Artificial Intelligence - Homework 3

Domenico Santone - mat. 288640

Reinforcement Learning Module A.A.2022/2023

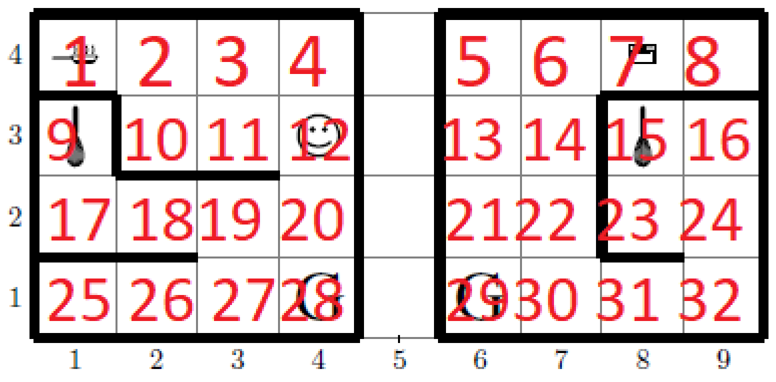
The Cooking Chef problem

Part A

Disclaimer: as specified in the exam text (*An episode will end when the agent successfully cooks the scrambled eggs (see the above description)*) I considered the cases only for the scrambled egg recipe. If the recipes are considered both the number of cases will increment accordingly and also a new nomenclature in the set of states should be introduced.

a) The set of states S in this Markov Decision Process (MDP) represents a chef's possible positions within a grid-like world. So we can have 32 states without the beater and 32 states with the beater so a total of 64 states. Therefore the cardinality of S is 64 $\rightarrow |S| = 64$. We can identify the states like this:

- $Beater_1$ to $Beater_{32}$ where $Beater \in \{W, B\}$
  - W stands for without beater, B stands for with beater and the number indicates the progressive numeration of the cell.
  - In the following image is defined the numeration followed also for the transition matrices:



b) The set of actions, A, includes all the possible directional actions that the chef can take. It also includes the action of taking the egg beater when the agent reaches one of the two states that contain it. Lastly, the agent can also perform the actions of traversing through the gates in either direction. Thus, the set of actions is represented as:

```
A = {RIGHT, LEFT, UP, DOWN, TAKE, GOLEFT, GORIGHT}
```

and there are a total of 7 possible actions. So $|A| = 7$.

c) As previously stated, the number of states in the problem is $|S| = 64$, and the number of actions is $|A| = 7$. Thus, the dimensionality of the transition function P will be 64x64x7, indicating that for each state s and each action a, there is a probability of transition to each of the 64 possible states. So P is a 64x64x7 matrix, where the rows correspond to the starting states, the columns correspond to the resulting states and the third dimension corresponds to the action.

d) Inside the attached folder there are the transition matrices inside the file TMA.xlsx In the table where the cells contains a probability of zero they're empty to improve the visibility.

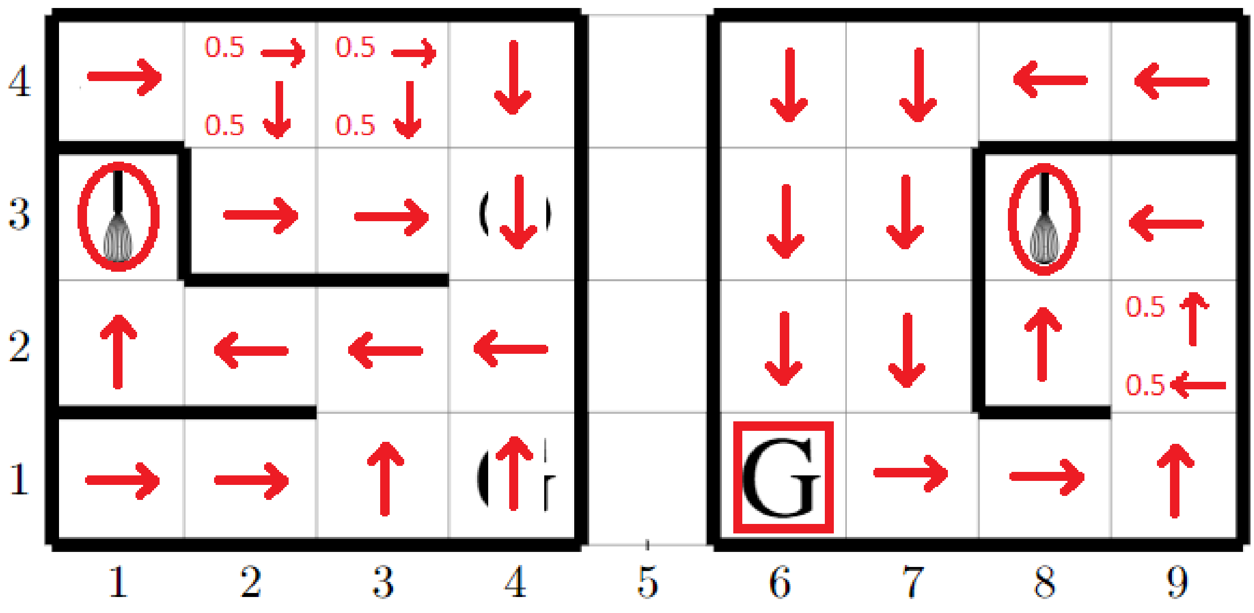
e) The reward function is a mechanism that assigns a numerical value to each combination of state, action, and resulting state. It helps to define the objective of the agent and guides its behavior, by providing a score for each transition. In the specific case of our chef agent, we want it to prioritize picking up the egg beater before heading to the cooking area. To encourage this behavior, we can assign negative values to all states except the ones where the egg beater is located, a positive value for those states, and a higher positive value for the final state (obviously this depends on the recipe we want to cook) where the chef reached the cooking area with the tool. The value of γ is a parameter that is used to balance the trade-off between short-term and long-term rewards. It helps to converge to the optimal solution by giving more or less weight to rewards obtained in the future. A good value for γ is 0.8 to converge fast.

f) No, a value of gamma within the range of (0,1) does not change the optimal policy since the same optimal policy can be obtained for any value in that range, and it only changes the speed convergence of the algorithm. However, it does affect how much weight is given to immediate rewards versus future rewards. A value closer to one would give more importance to future rewards, while A value closer to zero would give more importance to immediate rewards.

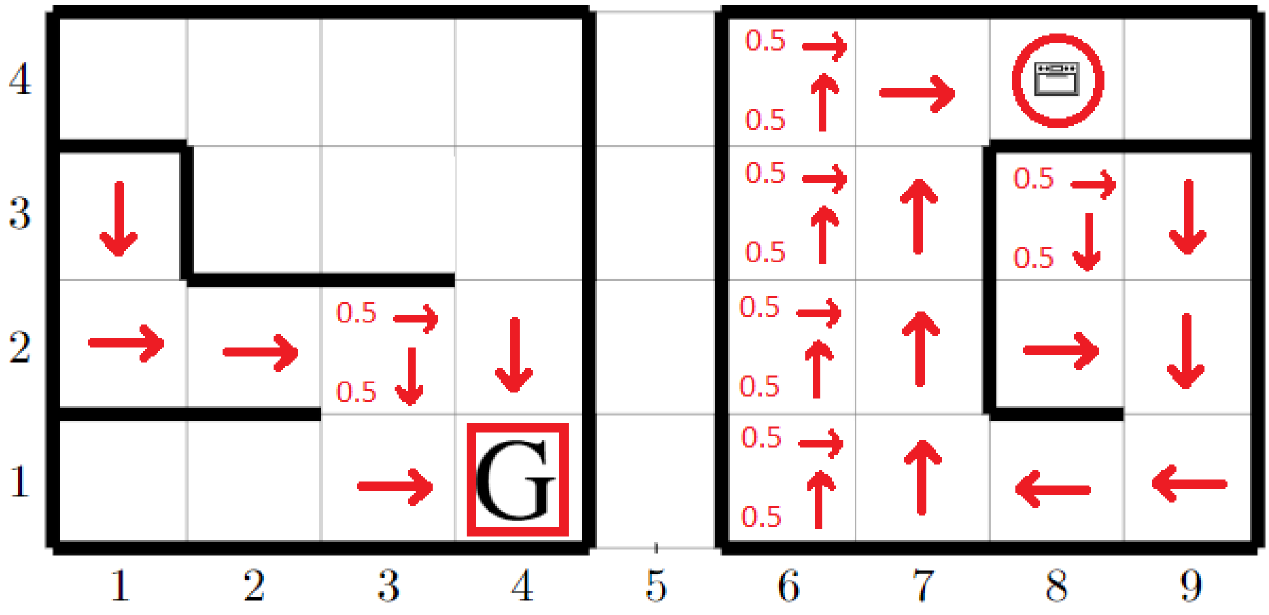
g) The agent can be in one of the 64 possible states (32 without the egg beater and 32 with the egg beater) and can take one of the 7 possible actions (up, down, left, right, take, go left, go right). So in total, there are $|A|^{|S|} = 7^{64}$ possible policies.

h) You can run the python script called `cookingchefpolicy.py` in order to visualize the results. I tried to implement a version that works well not only for pudding eggs but even for scrambled (just switch the recipe parameter). Here is the drawing of the optimal policies computed:

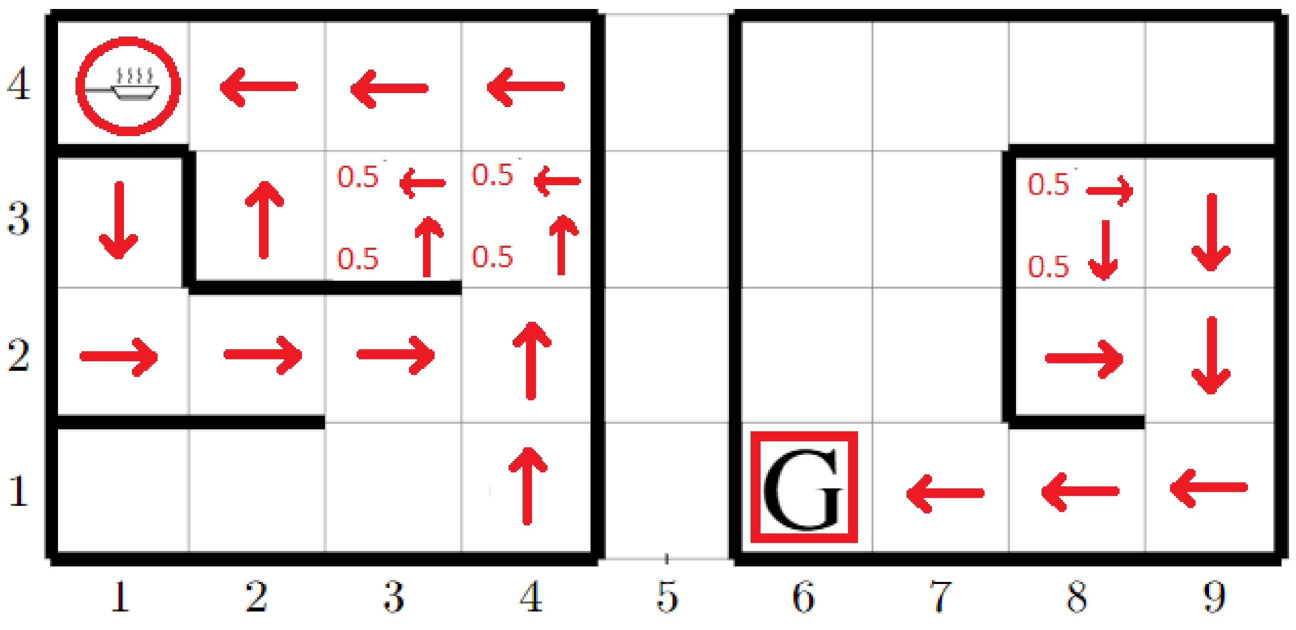
1. The first is the policy before the beater is obtained (this looks the same for both cases) some arrows have probabilities because the arrows obtained running the script several times produced the following (empirically derived). When there's a square around the G means that the agent decides to through the gate.



2. The following is the policy for the pudding, only arrows with a meaning are shown (the other are not possible due the fact that you need to pick up the beater first).



3. Same conditions as before



i) The computed policy is deterministic (the probabilities shown in the images before are just to let you know that the output produced by the program may have different directions).

j) No in this case there's no particular advantage in having a stochastic policy because we have the optimal one and we always use the shortest path to the objectives. This doesn't apply to all the possible problems so in other problems having a stochastic policy can lead to good results.

Part B

a) Inside the attached folder there are the transition matrices inside the file TMB.xlsx It's assumed that the "*perpendicular to the right*" is useful only for walking so TAKE, GOLEFT, GORIGHT are not taken into account. In the table where the cells contains a probability of zero they're empty to improve the visibility.

b) The optimal policy might change because the agent can now end up in a different state than intended due to the probability of going in the wrong direction. The agent would have to take into account the probability of going in the wrong direction when choosing the next action, in order to maximize the expected reward.

c) The value of the optimal policy could change as well, because the agent is now less likely to end up in the intended state when taking a certain action. The expected reward for each action would change, and the agent would need to adjust its policy to take into account the new probabilities.