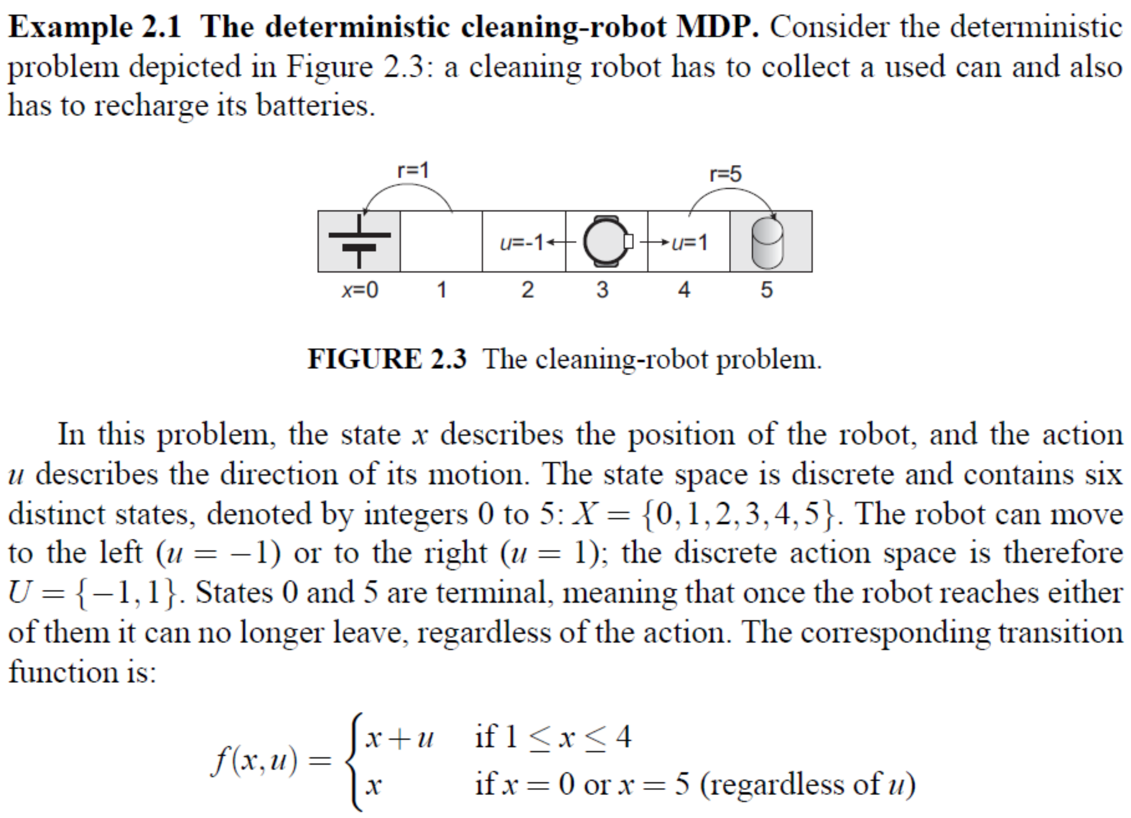
Reinforcement Learning: Programming Exercise #3

In this programing exercise we are going to solve the problem explained in Example 2.2 of the book “Reinforcement learning and dynamic programming using function approximators” using Monte-carlo method. (You don’t really need to refer to the book, everything needed is provided here, the book is available also in pdf online)

The problem is as follows:



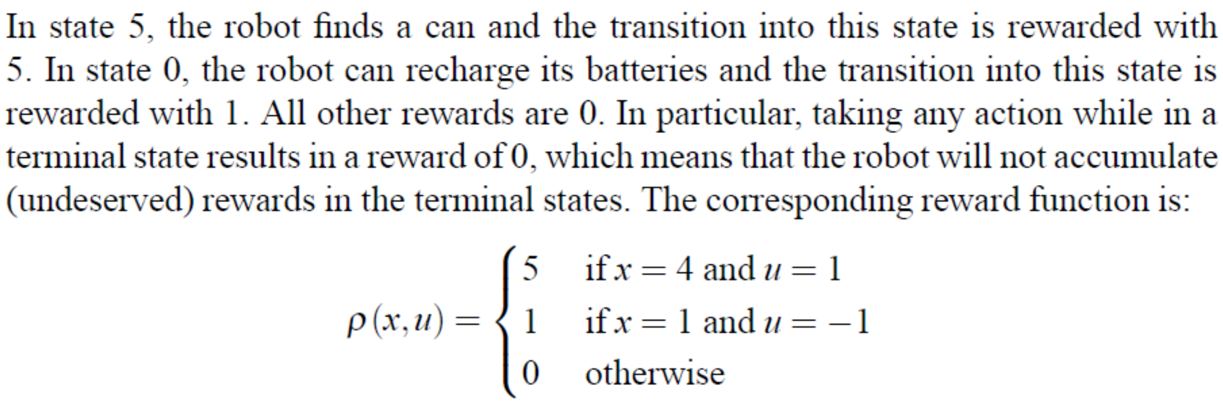
Example 2.1 The deterministic cleaning-robot MDP. Consider the deterministic problem depicted in Figure 2.3: a cleaning robot has to collect a used can and also has to recharge its batteries.

r=1
r=5
u=-1
u=1
x=0
1
2
3
4
5

FIGURE 2.3 The cleaning-robot problem.

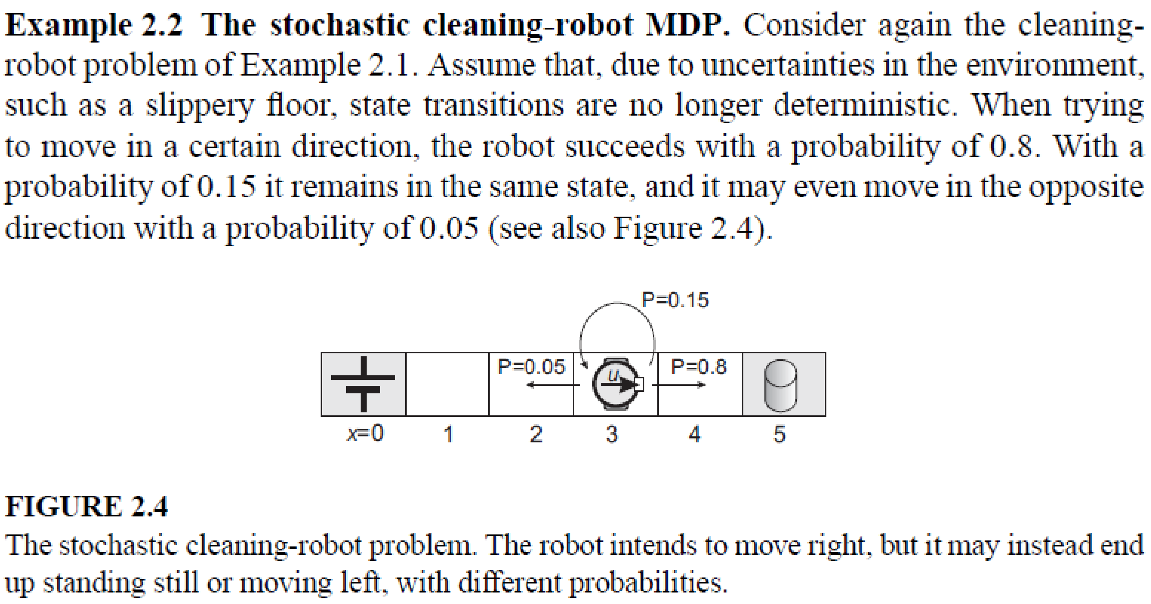
In this problem, the state x describes the position of the robot, and the action u describes the direction of its motion. The state space is discrete and contains six distinct states, denoted by integers 0 to 5: $X = \{0,1,2,3,4,5\}$. The robot can move to the left ($u = -1$) or to the right ($u = 1$); the discrete action space is therefore $U = \{-1,1\}$. States 0 and 5 are terminal, meaning that once the robot reaches either of them it can no longer leave, regardless of the action. The corresponding transition function is:

$$f(x,u) = \begin{cases} x+u & \text{if } 1 \leq x \leq 4 \\ x & \text{if } x = 0 \text{ or } x = 5 \text{ (regardless of } u\text{)} \end{cases}$$



In state 5, the robot finds a can and the transition into this state is rewarded with 5. In state 0, the robot can recharge its batteries and the transition into this state is rewarded with 1. All other rewards are 0. In particular, taking any action while in a terminal state results in a reward of 0, which means that the robot will not accumulate (undeserved) rewards in the terminal states. The corresponding reward function is:

$$\rho(x,u) = \begin{cases} 5 & \text{if } x = 4 \text{ and } u = 1 \\ 1 & \text{if } x = 1 \text{ and } u = -1 \\ 0 & \text{otherwise} \end{cases}$$



Example 2.2 The stochastic cleaning-robot MDP. Consider again the cleaning-robot problem of Example 2.1. Assume that, due to uncertainties in the environment, such as a slippery floor, state transitions are no longer deterministic. When trying to move in a certain direction, the robot succeeds with a probability of 0.8. With a probability of 0.15 it remains in the same state, and it may even move in the opposite direction with a probability of 0.05 (see also Figure 2.4).

FIGURE 2.4
The stochastic cleaning-robot problem. The robot intends to move right, but it may instead end up standing still or moving left, with different probabilities.

(Note: since the deterministic cleaning robot can get stuck in a loop in some random policies and the episode never ends, we are going to use stochastic cleaning-robot MDP in example 2.2 instead)

Since this problem is episodic, we can use Monte-carlo technique.

Obviously, the MC algorithm should not know about transition dynamics. The transition dynamics is for you to simulate the robot. The MC algorithm is only exposed to actions and rewards.

You need to calculate the optimal action value function and optimal policy using the following algorithms:

1. Monte-calro with exploring start (page 99 of Sutton&Barto)
2. On-policy first visit monte-carlo (page 101 of Sutton&Barto)

Plot the estimate of state-value function for each state as a function of number of episodes in each algorithm and comment on the convergence of the algorithm.