Exercises - Chapter 3

Carl Fredriksson, c@msp.se

Exercise 3.1

Devise three example tasks of your own that fit into the MDP framework, identifying for each its states, actions, and rewards. Make the three examples as different from each other as possible. The framework is abstract and flexible and can be applied in many different ways. Stretch its limits in some way in at least one of your examples.

My answer:

- Playing chess.
    - States: the positions on the chess board.
    - Actions: all legal moves in each position.
    - Rewards: 0 for all actions, except for when the agent selects an action that wins the game (checkmate or opponent resigning), which results in a reward of +1, or when the agent selects an action that is immediately losing, which results in a reward of -1.
- A drone that picks up a package from point A and delivers it to point B.
    - States: vectors that combine sensory readings such as position and velocity, and variables such as if a package is picked up.
    - Actions: vectors of voltages to motors driving propellers and package grabbing mechanism.
    - Rewards: negative rewards for actions that result in the drone crashing or flying outside of a designated zone. Positive rewards for actions that result in successful package delivery and possibly for some intermediate successes such as picking up a package (although one has to be careful not to give rewards in a way that incentivizes continuous dropping and picking of a package, maybe only the first pick up results in a reward for example). Possibly a small negative reward for each action to incentivize speedy delivery.
- Temperature control in an office.
    - States: temperature readings from thermometers in the office.
    - Actions: mechanically turning dials on heaters.
    - Rewards: the number of positive minus the number of negative comments about the temperature from employees on the company's Slack channel about office temperature, measured within some time period after the last action.

Exercise 3.2

Is the MDP framework adequate to usefully represent *all* goal-directed learning tasks? Can you think of any clear exceptions?

My answer:

The MDP framework seems general enough to usefully represent the vast majority of interesting goal-directed learning tasks. I can't think of any clear exceptions.

Exercise 3.3

Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out—say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in—say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of *where* to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

My answer:

On what level you want to draw the line between agent and environment depends on what you want the agent to learn. The lower the level, the more control of details the agent has, but the harder it is to achieve the end goal since the agent has to learn all the details. There are details that are worth learning to control as perfectly as possible and there are others that are not. For example, what would be the point of learning to control muscle twitches from scratch for driving when humans already know how to move their limbs. A driving instructor might teach a student how to move the steering wheel, but would never talk about how to send signals between the brain and limbs. If the agent already knows how to drive, but needs to learn where to drive, it might make sense to draw the line at a really high level. Even if you don't have a great autonomous driving system yet, it could make sense to utilize multiple agents. One agent could learn how to stay on the road and not hit anything, while another could learn where to go on a higher level and would send state information to the lower level agent. Letting agents focus on one level of abstraction rather than solving a complete end-to-end problem could be beneficial in multiple ways - such as specializing the training process for the different agents to be as efficient as possible.

Exercise 3.4

Give a table analogous to that in Example 3.3, but for $p(s', r|s, a)$. It should have columns for s, a, s', r, and $p(s', r|s, a)$, and a row for every 4-tuple for which $p(s0, r|s, a) > 0$.

My answer:

s	a	s'	r	$p(s', r\|s, a)$
high	search	high	r_{search}	α
high	search	low	r_{search}	$1 - \alpha$
low	search	high	-3	$1 - \beta$
low	search	low	r_{search}	β
high	wait	high	r_{wait}	1
low	wait	low	r_{wait}	1
low	recharge	high	0	1

Exercise 3.5

The equations in Section 3.1 are for the continuing case and need to be modified (very slightly) to apply to episodic tasks. Show that you know the modifications needed by giving the modified version of (3.3).

My answer:

Since s' could be the terminal state, we need to specify $s' \in \mathcal{S}^+$ rather than $s' \in \mathcal{S}$:

$$\sum_{s' \in \mathcal{S}^+} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s).$$

Exercise 3.6

Suppose you treated pole-balancing as an episodic task but also used discounting, with all rewards zero except for -1 upon failure. What then would the return be at each time? How does this return differ from that in the discounted, continuing formulation of this task?

My answer:

I was a bit uncertain about how to interpret this part of the task definition: "where K is the number of time steps before failure (as well as to the times of later failures)". I chose to interpret it as there being K number of steps until the next failure, and then K steps until the next failure after each failure.

Episodic with discounting:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T = -\gamma^{T-t-1} = -\gamma^{K-1}$$

Continuous (with discounting):

$$\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{K-1} R_{t+K} + \cdots + \gamma^{2K-1} R_{t+2K} + \dots \\
&= (-1)\gamma^{K-1} + (-1)\gamma^{2K-1} + (-1)\gamma^{3K-1} + \dots \\
&= -\gamma^{-1}(\gamma^K + [\gamma^K]^2 + [\gamma^K]^3 + \dots) \\
&= -\gamma^{-1}\left[\sum_{i=1}^{\infty} (\gamma^K)^i\right] \\
&= -\gamma^{-1}\left[-1 + \sum_{i=0}^{\infty} (\gamma^K)^i\right] \\
&= -\gamma^{-1}\left[-1 + \frac{1}{1-\gamma^K}\right] \\
&= \frac{1}{\gamma} - \frac{1}{\gamma(1-\gamma^K)} \\
&= \frac{(1-\gamma^K) - 1}{\gamma(1-\gamma^K)} \\
&= -\frac{\gamma^K}{\gamma(1-\gamma^K)} \\
&= -\frac{\gamma^{K-1}}{1-\gamma^K}
\end{aligned}$$

Exercise 3.7

Imagine that you are designing a robot to run a maze. You decide to give it a reward of +1 for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes — the successive runs through the maze — so you decide to treat it as an episodic task, where the goal is to maximize expected total reward (3.7). After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

My answer:

The agent gets the same reward from escaping, regardless of how long it takes. Thus it has no driver to learn how to escape quicker. We can make the agent learn to escape quicker by giving a negative reward to all time steps the agent is still in the maze.

Exercise 3.8

Suppose $\gamma = 0.5$ and the following sequence of rewards is received $R_1 = 1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, and $R_5 = 2$, with $T = 5$. What are $G_0, G_1, \ldots, G_5$? Hint: Work backwards.

My answer:

To work backwards we can use:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

Which gives:

$$\begin{aligned} G_5 &= G_T = 0 \\ G_4 &= R_5 + 0.5G_5 = 2 + 0 = 2 \\ G_3 &= R_4 + 0.5G_4 = 3 + 1 = 4 \\ G_2 &= R_3 + 0.5G_3 = 6 + 2 = 8 \\ G_1 &= R_2 + 0.5G_2 = 2 + 4 = 6 \\ G_0 &= R_1 + 0.5G_1 = 1 + 3 = 4 \end{aligned}$$

Exercise 3.9

Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2$ followed by an infinite sequence of 7s. What are G_1 and G_0?

My answer:

We have:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Thus:

$$\begin{aligned} G_1 &= 7 \sum_{k=0}^{\infty} \gamma^k = \frac{7}{1-\gamma} = 70 \\ G_0 &= R_1 + \gamma G_1 = 2 + 63 = 65 \end{aligned}$$

Exercise 3.10

Prove the second equality in (3.10).

My answer:

$$
\begin{aligned}
\sum_{k=0}^{\infty} \gamma^k &= 1 + \gamma + \gamma^2 + \dots \\
&= \lim_{n \to \infty} (1 + \gamma + \gamma^2 + \dots + \gamma^n) \\
&= \lim_{n \to \infty} \frac{1-\gamma}{1-\gamma} (1 + \gamma + \gamma^2 + \dots + \gamma^n) \\
&= \lim_{n \to \infty} \frac{1}{1-\gamma} ([1-\gamma] + \gamma[1-\gamma] + \gamma^2[1-\gamma] + \dots + \gamma^n[1-\gamma]) \\
&= \lim_{n \to \infty} \frac{1}{1-\gamma} (1 - \gamma + \gamma - \gamma^2 + \gamma^2 - \gamma^3 + \dots + \gamma^n - \gamma^{n+1}) \\
&= \lim_{n \to \infty} \frac{1}{1-\gamma} (1 + [-\gamma + \gamma] + [-\gamma^2 + \gamma^2] + [-\gamma^3 + \gamma^3] + \dots + [-\gamma^n + \gamma^n] - \gamma^{n+1}) \\
&= \lim_{n \to \infty} \frac{1}{1-\gamma} (1 - \gamma^{n+1}) \\
&= \frac{1}{1-\gamma}
\end{aligned}
$$

QED.

Exercise 3.11

If the current state is S_t, and actions are selected according to a stochastic policy π, then what is the expectation of R_{t+1} in terms of π and the four-argument function p (3.2)?

My answer:

$$
\begin{aligned}
\mathbb{E}_\pi[R_{t+1}|S_t] &= \sum_{a \in \mathcal{A}(S_t)} \pi(a|S_t) \mathbb{E}[R_{t+1}|S_t, A_t = a] \\
&= \sum_{a \in \mathcal{A}(S_t)} \pi(a|S_t) \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|S_t, a)
\end{aligned}
$$

Exercise 3.12

Give an equation for v_π in terms of q_π and π.

My answer:

$$
v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) q_\pi(s, a)
$$

Exercise 3.13

Give an equation for q_π in terms of v_π and the four-argument p.

My answer:

$$
\begin{aligned}
q_\pi(S_t = s, A_t = a) &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\
&= r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\pi(s') \\
&= \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) v_\pi(s') \\
&= \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) \big[r + \gamma v_\pi(s') \big]
\end{aligned}
$$

Note that $s' \in \mathcal{S}$ should be changed to $s' \in \mathcal{S}^+$ in the case of an episodic problem.

Exercise 3.14

The Bellman equation (3.14) must hold for each state for the value function v_π shown in Figure 3.2 (right) of Example 3.5. Show numerically that this equation holds for the center state, valued at +0.7, with respect to its four neighboring states, valued at +2.3, +0.4, 0.4, and +0.7. (These numbers are accurate only to one decimal place.)

My answer:

Since all actions are equally likely, the state transitions are deterministic, and no action from the center state will result in a reward, we have:

$$
\begin{aligned}
v_\pi(s) &= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)[r + \gamma v_\pi(s')] \\
&= \frac{0.9}{4}(2.3 + 0.4 - 0.4 + 0.7) \\
&= 0.675 \\
&\approx 0.7
\end{aligned}
$$

Exercise 3.15

In the gridworld example, rewards are positive for goals, negative for running into the edge of the world, and zero the rest of the time. Are the signs of these rewards important, or only the intervals between them? Prove, using (3.8), that adding a constant c to all the rewards adds a constant, v_c, to the values of all states, and thus does not affect the relative values of any states under any policies. What is v_c in terms of c and γ?

My answer:

The signs are not important in this example, only the intervals between them.

Let's denote the discounted return with an added constant G_t^c and the state-value function with an added constant v_t^c. Then we have:

$$
\begin{aligned}
G_t^c &= (R_{t+1} + c) + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + \dots \\
&= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + c \sum_{k=0}^{\infty} \gamma^k \\
&= G_t + \frac{c}{1 - \gamma}
\end{aligned}
$$

$$
\begin{aligned}
v_\pi^c &= \mathbb{E}_\pi[G_t^c | S_t = s] \\
&= \mathbb{E}_\pi[G_t + \frac{c}{1-\gamma} | S_t = s] \\
&= v_\pi(s) + v_c
\end{aligned}
$$

with

$$
v_c = \frac{c}{1-\gamma}
$$

Exercise 3.16

Now consider adding a constant c to all the rewards in an episodic task, such as maze running. Would this have any effect, or would it leave the task unchanged as in the continuing task above? Why or why not? Give an example.

My answer:

In an episodic task it would have an effect. If the agent receives positive rewards for all actions which do not lead to a terminal state, we would encourage the agent to prolong the episodes for as long as possible, e.g. run around in the maze forever (since this would result in infinite return). On the other hand, if the agent receives negative rewards for all actions which do not lead to a terminal state, we would encourage the agent to end episodes quickly, e.g. try to find its way out of the maze as quickly as possible.

Exercise 3.17

What is the Bellman equation for action values, that is, for q_π? It must give the action value $q_\pi(s, a)$ in terms of the action values, $q_\pi(s', a')$, of possible successors to the state–action pair (s, a). Hint: The backup diagram to the right corresponds to this equation. Show the sequence of equations analogous to (3.14), but for action values.

My answer:

$$
\begin{aligned}
q_\pi(s, a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\
&= \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r | s, a) \big[r + \gamma \sum_{a' \in \mathcal{A}(s')} \pi(a' | s') \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s', A_{t+1} = a'] \big] \\
&= \sum_{s', r} p(s', r | s, a) \big[r + \gamma \sum_{a'} \pi(a' | s') q_\pi(s', a') \big]
\end{aligned}
$$

Note that $s' \in \mathcal{S}$ should be changed to $s' \in \mathcal{S}^+$ in the case of an episodic problem. In the last step the sum notation was simplified.

Exercise 3.18

The value of a state depends on the values of the actions possible in that state and on how likely each action is to be taken under the current policy. We can think of this in terms of a small backup diagram rooted at the state and considering each possible action:

[See the book for the diagram]

Give the equation corresponding to this intuition and diagram for the value at the root node, $v_\pi(s)$, in terms of the value at the expected leaf node, $q_\pi(s,a)$, given $S_t = s$. This equation should include an expectation conditioned on following the policy, π. Then give a second equation in which the expected value is written out explicitly in terms of $\pi(a|s)$ such that no expected value notation appears in the equation.

My answer:

$$\begin{aligned} v_\pi(S_t = s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[q_\pi(s,a) | S_t = s] \\ &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) q_\pi(s,a) \end{aligned}$$

Exercise 3.19

The value of an action, $q_\pi(s,a)$, depends on the expected next reward and the expected sum of the remaining rewards. Again we can think of this in terms of a small backup diagram, this one rooted at an action (state–action pair) and branching to the possible next states:

[See the book for the diagram]

Give the equation corresponding to this intuition and diagram for the action value, $q_\pi(s,a)$, in terms of the expected next reward, R_{t+1}, and the expected next state value, $v_\pi(S_{t+1})$, given that $S_t = s$ and $A_t = a$. This equation should include an expectation but not one conditioned on following the policy. Then give a second equation, writing out the expected value explicitly in terms of $p(s', r|s, a)$ defined by (3.2), such that no expected value notation appears in the equation.

My answer:

$$\begin{aligned} q_\pi(S_t = s, A_t = a) &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r|s, a) \big[r + \gamma v_\pi(s')\big] \end{aligned}$$

Exercise 3.20

Draw or describe the optimal state-value function for the golf example.

My answer:

The optimal policy π_* is to use the driver for all states that are not on the green and use the putter on the green. The optimal state-values $v_*(s)$ are minus how many shots it takes to go into the hole from state s following π_* . In the hole (the terminal state) $v_*(s) = 0$, on the green $v_*(s) = -1$, one shot from the green with the driver $v_*(s) = -2$, two shots from the green with the driver $v_*(s) = -3$, and so on.

Exercise 3.21

Draw or describe the contours of the optimal action-value function for putting, $q_*(s, putter)$, for the golf example.

My answer:

The optimal action-value function for putting in state s and then following the optimal policy, $q_*(s, putter)$, is 0 if s is in the hole. Otherwise it is -1 plus looking ahead to the next state s' and adding its optimal state-value $v_*(s')$. $q_*(s, putter) = -1$ if s is on the green, $q_*(s, putter) = -2$ if the green is reachable by putting from s, $q_*(s, putter) = -3$ if the green is reachable by putting and then driving, $q_*(s, putter) = -4$ if the green is reachable by putting and then driving two times, and so on.

Exercise 3.22

Consider the continuing MDP shown to the right (see the book). The only decision to be made is that in the top state, where two actions are available, left and right. The numbers show the rewards that are received deterministically after each action. There are exactly two deterministic policies, π_{left} and π_{right}. What policy is optimal if $\gamma = 0$? If $\gamma = 0.9$? If $\gamma = 0.5$?

My answer:

$\pi_{left} \geq \pi_{right}$ if and only if $v_{\pi_{left}}(s) \geq v_{\pi_{right}}(s)$ for all states s. Since there is only one state where a decision is made, let's call it s_{top}, we need to compute for which policy π the state-value $v_\pi(s_{top})$ is greatest. We have:

$$
\begin{aligned}
v_{\pi_{left}}(s_{top}) &= \mathbb{E}_{\pi_{left}}[G_t | S_t = s_{top}] \\
&= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
&= 1 + 0 + \gamma^2 + 0 + \gamma^4 + \dots \\
&= (\gamma^2)^0 + (\gamma^2)^1 + (\gamma^2)^2 + \dots \\
&= \sum_{k=0}^{\infty} \gamma^2 \\
&= \frac{1}{1 - \gamma^2}
\end{aligned}
$$

$$
\begin{aligned}
v_{\pi_{right}}(s_{top}) &= \mathbb{E}_{\pi_{right}}[G_t | S_t = s_{top}] \\
&= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
&= 0 + 2\gamma + 0 + 2\gamma^3 + \dots \\
&= 2\gamma(1 + \gamma^2 + \gamma^4 + \dots) \\
&= 2\gamma\left[(\gamma^2)^0 + (\gamma^2)^1 + (\gamma^2)^2 + \dots\right] \\
&= 2\gamma \sum_{k=0}^{\infty} \gamma^2 \\
&= \frac{2\gamma}{1 - \gamma^2}
\end{aligned}
$$

Thus:

- If $\gamma = 0$, we have $v_{\pi_{left}}(s_{top}) = 1$ and $v_{\pi_{right}}(s_{top}) = 0$, and thus π_{left} is optimal.
- If $\gamma = 0.9$, we have $v_{\pi_{left}}(s_{top}) \approx 5.3$ and $v_{\pi_{right}}(s_{top}) \approx 9.5$, and thus π_{right} is optimal.
- If $\gamma = 0.5$, we have $v_{\pi_{left}}(s_{top}) = v_{\pi_{right}}(s_{top}) = \frac{4}{3}$, and thus both policies are optimal.

Exercise 3.23

Give the Bellman equation for q_* for the recycling robot.

My answer:

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}\big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a\big] \\
&= \sum_{s',r} p(s', r|s, a)\big[r + \gamma \max_{a'} q_*(s', a')\big] \\
&= \sum_{s'} p(s'|s, a)\big[r(s, a, s') + \gamma \max_{a'} q_*(s', a')\big]
\end{aligned}
$$

Multiple steps are included for the state-action pair $(high, search) = (h, s)$. The procedure is very similar for the other state-action pairs and steps have been omitted for brevity.

$$
\begin{aligned}
q_*(h,s) &= \max \begin{cases}
p(h|h,s)\big[r(h,s,h) + \gamma q_*(h,s)\big] + p(l|h,s)\big[r(h,s,l) + \gamma q_*(l,s)\big], \\
p(h|h,s)\big[r(h,s,h) + \gamma q_*(h,s)\big] + p(l|h,s)\big[r(h,s,l) + \gamma q_*(l,w)\big], \\
p(h|h,s)\big[r(h,s,h) + \gamma q_*(h,w)\big] + p(l|h,s)\big[r(h,s,l) + \gamma q_*(l,s)\big], \\
p(h|h,s)\big[r(h,s,h) + \gamma q_*(h,w)\big] + p(l|h,s)\big[r(h,s,l) + \gamma q_*(l,w)\big]
\end{cases} \\
&= \max \begin{cases}
\alpha\big[r_s + \gamma q_*(h,s)\big] + (1-\alpha)\big[r_s + \gamma q_*(l,s)\big], \\
\alpha\big[r_s + \gamma q_*(h,s)\big] + (1-\alpha)\big[r_s + \gamma q_*(l,w)\big], \\
\alpha\big[r_s + \gamma q_*(h,w)\big] + (1-\alpha)\big[r_s + \gamma q_*(l,s)\big], \\
\alpha\big[r_s + \gamma q_*(h,w)\big] + (1-\alpha)\big[r_s + \gamma q_*(l,w)\big]
\end{cases} \\
&= \max \begin{cases}
r_s + \gamma\big[\alpha q_*(h,s) + (1-\alpha) q_*(l,s)\big], \\
r_s + \gamma\big[\alpha q_*(h,s) + (1-\alpha) q_*(l,w)\big], \\
r_s + \gamma\big[\alpha q_*(h,w) + (1-\alpha) q_*(l,s)\big], \\
r_s + \gamma\big[\alpha q_*(h,w) + (1-\alpha) q_*(l,w)\big]
\end{cases}
\end{aligned}
$$

$$
q_*(h,w) = \max \begin{cases}
r_w + \gamma q_*(h,s), \\
r_w + \gamma q_*(h,w)
\end{cases}
$$

$$
q_*(l,s) = \max \begin{cases}
(1-\beta)\big[-3 + \gamma q_*(h,s)\big] + \beta\big[r_s + \gamma q_*(l,s)\big], \\
(1-\beta)\big[-3 + \gamma q_*(h,s)\big] + \beta\big[r_s + \gamma q_*(l,w)\big], \\
(1-\beta)\big[-3 + \gamma q_*(h,s)\big] + \beta\big[r_s + \gamma q_*(l,r)\big], \\
(1-\beta)\big[-3 + \gamma q_*(h,w)\big] + \beta\big[r_s + \gamma q_*(l,s)\big], \\
(1-\beta)\big[-3 + \gamma q_*(h,w)\big] + \beta\big[r_s + \gamma q_*(l,w)\big], \\
(1-\beta)\big[-3 + \gamma q_*(h,w)\big] + \beta\big[r_s + \gamma q_*(l,r)\big]
\end{cases}
$$

$$
q_*(l,w) = \max \begin{cases}
r_w + \gamma q_*(l,s), \\
r_w + \gamma q_*(l,w), \\
r_w + \gamma q_*(l,r)
\end{cases}
$$

$$
q_*(l,r) = \max \begin{cases}
\gamma q_*(h,s), \\
\gamma q_*(h,w)
\end{cases}
$$

Exercise 3.24

Figure 3.5 gives the optimal value of the best state of the gridworld as 24.4, to one decimal place. Use your knowledge of the optimal policy and (3.8) to express this value symbolically, and then to compute it to three decimal places.

My answer:

$$
\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
&= 10 + 0 + 0 + 0 + 0 + 10\gamma^5 + 0 + 0 + 0 + 0 + 10\gamma^{10} + \dots \\
&= 10\big[(\gamma^5)^0 + (\gamma^5)^1 + (\gamma^5)^2 + \dots\big] \\
&= \frac{10}{1-\gamma^5} \\
&= \frac{10}{1-0.9^5} \\
&\approx 24.419
\end{aligned}
$$

Exercise 3.25

Give an equation for v_* in terms of q_*.

My answer:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s, a)$$

Exercise 3.26

Give an equation for q_* in terms of v_* and the four-argument p.

My answer:

$$
\begin{aligned}
q_*(s, a) &= \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}\big[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\big] \\
&= \sum_{s', r} p(s', r | s, a)\big[r + \gamma v_*(s')\big]
\end{aligned}
$$

Exercise 3.27

Give an equation for π_* in terms of q_*.

My answer:

$$\pi_*(s) = \arg\max_{a \in \mathcal{A}(s)} q_*(s, a)$$

Exercise 3.28

Give an equation for π_* in terms of v_* and the four-argument p.

My answer:

$$\begin{aligned}\pi_*(s) &= \arg\max_{a \in \mathcal{A}(s)} q_*(s,a) \\ &= \arg\max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big]\end{aligned}$$

Exercise 3.29

Rewrite the four Bellman equations for the four value functions (v_π, v_*, q_π, and q_*) in terms of the three argument function p (3.4) and the two-argument function r (3.5).

My answer:

The original Bellman equation is the first step in rewriting each function below.

$$\begin{aligned}v_\pi(s) &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_\pi(s')\big] \\ &= \sum_a \pi(a|s) \left[\sum_r r \sum_{s'} p(s',r|s,a) + \gamma \sum_{s'} \sum_r p(s',r|s,a) v_\pi(s') \right] \\ &= \sum_a \pi(a|s) \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v_\pi(s') \right]\end{aligned}$$

$$\begin{aligned}v_*(s) &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big] \\ &= \max_a \left[\sum_r r \sum_{s'} p(s',r|s,a) + \gamma \sum_{s'} \sum_r p(s',r|s,a) v_*(s') \right] \\ &= \max_a \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v_*(s') \right]\end{aligned}$$

$$\begin{aligned}q_\pi(s,a) &= \sum_{s',r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right] \\ &= \sum_r r \sum_{s'} p(s',r|s,a) + \gamma \sum_{s'} \sum_r p(s',r|s,a) \sum_{a'} \pi(a'|s') q_\pi(s',a') \\ &= r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') q_\pi(s',a')\end{aligned}$$

$$\begin{aligned}q_*(s,a) &= \sum_{s',r} p(s',r|s,a) \left[r + \gamma \max_{a'} q_*(s',a') \right] \\ &= \sum_r r \sum_{s'} p(s',r|s,a) + \gamma \sum_{s'} \sum_r p(s',r|s,a) \max_{a'} q_*(s',a') \\ &= r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} q_*(s',a')\end{aligned}$$