COMPSCI 687 Homework 2
(100 Points)

Instructions: Submit your answers as a single typed PDF file in Gradescope. You may use any software to create the PDF, and any reasonable formatting of your answers will suffice. You may collaborate with other students and may use generative AI to complete this assignment without reporting the collaboration or generative AI usage, but you are responsible for the accuracy of your answers.

Part One: Written (75 Points)

1. Given an MDP $M = (\mathcal{S}, \mathcal{A}, p, R, d_0, \gamma)$ and a fixed policy, π, the probability that the action at time $t = 0$ is $a \in \mathcal{A}$ is:

$$\Pr(A_0 = a) = \sum_{s \in \mathcal{S}} d_0(s)\pi(s, a). \tag{1}$$

Write similar expressions (using only $\mathcal{S}, \mathcal{A}, p, R, d_0, \gamma$, and π) for the following problems. Your final answer should not include Pr terms, e.g., $\Pr(S_3 = s)$, although such terms will show up as you derive your final answer. Assume that every event on which you are asked to condition has positive probability.

Example problem: The probability that the state at time $t = 1$ is $s \in \mathcal{S}$.

$$\begin{aligned}
\Pr(S_1 = s) &= \sum_{s_0 \in \mathcal{S}} \Pr(S_0 = s_0) \Pr(S_1 = s | S_0 = s_0) && (2)\\
&= \sum_{s_0 \in \mathcal{S}} d_0(s_0) \Pr(S_1 = s | S_0 = s_0) && (3)\\
&= \sum_{s_0 \in \mathcal{S}} d_0(s_0) \sum_{a_0 \in \mathcal{A}} \Pr(A_0 = a_0 | S_0 = s_0) \Pr(S_1 = s | S_0 = s_0, A_0 = a_0) && (4)\\
&= \sum_{s_0 \in \mathcal{S}} d_0(s_0) \sum_{a_0 \in \mathcal{A}} \pi(s_0, a_0) p(s_0, a_0, s). && (5)
\end{aligned}$$

Problems:

For this question, all probabilities and expectations are evaluated under the fixed policy π. We suppress the notation "; π" for brevity. You may similarly drop the "; π" term in assignments and exams when the policy in question is clear from context.

(a) The probability that the action at time $t = 3$ is either $a \in \mathcal{A}$ or $a' \in \mathcal{A}$, with $a \neq a'$.

$$\Pr(A_3 \in \{a, a'\}) \overset{(a)}{=} \sum_{s_3 \in \mathcal{S}} \Pr(A_3 \in \{a, a'\} | S_3 = s_3) \Pr(S_3 = s_3) \tag{6}$$

$$\overset{(b)}{=} \sum_{s_3 \in \mathcal{S}} \Pr(S_3 = s_3) \big(\Pr(A_3 = a | S_3 = s_3) + \Pr(A_3 = a' | S_3 = s_3) \big) \tag{7}$$

$$\overset{(c)}{=} \sum_{s_3 \in \mathcal{S}} \Pr(S_3 = s_3) \big(\pi(s_3, a) + \pi(s_3, a') \big) \tag{8}$$

$$\overset{(d)}{=} \sum_{s_2 \in \mathcal{S}, a_2 \in \mathcal{A}} \Pr(S_2 = s_2) \Pr(A_2 = a_2 | S_2 = s_2) \tag{9}$$

$$\times \sum_{s_3 \in \mathcal{S}} \Pr(S_3 = s_3 | S_2 = s_2, A_2 = a_2) \big(\pi(s_3, a) + \pi(s_3, a') \big) \tag{10}$$

$$\overset{(e)}{=} \sum_{s_2 \in \mathcal{S}} \Pr(S_2 = s_2) \sum_{a_2 \in \mathcal{A}} \pi(s_2, a_2) \sum_{s_3 \in \mathcal{S}} p(s_2, a_2, s_3) \big(\pi(s_3, a) + \pi(s_3, a') \big) \tag{11}$$

$$\overset{(f)}{=} \sum_{s_0 \in \mathcal{S}} \Pr(S_0 = s_0) \sum_{a_0 \in \mathcal{A}} \Pr(A_0 = a_0 | S_0 = s_0) \sum_{s_1 \in \mathcal{S}} \Pr(S_1 = s_1 | S_0 = s_0, A_0 = a_0) \tag{12}$$

$$\times \sum_{a_1 \in \mathcal{A}} \Pr(A_1 = a_1 | S_1 = s_1) \sum_{s_2 \in \mathcal{S}} \Pr(S_2 = s_2 | S_1 = s_1, A_1 = a_1) \tag{13}$$

$$\times \sum_{a_2 \in \mathcal{A}} \pi(s_2, a_2) \sum_{s_3 \in \mathcal{S}} p(s_2, a_2, s_3) \big(\pi(s_3, a) + \pi(s_3, a') \big) \tag{14}$$

$$\overset{(g)}{=} \sum_{s_0 \in \mathcal{S}} d_0(s_0) \sum_{a_0 \in \mathcal{A}} \pi(s_0, a_0) \sum_{s_1 \in \mathcal{S}} p(s_0, a_0, s_1) \sum_{a_1 \in \mathcal{A}} \pi(s_1, a_1) \sum_{s_2 \in \mathcal{S}} p(s_1, a_1, s_2) \tag{15}$$

$$\times \sum_{a_2 \in \mathcal{A}} \pi(s_2, a_2) \sum_{s_3 \in \mathcal{S}} p(s_2, a_2, s_3) \big(\pi(s_3, a) + \pi(s_3, a') \big), \tag{16}$$

where **(a)** follows from the law of total probability, **(b)** follows from the addition rule since the events $A_3 = a$ and $A_3 = a'$ are exclusive, **(c)** follows from the definition of π, **(d)** follows from the law of total probability and the product rule, **(e)** follows from the definitions of π and p, **(f)** follows from the continued application of the law of total probability and the product rule, and **(g)** follows from the definitions of d_0, π, and p.

(b) The expected reward at time $t = 6$ given that the action at time $t = 5$ is $a \in \mathcal{A}$ and the state at time $t = 4$ is $s \in \mathcal{S}$. The expected value of R_6 will be expressed using R. For each possible state-action pair (s_6, a_6), the corresponding expected reward is $R(s_6, a_6)$. To obtain this $R(s_6, a_6)$ term, we first apply the law of total expectation to consider all possible values of S_6 and A_6:

$$\mathbf{E}\left[R_6|A_5 = a, S_4 = s\right] = \sum_{s_6 \in \mathcal{S}, a_6 \in \mathcal{A}} \Pr(S_6 = s_6, A_6 = a_6|S_4 = s, A_5 = a)\mathbf{E}[R_6|S_6 = s_6, A_6 = a_6, A_5 = a, S_4 = s]. \quad (17)$$

R_6 is conditionally independent of A_5 and S_4 given S_6 and A_6, and so

$$\mathbf{E}[R_6|S_6 = s_6, A_6 = a_6, A_5 = a, S_4 = s] = \mathbf{E}[R_6|S_6 = s_6, A_6 = a_6] \quad (18)$$
$$= R(s_6, a_6). \quad (19)$$

Substituting this into (17) we obtain:

$$\mathbf{E}\left[R_6|A_5 = a, S_4 = s\right] = \sum_{s_6 \in \mathcal{S}, a_6 \in \mathcal{A}} \Pr(S_6 = s_6, A_6 = a_6|S_4 = s, A_5 = a)R(s_6, a_6). \quad (20)$$

Next we will handle the conditioning on $S_4 = s$ and $A_5 = a$. This conditioning is awkward because these events occur at different time steps. From the product rule we have that

$$\Pr(S_6 = s_6, A_6 = a_6, A_5 = a|S_4 = s) = \Pr(S_6 = s_6, A_6 = a_6|S_4 = s, A_5 = a)\Pr(A_5 = a|S_4 = s). \quad (21)$$

Dividing both sides by $\Pr(A_5 = a|S_4 = s)$ gives:

$$\frac{\Pr(S_6 = s_6, A_6 = a_6, A_5 = a|S_4 = s)}{\Pr(A_5 = a|S_4 = s)} = \Pr(S_6 = s_6, A_6 = a_6|S_4 = s, A_5 = a). \quad (22)$$

Substituting this expression for $\Pr(S_6 = s_6, A_6 = a_6|S_4 = s, A_5 = a)$ into (20) we obtain:

$$\mathbf{E}\left[R_6|A_5 = a, S_4 = s\right] = \sum_{s_6 \in \mathcal{S}, a_6 \in \mathcal{A}} \frac{\Pr(S_6 = s_6, A_6 = a_6, A_5 = a|S_4 = s)}{\Pr(A_5 = a|S_4 = s)} R(s_6, a_6). \quad (23)$$

The numerator and denominator of the fraction in this expression can each be expanded similar to the first question, via repeated application of the law of total probability and the product rule. This gives the following two equalities for the numerator and denominator:

$$\Pr(S_6 = s_6, A_6 = a_6, A_5 = a|S_4 = s) = \sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a)p(s_5, a, s_6)\pi(s_6, a_6) \quad (24)$$

$$\Pr(A_5 = a|S_4 = s) = \sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a). \quad (25)$$

Substituting these into (23) we obtain:

$$\mathbf{E}\left[R_6|A_5 = a, S_4 = s\right] = \sum_{s_6 \in \mathcal{S}, a_6 \in \mathcal{A}} \frac{\sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a)p(s_5, a, s_6)\pi(s_6, a_6)}{\sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a)} R(s_6, a_6). \quad (26)$$

While this expression is complete, we can simplify it by recognizing that s_6 and a_6 only appear in the numerator and $R(s_6, a_6)$ term, and so those summations and the $R(s_6, a_6)$ term can be moved into the numerator to give the solution:

$$\mathbf{E}\left[R_6|A_5 = a, S_4 = s\right] = \frac{\sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a) \sum_{s_6 \in \mathcal{S}} p(s_5, a, s_6) \sum_{a_6 \in \mathcal{A}} \pi(s_6, a_6)R(s_6, a_6)}{\sum_{a_4 \in \mathcal{A}} \pi(s, a_4) \sum_{s_5 \in \mathcal{S}} p(s, a_4, s_5)\pi(s_5, a)}. \quad (27)$$

(c) The probability that the action at time $t = 16$ is $a' \in \mathcal{A}$ given that the action at time $t = 14$ is $a \in \mathcal{A}$, and the state at time $t = 15$ is $s \in \mathcal{S}$.
Since A_{16} is conditionally independent of A_{14} given S_{15}, $\Pr(A_{16} = a' | S_{15} = s, A_{14} = a) = \Pr(A_{16} = a' | S_{15} = s)$. Applying the law of total probability similar to the earlier questions, we obtain

$$\Pr(A_{16} = a' | S_{15} = s, A_{14} = a) = \sum_{a_{15} \in \mathcal{A}} \pi(s, a_{15}) \sum_{s_{16} \in \mathcal{S}} p(s, a_{15}, s_{16}) \pi(s_{16}, a'). \tag{28}$$

(d) The probability that the initial state was $s \in \mathcal{S}$ given that the action at time $t = 1$ is $a' \in \mathcal{A}$.
By rearranging terms in the product rule, we obtain

$$\Pr(S_0 = s | A_1 = a') = \frac{\Pr(S_0 = s, A_1 = a')}{\Pr(A_1 = a')}. \tag{29}$$

Expanding the numerator and denominator using the law of total probability as in the previous questions, we obtain:

$$\Pr(S_0 = s | A_1 = a') = \frac{\Pr(S_0 = s, A_1 = a')}{\Pr(A_1 = a')} \tag{30}$$

$$= \frac{d_0(s) \sum_{a_0 \in \mathcal{A}} \pi(s, a_0) \sum_{s_1 \in \mathcal{S}} p(s, a_0, s_1) \pi(s_1, a')}{\sum_{s_0 \in \mathcal{S}} d_0(s_0) \sum_{a_0 \in \mathcal{A}} \pi(s_0, a_0) \sum_{s_1 \in \mathcal{S}} p(s_0, a_0, s_1) \pi(s_1, a')}. \tag{31}$$

2. In 687-Gridworld, if we changed how rewards are generated so that hitting a wall (i.e., when the agent would enter an obstacle state, and is placed back where it started) results in a reward of -10, then what is $\mathbf{E}[R_t | S_t = 17, A_t = \text{AL}, S_{t+1} = 17]$?
Let H be the event that the agent hits the obstacle to the left of state 17 (here H stands for "hit"). For brevity, we will avoid constantly conditioning on $S_t = 17$, $A_t = \text{AL}$, and $S_{t+1} = 17$ by defining the event E to be the event that $S_t = 17$, $A_t = \text{AL}$, and $S_{t+1} = 17$. Hence, we aim to compute $\mathbf{E}[R_t | E]$, which we can expand by the law of total expectation to:

$$\mathbf{E}[R_t | E] = \Pr(H | E) \mathbf{E}[R_t | H, E] + \Pr(H^c | E) \mathbf{E}[R_t | H^c, E], \tag{32}$$

where H^c denotes the complement of H, i.e., the event that H does not occur. Notice that $\mathbf{E}[R_t | H, E] = -10$ and $\mathbf{E}[R_t | H^c, E] = 0$, since the reward is always -10 when the agent hits the wall (i.e., when H and E jointly occur) and is always 0 when the agent does not hit the wall (i.e., when H^c and E jointly occur). Hence, (32) becomes:

$$\mathbf{E}[R_t | E] = -10 \Pr(H | E). \tag{33}$$

To compute $\Pr(H | E)$, we expand the definition of E and apply the product rule (with terms rearranged) to obtain:

$$\Pr(H | E) = \Pr(H | S_t = 17, A_t = \text{AL}, S_{t+1} = 17) \tag{34}$$

$$= \frac{\Pr(H, S_{t+1} = 17 | S_t = 17, A_t = \text{AL})}{\Pr(S_{t+1} = 17 | S_t = 17, A_t = \text{AL})}. \tag{35}$$

Since the event H implies that $S_{t+1} = 17$ when $S_t = 17$ and $A_t = \text{AL}$, $\Pr(H, S_{t+1} = 17 | S_t = 17, A_t = \text{AL}) = \Pr(H | S_t = 17, A_t = \text{AL})$. From the transition dynamics of 687-Gridworld, this value is 0.8 (the chance the "Attempt Left" action succeeds). Hence,

$$\Pr(H | E) = \frac{0.8}{\Pr(S_{t+1} = 17 | S_t = 17, A_t = \text{AL})}. \tag{36}$$

The event in the denominator occurs when the agent's "Attempt Left" action either succeeds or results in the "Stay" behavior. The former occurs with probability 0.8 and the latter with probability 0.1. Hence the total probability that $S_{t+1} = 17$ given that $S_t = 17$ and $A_t = \text{AL}$ is $0.8 + 0.1 = 0.9$. Hence,

$$\Pr(H | E) = \frac{0.8}{0.9}. \tag{37}$$

Substituting this into (33) we obtain:

$$\mathbf{E}[R_t | E] = -10 \frac{0.8}{0.9} \tag{38}$$

$$= \frac{-8}{0.9} \tag{39}$$

$$= -8.\overline{8}. \tag{40}$$

3. How many deterministic policies are there for an MDP with $|\mathcal{S}| < \infty$ and $|\mathcal{A}| < \infty$? (You may write your answer in terms of $|\mathcal{S}|$ and $|\mathcal{A}|$).
In each of the $|\mathcal{S}|$ states there are $|\mathcal{A}|$ possible actions, resulting in a total of $|\mathcal{A}|^{|\mathcal{S}|}$ deterministic policies.

4. Give an example of an MDP with $\mathcal{A} = \mathbb{R}$, bounded rewards (but perhaps not with finite support), and $\gamma < 1$ that does not have an optimal policy. Prove that it does not have an optimal policy. Hint: J is well-defined in this case.
Let the MDP have a single state s and $\gamma = 0$. Let

$$R_t = \begin{cases} A_t & \text{if } 0 \leq A_t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

Notice that this satisfies the requirement that the rewards are bounded since $R_t \in [0, 1)$.

This MDP does not have an optimal policy because an optimal policy would select the largest action less than 1, and the maximum real number less than 1 does not exist. To see this more clearly, we will provide a proof by contradiction starting with the assumption that there exists an optimal policy π^* for this MDP. Let A^* be a random variable with distribution $\pi^*(s, \cdot)$.[1] Define a new policy π' that produces actions

$$A' = \begin{cases} \frac{1+A^*}{2} & \text{if } 0 \leq A^* < 1 \\ 0.5 & \text{otherwise.} \end{cases} \tag{42}$$

When $0 \leq A^* < 1$, this new action A' is closer to 1 but still in $[0, 1)$, and so it results in a larger reward. Any time $A^* \notin [0, 1)$, it results in a reward of 0 whereas $A' = 0.5$ results in a reward of 0.5. Hence, the rewards produced by A' are always greater than the rewards produced by the actions A^*. Hence, the expected reward (discounted sum of rewards with $\gamma = 0$) when using π' is greater than the expected reward when using π^*, and hence π^* is not an optimal policy. This is a contradiction with the assumption that π^* is an optimal policy.

5. Give an example of an MDP with $\mathcal{A} = \mathbb{N}_1$, bounded rewards (but perhaps not with finite support), and $\gamma < 1$ that does not have an optimal policy. Prove that it does not have an optimal policy. Hint: J is well-defined in this case.
Similar to the previous example, let the MDP have one state s and $\gamma = 0$. Then let

$$R_t = \frac{-1}{A_t}. \tag{43}$$

These rewards are bounded within $[-1, 0)$.

This MDP does not have an optimal policy because an optimal policy selects the largest possible action, and there is no largest element of the natural numbers. More formally, we provide a proof by contradiction. Assume that there exists an optimal policy π^* for this MDP. Let A^* be a random variable with distribution $\pi^*(s, \cdot)$. Define a new policy π' that produces actions

$$A' = A^* + 1. \tag{44}$$

The reward produced by A^* is $\frac{-1}{A^*}$, while the reward produced by A' is $\frac{-1}{A^*+1}$. Since

$$\frac{-1}{A^* + 1} > \frac{-1}{A^*}, \tag{45}$$

A' always results in a larger reward. Hence, the expected reward (discounted sum of rewards with $\gamma = 0$) when using π' is greater than the expected reward when using π^*, and hence π^* is not an optimal policy. This is a contradiction with the assumption that π^* is an optimal policy.

6. We refer to the discounted sum of rewards, $\sum_{t=0}^{\infty} \gamma^t R_t$, as the *return*. Let an MDP exist such that it has two optimal policies and $\gamma < 1$. Can the expected value of their returns differ? If so, give an example. If not, explain why. Can the variance of their returns differ? If so, give an example. If not, explain why.
By the definition of an optimal policy described in class, the expected value of the returns of the policies cannot differ. However, the variance of their returns can differ. Answers may vary for the example. One example is an MDP with one non-terminal state that always transitions to s_∞ and which has two actions. The first action results in $R_t = 0.5$ always and the second results in R_t being Bernoulli with parameter $p = 0.5$. Every policy for this MDP is optimal with expected discounted return of 0.5, but the variance of returns can vary from zero (when always selecting the first action) to 0.25 (when always selecting the second action).

[1] Since actions are continuous, you might consider $\pi^*(s, \cdot)$ to be a probability density function, or more generally a probability measure.

7. Read the Wikipedia page on Markov chains. A *Markov chain* is *irreducible* if every state can be reached from every other state with positive probability and in finitely many steps. An MDP is *irreducible* if the Markov chain associated with every policy is irreducible.[2] A state in a *Markov chain* has *period* k if k is the greatest integer that divides every possible return time. More formally,

$$k = \gcd\{t > 0 : \Pr(S_t = s | S_0 = s) > 0\}.$$

A Markov chain is *aperiodic* if the period of every state is $k = 1$. An MDP is aperiodic if the Markov chain associated with every deterministic policy is aperiodic (or equivalently, if the Markov chain associated with every policy is aperiodic). Can an MDP be aperiodic and *not* irreducible? If so, give an example. If not, explain your reasoning.
Yes, an MDP can be aperiodic and not irreducible. Student answers may vary for the example. One example is an MDP with two states that always transition to themselves.

8. The state of a Markov chain is *positive recurrent* if the expected time until the state recurs is finite. A Markov chain is *positive recurrent* if all states are positive recurrent. An MDP is *positive recurrent* if the Markov chains associated with every policy are all positive recurrent. Give an example of an MDP with $|\mathcal{S}| > 2$ states that is positive recurrent and aperiodic. For any number of states $|\mathcal{S}|$, can you think of a simple way of defining state transitions such that the MDP is positive recurrent and aperiodic? Explain your methodology (a picture might be useful).
Student answers may vary. An example is an MDP with states that always transition to themselves, making it both positive recurrent and aperiodic.

9. Describe a real-world problem and how it can be reasonably modeled as an MDP where R_t is *not* a deterministic function of S_t, A_t, and S_{t+1}.
Student answers may vary. Example: Consider scanning space for extraterrestrial or other interesting radio signals. The state might be a vector representing the current spatial location and frequency the radio is scanning. The action space is the direction to point the radio and what frequency to listen to next. The reward would be $+1$ for finding an interesting signal and 0 otherwise. The signal may not always be received even when the radio focuses in the right direction/frequency due to noise in the signal, so it is not a deterministic function of the state, action, or next state.

10. If you know $(\mathcal{S}, \mathcal{A}, p, R, d_0, \gamma)$, can you derive d_R? Prove your answer is correct.

You cannot derive d_R given R. To prove this, we provide two MDPs that have the same $\mathcal{S}$, $\mathcal{A}$, p, R, d_0, and γ, but different d_R. For all $s \in \mathcal{S}, a \in \mathcal{A}$, and $s' \in \mathcal{S}$ let $d_R^1(s, a, s')$ be a constant distribution such that $R_t = 0.5$ always and let $d_R^2(s, a, s')$ be a Bernoulli distribution with parameter $p = 0.5$. For every $s \in \mathcal{S}$ and $a \in \mathcal{A}$, the expected reward given $S_t = s$ and $A_t = a$ is 0.5 under both reward distributions, and hence $R(s, a) = 0.5$. So, these two MDPs with different reward distributions share the same reward function. Hence, the reward function R is insufficient to differentiate between these two different reward distributions, d_R^1 and d_R^2.

11. You are given a program, `VI`, that takes as input an MDP of the form $(\mathcal{S}, \mathcal{A}, p, R, d_0, \gamma)$, and which returns an optimal policy, π^*. The program is highly efficient – significantly more efficient than any other solver. However, its inner workings are secret, and so you are unable to inspect how it works or make changes to it. You are tasked with using this program to solve a large real-world problem. However, for this problem not every action is available in every state. For all $s \in \mathcal{S}$, let $\mathcal{A}'(s)$ be the set of actions available in state s. So, you need to write a program that takes as input $(\mathcal{S}, \mathcal{A}', p, R, d_0, \gamma)$, and which returns an optimal policy π^*, and your program must use the `VI` program to do the majority of the work (since it is particularly efficient). If needed, you may assume that $\max_{s \in \mathcal{S}} |\mathcal{A}'(s)|$ is small (roughly 5) and that $\left|\bigcup_{s \in \mathcal{S}} \mathcal{A}'(s)\right|$ is also relatively small (roughly 10), and so the complexity of the large problem comes from having a large state set. Explain in English how your program would work. It does not need to be maximally efficient. Although this will not be as precise as pseudocode (since we haven't specified the format of the inputs to `VI` or your program), it should explain how the inputs to your program are transformed before being provided to `VI`, and how the output of `VI` is transformed before being returned. Assume that $\mathcal{A}'(s) \neq \varnothing$ for every $s \in \mathcal{S}$.
There are many correct answers. One solution is to make each disallowed action behave exactly like an allowed action. Let $\mathcal{A}^+$ be the set of all actions that are admissible in at least one state:

$$\mathcal{A}^+ := \bigcup_{s \in \mathcal{S}} \mathcal{A}'(s). \tag{46}$$

We will apply `VI` to an extension of the provided MDP that uses this larger action set $\mathcal{A}^+$.

For each state $s \in \mathcal{S}$, choose a default action $a_{\text{default}}(s) \in \mathcal{A}'(s)$. The MDP that we create will be defined to allow

[2] Note that once a policy is fixed, the state transitions of an MDP can be modeled as a Markov chain.

any action in $\mathcal{A}^+$ in any state. If the agent ever selects an action that should be disallowed (i.e., one in $\mathcal{A}^+ \setminus \mathcal{A}'(s)$), we will treat it as though the agent selected the allowed action $a_{\text{default}}(s)$.

To do this, construct a new MDP with state set $\mathcal{S}$ and action set $\mathcal{A}^+$, where allowed actions keep their original transition probabilities and expected rewards. Disallowed actions are assigned the transition probabilities and expected rewards of the default action. Specifically, for every $s \in \mathcal{S}$, $s' \in \mathcal{S}$, and $a \in \mathcal{A}^+$, let

$$p^+(s, a, s') \coloneqq \begin{cases} p(s, a, s') & \text{if } a \in \mathcal{A}'(s) \\ p(s, a_{\text{default}}(s), s') & \text{if } a \notin \mathcal{A}'(s) \end{cases}, \tag{47}$$

and

$$R^+(s, a) \coloneqq \begin{cases} R(s, a) & \text{if } a \in \mathcal{A}'(s) \\ R(s, a_{\text{default}}(s)) & \text{if } a \notin \mathcal{A}'(s). \end{cases} \tag{48}$$

Keep d_0 and γ unchanged, passing $(\mathcal{S}, \mathcal{A}^+, p^+, R^+, d_0, \gamma)$ to the `VI` algorithm.

Let π^+ be the optimal policy returned by `VI`. This policy may place probability on disallowed actions. However, any time the agent selects a disallowed action, we simulated transitions as though the agent selected the allowed action $a_{\text{default}}(s)$. So, we can modify the returned policy so that it actually selects $a_{\text{default}}(s)$ any time that it would have selected a disallowed action. That is, for each state $s \in \mathcal{S}$, the policy that our algorithm returns will shift any probability on disallowed actions to $a_{\text{default}}(s)$.

More formally, our algorithm returns π^* defined as follows for every $s \in \mathcal{S}$ and $a \in \mathcal{A}^+$:

$$\pi^*(s, a) \coloneqq \begin{cases} \pi^+(s, a) + \sum_{b \in \mathcal{A}^+ \setminus \mathcal{A}'(s)} \pi^+(s, b) & \text{if } a = a_{\text{default}}(s) \\ \pi^+(s, a) & \text{if } a \in \mathcal{A}'(s) \setminus \{a_{\text{default}}(s)\} \\ 0 & \text{if } a \notin \mathcal{A}'(s). \end{cases} \tag{49}$$

The first case above says that the probability of $a_{\text{default}}(s)$ is the sum of the probability initially placed on $a_{\text{default}}(s)$ and the total probability placed on disallowed actions b. The second case says that the probability of any allowed action other than $a_{\text{default}}(s)$ is unchanged, and the third case says that the probability of disallowed actions is zero.

Part Two: Programming (25 Points)

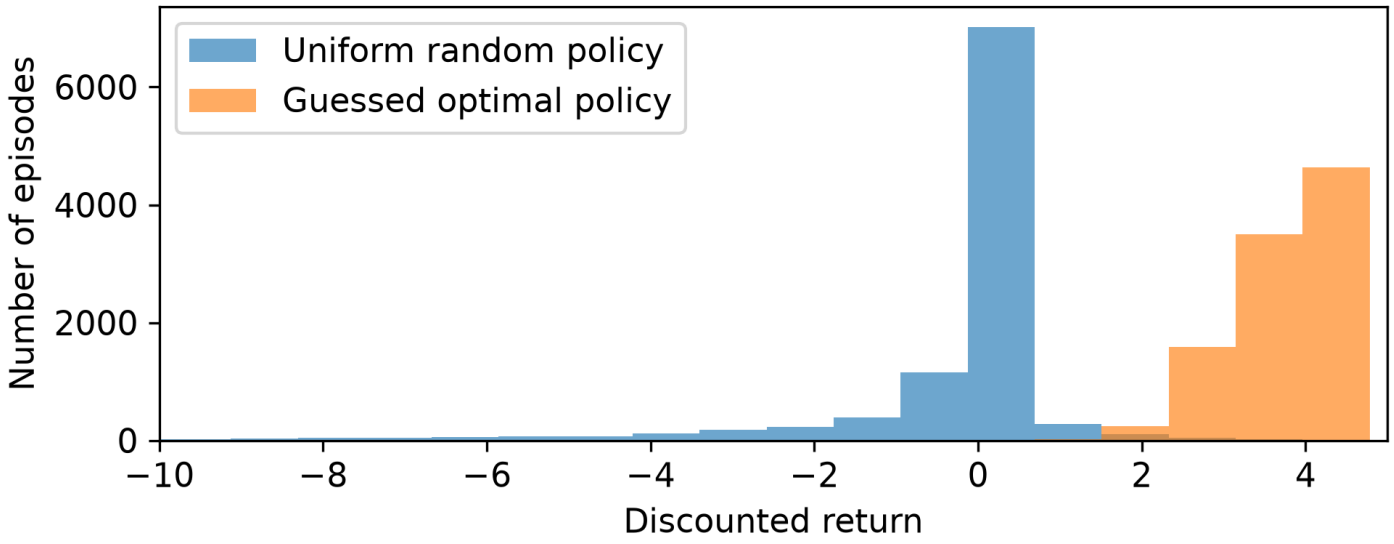
Implement the 687-Gridworld environment. I recommend creating an object with member variables to store the current state and the following member functions:

- `newEpisode`: Resets to the initial state
- `getState`: Returns the current state.
- `inTerminalState`: Returns true if the current state is the terminal absorbing state. You may alternatively have this function return true if the current state always transitions to the terminal absorbing state.
- `step`: Takes an action as input, updates the state to the next state and returns the resulting reward (not the state – the resulting state must be queried using a separate call to `getState`).

The answers to the questions below should be numerical values and plots. You do not need to submit your source code.

1. Run 10,000 episodes using the policy that selects actions uniformly randomly from $\mathcal{A}$. Report the average discounted return, the standard deviation of the discounted returns, and standard error of the discounted returns. When run for 1 million episodes, the average return is ≈ -0.596 with a standard deviation of ≈ 2.306. The number of episodes influences the standard error. When run for 10,000 episodes, I observed a standard error of roughly 0.0229.
2. Pick one policy that you think is optimal for this MDP. You do not need to be correct – just guess based on your understanding of the MDP. Report what this policy is and run 10,000 episodes using this policy. Report the average discounted return, the standard deviation of the discounted returns, and standard error of the discounted returns. Answers may vary depending on the chosen policy. For my guessed optimal policy, the average discounted return was close to 4, and the standard deviation of the discounted returns was roughly 0.7. When run for 10,000 episodes, I observed a standard error of roughly 0.007. These values may differ if students select different approximately optimal policies.

3. Create histograms of the discounted return distributions under (a) the uniform random policy and (b) your guess of the optimal policy. You may provide two plots, each with one histogram, or one plot showing both histograms.



Histograms of the discounted returns under the uniform random policy and my guessed optimal policy. The shape for the uniform random policy should be very close to this. Your results for the approximately optimal policy may differ depending on the policy you select, but it should have an average discounted return around 4.0.

4. Using Monte Carlo simulations, approximate the probability that $S_{19} = 21$ given that $S_8 = 18$ (the state above the goal) when running the uniform random policy. Note that sampling episodes as in the previous questions would be inefficient (the condition that $S_8 = 18$ is exceptionally rare), and so you should devise a more efficient sampling strategy. Explain your methodology (sampling procedure and number of samples) and report your final estimate along with a 95% confidence interval. Report the method that you used to compute the 95% confidence interval. If you're not familiar with such confidence intervals, you may use the confidence interval from Hoeffding's inequality. That is, given n i.i.d. samples $X_1, \ldots, X_n$, where each $X_i \in [a, b]$, $[L, U]$ is a $1 - \delta$ confidence interval for $\mathbf{E}[X_1]$, where:

$$L := \frac{1}{n}\sum_{i=1}^{n} X_i - (b-a)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} \tag{50}$$

$$U := \frac{1}{n}\sum_{i=1}^{n} X_i + (b-a)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}. \tag{51}$$

The estimated probability was 0.0187474, based on 187,474 successes in 10,000,000 Monte Carlo samples. The corresponding two-sided 95% Clopper–Pearson confidence interval is approximately $[0.01866343, 0.01883165]$. Confidence intervals using Hoeffding's inequality will be looser.

Using Hoeffding, $b = 1$ and $a = 0$, and so $\sqrt{\frac{\ln(2/0.05)}{2\times 10{,}000{,}000}} \approx 0.0004295$ would be added to and subtracted from the estimate, giving approximately $[0.0183, 0.0191]$.

The key insight for this problem is that sampling episodes using the uniform random policy and filtering for the episodes where $S_8 = 18$ is extremely inefficient, since this event rarely happens. Instead, we can initialize the state to 18 and simulate the 11 transitions from time $t = 8$ to time $t = 19$. This is valid because, under the fixed policy, the future states are conditionally independent of the preceding trajectory given $S_8 = 18$. That is, the events before $S_8 = 18$ don't impact the events after, and so *how* the agent reached state 18 does not impact the probability in question.