Adam Bouafia, Matricula : 293137 , ASE Master

The Cooking Chef Problem

Part A)

a) Provide a concise description of the states of the MDP. How many states are in this MDP? (i.e. what is |S|):

Answer:

States (S):
The states in an MDP represent all the possible situations the agent can be in. In the Cooking Chef Problem, a state can be defined by the agent's position on the grid, whether the agent has the eggbeater, and which cooking station the agent has reached. Since the agent can be in any cell, we have $9 \times 4 = 36$ possible positions. The agent can have two states concerning the eggbeater (has it or doesn't have it), and three concerning the cooking station (not reached, reached frying pan, reached oven). However, once cooking starts, that state becomes absorbing. Thus, there are $36 \times 2 \times 3 = 216$ states, plus the absorbing state once cooking starts. Therefore, $|S| = 217$.

---

b) Provide a concise description of the actions of the MDP. How many actions are in this MDP? (i.e. what is |A|) :

Answer:

Actions (A):
The agent can move up, down, left, or right, but cannot move diagonally and cannot move through walls. Additionally, there's an action to "express will" to use the gate. So, there are 5 actions: up, down, left, right, and use gate. Therefore, $|A| = 5$.

---

c) What is the dimensionality of the transition function P?

Answer:

Transition Function (P) Dimensionality:
The transition function P would have dimensions |S|×|A|×|S|, which represents the probability of transitioning from any state to any other state given an action. In this case, the dimensionality is 217×5×217.

---

d) Report the transition function P for any state s and action a in a tabular format.

Answer:

Transition Function (P): Describing this function in a tabular format would involve detailing the probabilities of reaching every possible state from every other state for each action. It's a large matrix where most entries will be 0 since from most states, only a few actions are possible (e.g., you can't move into a wall, so transitions from a state next to a wall to the state of the wall are 0).
The answer will be included in a separate Spreadsheet.

https://docs.google.com/spreadsheets/d/1f7p0lnkbDtAPAcm8aec4_uWdNKLmaYf9IF_1zPdDhFI/edit?usp=sharing

e) Describe a reward function R: S × A × S and a value of γ that will lead to an optimal policy.

Answer:

Reward Function (R) and Discount Factor (γ):
Mathematically, the reward function R can be defined as:

$$\begin{cases} +1000 & \text{if } s' \text{ is a goal state (successful cooking)} \\ -1 & \text{otherwise} \end{cases}$$

Here, a high positive reward (+1000) is given for reaching the goal state (successful cooking), and a small penalty (-1) is assigned for every other transition. This encourages the agent to find the quickest path to the goal.

For the discount factor γ, choosing a value close to 1 (e.g., 0.9) is suitable. This balances the importance of immediate versus future rewards.

A high γ (close to 1) means future rewards are almost as significant as immediate rewards, which is appropriate for scenarios were reaching the goal efficiently is crucial.

$\gamma=0.9$
This mathematical formulation of the reward function R and the discount factor γ is designed to drive the agent towards efficient completion of the task, prioritizing the primary objective of cooking the eggs in the shortest time possible.

---

f) Does $\gamma \in (0, 1)$ affect the optimal policy in this case? Explain why.

Answer:

Impact of γ on the Optimal Policy:
In an infinite horizon problem where the task is to complete an action as quickly as possible, γ does not significantly affect the optimal policy, as long as it's less than 1. This is because the optimal policy is simply the shortest path to the goal, and that doesn't change whether future rewards are discounted heavily or lightly. However, if γ was exactly 1, it could potentially make the problem ill-defined if there are cycles that do not lead to an absorbing state.

---

g) How many possible policies are there? (All policies, not just optimal policies.):

Answer:

Number of Possible Policies: The number of possible policies is the number of actions raised to the power of the number of non-absorbing states because each state can map to any of the actions. That would be 5^{217} .

- **Number of Actions (|A|)**: Assuming there are 5 basic actions (Up, Down, Left, Right, Use Gate) available in each state.
- **Number of States (|S|)**: If we have 217 states as we previously mentioned in Question a) (including the absorbing state once cooking starts).

A policy is a mapping from states to actions, indicating what action to take in each state. Therefore, for each state, there are 5 choices of action. Since decisions are made independently for each state,
the total number of possible policies is calculated by raising the number of actions to the power of the number of states:
Total Possible Policies= $|A|^{|S|}$
Substituting our values:
Total Possible Policies= 5^{217}

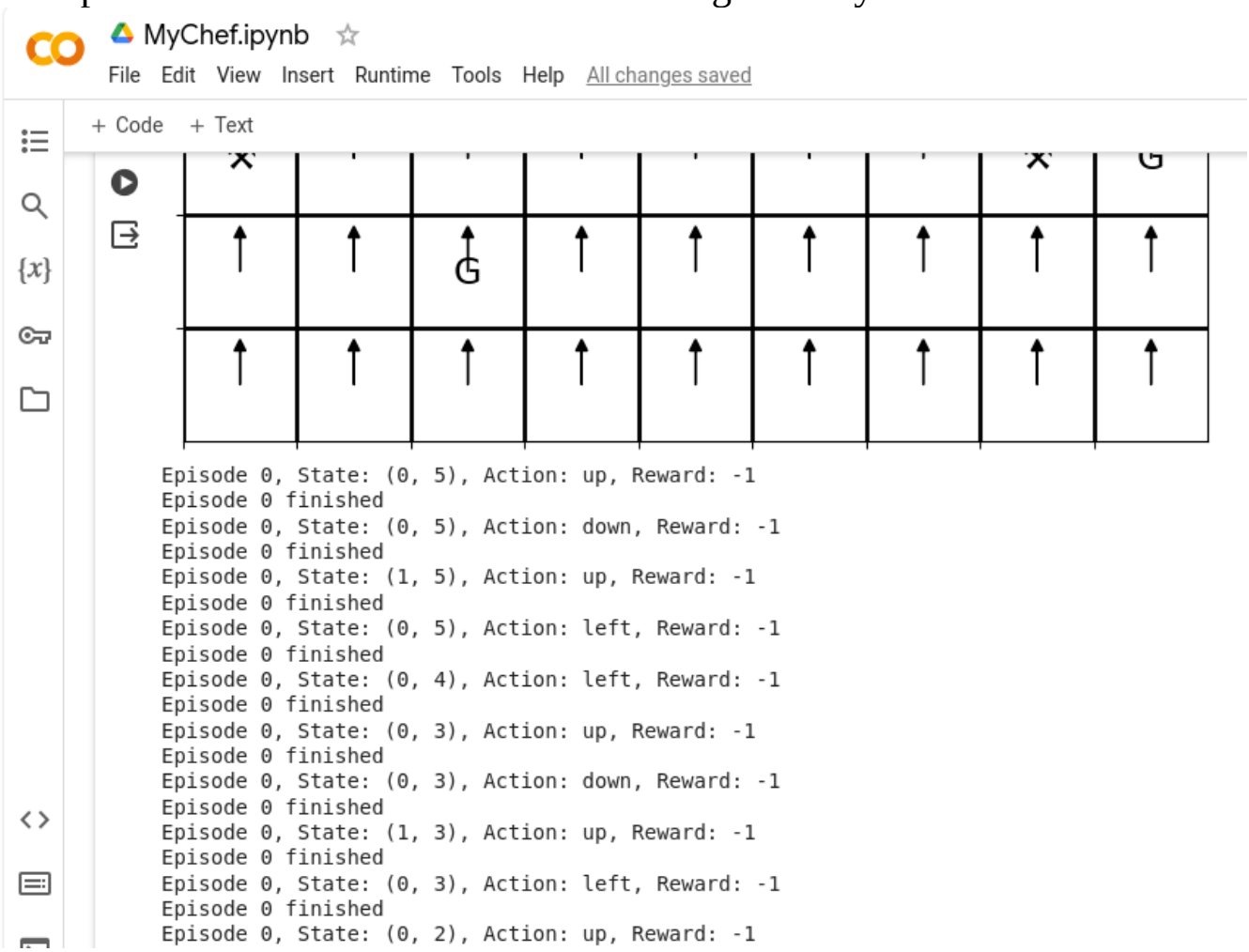
---

h) Now, considering the problem as a model-free scenario, provide a program (written in Python, possibly based on the labs) that can compute the optimal policy for this world by solely considering the pudding eggs scenario.
Draw the computed policy in the grid by putting the optimal action in each cell.
If multiple actions are possible, include the probability of each arrow.
There may be multiple optimal policies; pick one to show it.
Note that the model is not available for computation but must be encoded to be used as the "real-world" environment.

Answer :
I attached the answer separately a Jupiter notebook Google collab file with every function explained with details for the solution also I'll provide the link here
Jupiter link: https://colab.research.google.com/drive/1JE1RS_7egchDMIv-n8e49scvdpAb8Dec

Nb:

- the probabilities of each arrow are in the logs when you run the code.



- please configure how many episodes you want to run before running the notebook

```
# Create an instance of our MDP model
mdp_model = CookingChefMDP()

# Parameters for the MDP run
num_episodes = 10  # Number of episodes to run
alpha = 0.1          # Learning rate
gamma = 0.9          # Discount factor
epsilon = 0.1        # Epsilon for the epsilon-greedy policy

# Run the MDP model
mdp_model.run_episodes(num_episodes, alpha, gamma, epsilon)
```


-

i) Is the computed policy deterministic or stochastic?

Answer:

Deterministic vs. Stochastic Policy: The policy computed by model-free methods like Q-learning is typically deterministic, meaning for each state there is a specific action that the policy dictates.

---

j) Is there any advantage to having a stochastic policy? Explain.

Answer:

Advantage of Stochastic Policy: A stochastic policy may be advantageous in environments where there is uncertainty or noise in the execution of actions, as it can potentially provide a form of exploration or robustness against perturbations in state transitions. However, for the given problem, as described, a deterministic policy would be sufficient because it's a fully observable, deterministic environment.

Part B)

a) Report the transition function P for any state s and action a ∈ A.

Answer :

Transition Function P:

The transition function P would have to account for the stochastic nature of the agent's movement. For any state s and action a∈A, P would be defined as follows:

- If the action is to move in a certain direction (up, down, left, right), there is a 50% chance that the agent will move in that direction, and a 50% chance it will move to the right of that direction (up -> right, right -> down, down -> left, left -> up).
- If the action is to use the gate, the agent will use the gate with 100% probability, as this action is not affected by tiredness.
- For each possible resulting state s′, P(s′|s,a) would be 0.5 for the intended movement and 0.5 for the perpendicular movement, unless either movement is blocked by a wall or is off the grid, in which case the probability mass would be redistributed to the possible movements.

---

b) Does the optimal policy change compared to Part a? Justify your answer.

Answer :

Optimal Policy: The optimal policy might indeed change.
With the deterministic model, the optimal policy is simply the shortest path to the goal. However, with a 50% chance of deviation, the agent might need to choose actions that consider the likelihood of errant moves.
This could lead to a different policy that may take into account safer paths with less risk of moving in an undesired direction, even if it's not the shortest path.

---

c) Will the value of the optimal policy change compared to Part a? Explain how.

Answer :

Value of the Optimal Policy: The value of the optimal policy will also change. Since there is now a risk associated with each move due to the potential for error, the expected return for each state-action pair will differ from the deterministic case. The agent must balance the risk of a wrong turn with the reward of reaching the goal. This will lower the value of the optimal policy compared to the deterministic case, where the agent could always move directly to the goal without any chance of error.