[DT0171] - ARTIFICIAL INTELLIGENCE
Reinforcement Learning Module
a.a.2023/2024 - Prof. G. Stilo

Instructions

- The current assignment must be carried out individually;
- The discussion of the solution must be submitted as PDF;
- Tabular answers must be attached as spreadsheets.
- The main report must refer to each one of the tables.
- The text is intentionally underspecified to check students' preparation;
- Answers and their attachments must be submitted through email (at least 3 days before the exam date) with the following subject:
  [DT0171] - Final Exam - <surname> <name> - 2023/24
- – **Written + Project:** Must complete all the parts and letters;
  – **Only Project:** Must complete part a letters h and the d and e used by h.

The Cooking Chef Problem

Consider the case where the agent is your personal Chef.
In particular, the agent (the smiley on the map) wants to cook the eggs recipe according to your indication (scrambled or pudding).
In order to cook the desired recipe, the agent must first collect the needed tools (the egg beater on the map). Then he must reach the stove (the frying pan or the oven on the map). Finally, he can cook.

Not that there are two special interlinked cells (marked with the G) that allow the agent to go from one side of the map to the other. But to do so, the agent needs to express his will to go on the other side.

Cells in $(4, 2)$ and $(9, 3)$ are the special gate ones. They allow the agent to go from one side of the map to another. Those two special cells are interlinked, but the agent needs to express his will to go on the other side.

Since you are very hungry, it is fundamental that the agent cooks the eggs according to your taste (scrambled/pudding) as fast as he can without letting you wait for more than necessary.

In order to apply optimal control techniques such as value iteration, you need to model the aforementioned scenario as an MDP. Recall that an MDP is defined as a tuple (S, A, P, R, γ), where:

S**:** The (finite) set of all possible states.

A**:** The (finite) set of all possible actions.

P**:** The transition function $P : S \times S \times A \to [0, 1]$, which maps (s', s, a) to $P(s'|s, a)$, i.e., the probability of transitioning to state $s' \in S$ when taking action $a \in A$ in state $s \in S$. Note that $\sum_{s' \in S} P(s'|s, a) = 1$ for all $s \in S, a \in A$.

R**:** The reward function $R : S \times A \times S \to \mathbb{R}$, which maps (s, a, s') to $R(s, a, s')$, i.e., the reward obtained when taking action $a \in A$ in state $s \in S$ and arriving at state $s' \in S$.

γ**:** The discount factor which controls how important rewards are in the future.

In order to encode this problem as an MDP, you need to define each of the components of the tuple for our particular problem. Note that there may be many different possible encodings (specify them in the report).

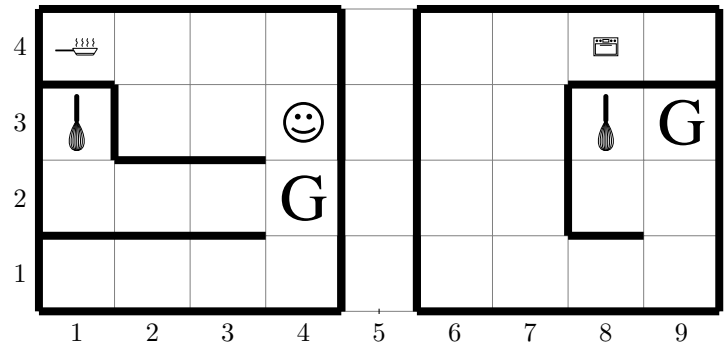

Figure 1: A particular instance of the cooking Chef problem. The goal is for the agent currently located in state $(4, 3)$ to have a policy that always leads to cooking the eggs in location $(1, 4)$ or $(8, 4)$. Cells in $(4, 2)$ and $(9, 3)$ are the special gate ones.

To answer the questions, consider the instance shown in Figure 1:

- In the figure, the agent is at $(4, 3)$ (but it can start at any of the grid cells).
- The agent needed cooking tools as the egg beater is in position $(1, 3)$ and $(8, 3)$.
- There are two different final goals, displayed as the frying pan is in position $(1, 4)$ and the oven in position $(8, 4)$.
- Cells in $(4, 2)$ and $(9, 3)$ are the special gate ones. They allow the agent to go from one side of the map to another. Those two special cells are interlinked, but the agent needs to express his will to go on the other side.
- The agent is not able to move diagonally.
- Walls are represented by thick black lines.
- The agent cannot move through walls.
- An episode will end when the agent successfully cooks the scrambled eggs (see the above description).

Part a

Modeling the MDP as an **infinite** horizon MDP: the agent, once he starts to cook successfully, never ends, and it remains in an absorbing state.

Using the above problem description, answer the following questions:

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

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

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

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

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

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

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

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.

i) Is the computed policy deterministic or stochastic?

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

Part b

Now consider that your agent might go in the wrong direction because of his tiredness. Then, each action has a 50% chance of going in the chosen direction and 50% chance of going perpendicular to the right of the direction chosen. Accordingly, with these new settings, answer the following questions:

a) Report the transition function P for any state s and action $a \in A$.

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

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