Planning and Approximate Reasoning (2nd call)

PART 1: Planning

January, 23th, 2023

Time: 90 min. You cannot use any printed or electronic materials. Please give clear and detailed answers (with examples, if appropriate). All the answers must be properly justified to be valid. You should show your work and the necessary calculations for the full credit.

---

1. Classical Planning and High Level Representation

(1 point) Explain briefly the idea of the problem solver of Means-Ends analysis (MEA).

(1 point) Non-linear Planning (N-LP) uses a goal set instead of a goal stack and it includes in the search space all possible subgoal orderings. Explain in details the advantages and the disadvantages of N-LP, and give an example of N-LP algorithms.

(1 point) With state-space planning, we generate plans by searching through state space using Forward or Backward searching. Explain the difference between the two types of searching.

2. PDDL Implementation

(2 points) For the cleaning task of an indoor environment, we want to use a PDDL planner to develop a model of a robotic cleaner (agent) that deals with the environment features and constraints. The environment is a set of *n* adjacent rooms. The main goal of the robot is to move from a source room to a destination room to clean it, under a constraint that is the two rooms of the source and destination have to be adjacent.

In this example, to make the planning problem less complex, we simplify it by defining a PDDL domain containing only three predicates and two actions.

The domain has only one type of objects:

```
(:types
        room
 )
```

The three predicates to solve the robotic cleaner planning problem are:

```
(:predicates
        (is-in ?r - room)                          ;the robot is in the room ?r
        (adjacent ?r1 - room ?r2 - room))          ;room ?r1 is adjacent to room ?r2
        (clean ?r - room)                          ;room ?r is cleaned
  )
```

Besides, there are two actions, you need to select from them to update the world state.

Action	Usage
move (?from - room, ?to - room)	Action "**move**" for the robot to move from room 1 (source) to room 2 (destination).
suck (?r - room)	Action "**suck**" for the robot to achieve the cleaning of the room.

Based on the problem described above, write the declaration of actions ***move*** and ***suck*** in PDDL showing the parameters, preconditions, and effects including add and delete lists.

3. Markov Decision Process (MDP)

Consider the Markov decision process (MDP) with two states $s \in \{s_0, s_1\}$, two actions $a \in \{right, left\}$, discount factor $\lambda = 0.9$, and the reward, R, of each state is shown below:

s	***R(s)***
s_0	-1
s_1	+2

And the probability of transition between each pair of states for the right and left actions as follows:

s	**s´**	***P(s´|s ,a=right)***
s_0	s_0	0.25
s_0	s_1	0.75
s_1	s_0	0.25
s_1	s_1	0.75

s	**s´**	***P(s´|s, a=left)***
s_0	s_0	0.75
s_0	s_1	0.25
s_1	s_0	0.75
s_1	s_1	0.25

(1 point) (a) consider the policy π that chooses the action $a = right$ in each state. For this policy, solve the linear system of Bellman equations (by hand) to compute the state-value function $V_\pi(s)$ for $s \in \{ s_0, s_1\}$. <u>You should fill the following Table and show the necessary calculations for full credit.</u>

s	***π(s)***	***$V^\pi(s)$***
s_0	*right*	
s_1	*right*	

(1 point) (b) Compute the greedy policy $\pi'(s)$ with respect to the state-value function $V_\pi(s)$ computed in part (a). <u>You should fill the following table and show the necessary calculations for full credit.</u>

s	***π(s)***	***π'(s)***
s_0	*right*	
s_1	*right*	

A goal without a plan is just a wish.