Planning and Approximate Reasoning

PART 1: Planning

January, 24th, 2022

Time: 2 h. 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

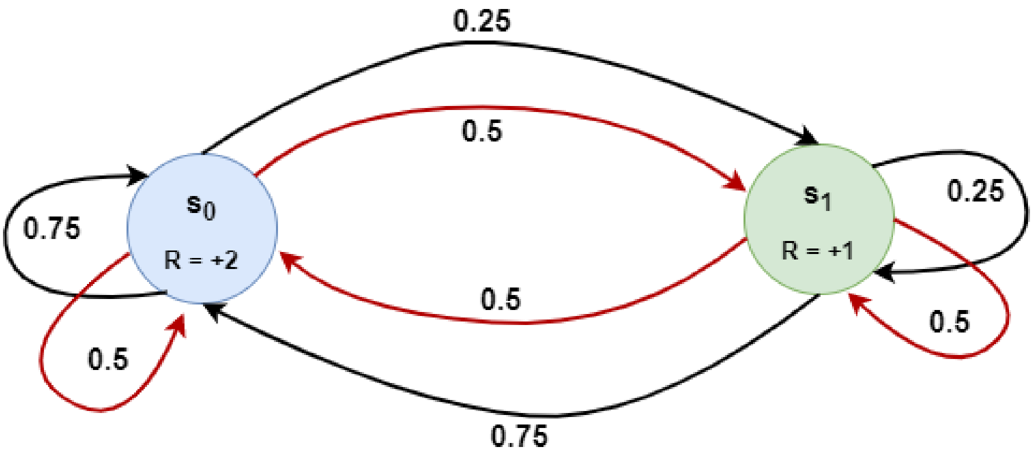
(1 point) Explain briefly the idea of the problem solver of Means-Ends analysis (MEA).

(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 for N-LP algorithms.

(1 point) There are two main approaches to planning: state-space planning and plan-space planning, sometimes called partial-order planning. Explain the difference between the two plaining approaches.

2. Markov Decision Process (MDP)

Consider the Markov decision process (***MDP***) with two states $s \in \{s_0, s_1\}$, with two actions $a \in \{$red, black$\}$, discount factor $\lambda = 0.8$. The reward, R, of each state and the probability of transition between each pair of states are shown in the following figure:



(1 point) (a) consider the policy π that chooses the action a = *red* 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 $s \in \{s_0, s_1\}$. You should fill the following Table and show the necessary calculations for full credit.

s	π(s)	V^π(s)
s_0	red	
s_1	red	

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

s	π(s)	π'(s)
s_0	red	
s_1	red	

3. 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. The main goal of the robot is to move from a source location to a destination to clean it. In this example, to make the planning problem less complex, we simplify it by defining a PDDL domain containing only two predicates and two actions.

The domain has two types of objects, such as:

```
(:types
        location
        robot
 )
```

In addition, in the simplest case, there are two predicates to solve the robotic cleaner planning problem:

```
(:predicates
        (at ?r - robot ?l - location)           ; robot r? is at location l?
        (clean ?l - location)                   ; location ?l is clean
 )
```

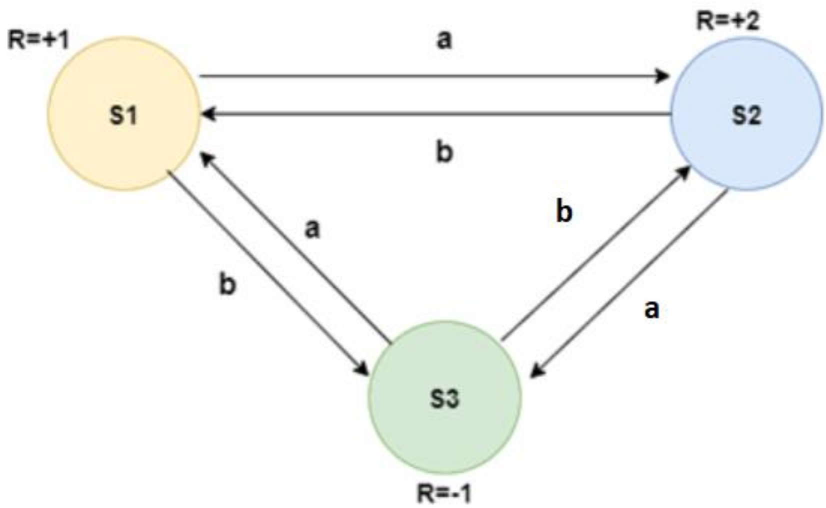
Besides, there are two actions, you need to select from them to update the world state.

Action	Usage
move (?r - robot, ?l1 - location1, ?l2 - location2)	Action "***move***" for the robot to move from location 1 (source) to location 2 (destination).
cleanout (?r - robot, ?l - location)	Action "***cleanout***" for the robot to achieve the cleaning of the destination.

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

4. Reinforcement Learning (RL)

(3 points) Assume the following world, given in the figure above, with three states {S1, S2, S3} and two actions {a, b}.



The current Q-table contains initially the following values:

	Value
Q(S1,a)	0
Q(S1,b)	0
Q(S2,a)	0

Q(S2,b)	0
Q(S3,a)	0
Q(S3,b)	0

To use the matrix Q, the agent simply traces the sequence of states, from the initial state to goal state. The algorithm finds the actions with the highest reward values recorded in the Q table for the current state using the following training rule:

$$\boxed{\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')}$$

where s is the current state, *s´* is the next state, *a* is the current action, *a´* is the next action, *r* is the initial reword and γ is the discount factor.

Update the Q-table after 3 time-steps with γ = 0.5. Show the necessary calculations for full credit.

A goal without a plan is just a wish.