be implementing a discrete planner for a task planning problem. This homework does not require the use of the PyRobot API like in the previous homework assignments. We have reviewed discrete search methods, such as dijkstra's algorithm, in previous homework assignments and lecture (e.g. Lecture 8). You can find more information on task planning in Lecture 11 (Task Planning). As discussed in lecture, the goal of task planning is to model and plan sequences of skills and actions to accomplish a complex task. When executing multiple skills, you need to ensure that the current state is suitable for executing the next skill. This requires representing the state and actions.

This assignment: The **SearchDiscrete.py** file contains the skeleton code for this problem and is where your implementation will go.

We provide you with a list of predicates and objects that you will need for defining the states of the task planning problem. The predicates are as follows: 'InHallway', 'InKitchen', 'InOffice', 'InLivingRoom', 'InGarden', 'InPantry', 'Chopped', and 'OnRobot'. Additionally, the objects that can be used to define the state are: 'Robot', 'Strawberry', 'Lemon', 'Paper', and 'Knife'. The predicates can be used to describe the status of the individual objects and the state of the environment can be described using the status of each of the objects. Additionally, we have provided 3 utility functions that you may find helpful for your discrete planner implementation: **CheckCondition()**, **CheckVisited()**, **ComputeNextState()**.

One difficulty in task planning is properly defining the actions. We have provided you with 4 of the actions for the task planning problem (**Move to hallway**, **Move to room**, **Pickup object**, and **Place object**), but you will need to define the remaining 3:

- **Move to Pantry**
- **Move from Pantry**
- **Cut fruit in kitchen**

To properly define these actions you will need to define all of the preconditions and the effects for each of the 3 actions. You can use the 4 actions provided to you as references.

Once you have defined the actions, you will need to implement your discrete planner that takes a given initial state and provides the actions required to reach the goal state. The initial state and the goal state are defined in the **SearchDiscrete.py** file. Feel free to optimize your planner for what you feel is best (e.g. you can optimize your discrete planner to provide a plan with the shortest number of actions).

Deliverables:

- The completed SearchDiscrete.py file containing the 3 extra actions and the discrete planner.
- Provide the actions you took to get from the initial state to the goal state.