Chapter 3 solutions **Hanna Gábor**

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.*

   - Games are an easy example. For example, in Snake the state is the current board, actions are turning left or right or continuing forward. Rewards are whether you picked up a chip or not.
   - Public transportation development. Suppose we measured when and how many people use which line. (E.g. if we have an electronic ticket system, then this can be easily done.) Also suppose that we can simulate how people choose transportation vehicles. The actions would be like building a bus line here or a tram line there. The state is the graph created from the current lines and the possible new lines with their costs. Your current budget is also part of the state. The reward can be the number of people using public transport (the more the better) or the overall time needed for transportation (the less the better). We can add many things to this example. For example, you might need to bar some roads while you're building a new railway line.

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

   I'm not sure I'm working with the correct definition of "goal-directed" learning tasks in this exercise.

   An exception is if we can't make good intermediate rewards and reaching a good end-state is extremely unlikely. For example, if I don't know anything about how a car works and want a program that can learn to make a car. I can give a robot tools and give a reward if it builds a working car, but it will never get into a state like that.

   Another exception is if we can't really simulate the thing and can't try different methods in real life. E.g. I would like to make people happier. Should I start researching positive psychology or teach programming to poor kids or do something else? It's impossible to try taking different routes, to measure the resulting happiness or make a useful simulation for a virtual learning algorithm.

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?*

   If it's about learning to drive a car, then the actions should be defined in terms of the accelerator, steering wheel and brake. If it's about learning to plan my days more efficiently then the actions should be defined as where I drive.

I'm pretty sure that these are the good boundaries, but can't think of good rules.

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 $p(s', r|s, a) > 0$.*

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	low	0	1

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).*

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

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?*

   The return would be $-\gamma^K$ where K is the number of time steps before the next failure. Opposed to the continuing formulation, only the reward for the next failure is included in the return. In the continuing formulation, all the later failures have an effect on the return. (Although the later the failure occurs, the smaller the effect becomes.)

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?*

   The expected total reward is always 1, the agent has no motivation to escape the maze early.

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 $G_0, G_1, ..., G_5$? Hint: Work backwards.*

$$
\begin{aligned}
G_5 &= 0 \\
G_4 &= 2 \\
G_3 &= 3 + 0.5 \cdot 2 = 4 \\
G_2 &= 6 + 0.5 \cdot 4 = 8 \\
G_1 &= 2 + 0.5 \cdot 8 = 6 \\
G_0 &= -1 + 0.5 \cdot 6 = 2
\end{aligned}
$$

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?*

   $G_1 = 7 + \gamma 7 + \gamma^2 7 + \cdots = 7 \sum_{i=0}^{\infty} \gamma^i = 7 \frac{1}{1-\gamma} = \frac{7}{0.1} = 70$

   $G_0 = 2 + 0.9 \cdot 70 = 65$

10. *Prove the second equality in (3.10).*

$$
\begin{aligned}
S_n &= \sum_{i=0}^{n} \gamma^i \\
(\gamma - 1) S_n &= \gamma^{n+1} - 1 \\
S_n &= \frac{\gamma^{n+1} - 1}{\gamma - 1} \\
\lim_{n \to \infty} S_n &= \frac{-1}{\gamma - 1} = \frac{1}{1 - \gamma}
\end{aligned}
$$

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

$$
\mathbb{E}(R_{t+1}) = \sum_{a \in \mathcal{A}} \pi(a|S_t) \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a)
$$

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

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

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

$$
q_\pi(s, a) = \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} p(s', r|s, a)(r + \gamma v_\pi(s'))
$$

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.)*

Let s denote the state with value 0.7.

$$v_\pi(s) = \sum_{s' \text{is a neigbor of } s} 0.25{\cdot}(0{+}0.9{\cdot}v_\pi(s')) = 0.25{\cdot}0.9(2.3{+}0.4{-}0.4{+}0.7) = 0.25{\cdot}0.9{\cdot}2.6 \approx 0.7$$

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?*

    Only the intervals are important. Adding a constant c to all rewards results in the following change.

$$v_\pi(s) = \mathbb{E}_\pi\Big(\sum_{k=0}^{\infty}\gamma^k(R_{t+k+1}+c) \;\Big|\; S_t = s\Big) = \mathbb{E}_\pi\Big(\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \;\Big|\; S_t = s\Big) + \sum_{k=0}^{\infty}\gamma^k c$$

    Hence, $v_c = \sum_{k=0}^{\infty}\gamma^k c$.

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.*

    In the episodic case, adding a constant to all the rewards does have an effect, because the rewards after the end of the episode are still zeros. E.g. having a constant 1 reward while you are in the maze, encourages staying in the maze, (After you are out of the maze, you won't collect rewards any more.) If the rewards are -1s, then you're encouraged to leave the maze. (You won't get penalties after you've find your way out of the maze.)

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.

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

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. [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.*

$$v_\pi(s) = \mathbb{E}(q_\pi(s,a)|S_t = s, a = A_t) = \sum_a \pi(a|s) q_\pi(s,a)$$

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: [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.*

$$q_\pi(s,a) = \mathbb{E}(R_{t+1}) + \gamma \mathbb{E}(v_\pi(S_{t+1})|s = S_t, a = A_t) = \sum_{s',r} p(s', r|s, a)(r + \gamma v_\pi(s'))$$

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

$$v_\pi(s) = \begin{cases} -\infty & \text{if the ball is in the sand} \\ -1 & \text{if the ball is in the green area} \\ q_*(s, \text{driver}) & \text{otherwise} \end{cases}$$

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

$$q_\pi(s, \text{putter}) = \begin{cases} -\infty & \text{if the ball is in the sand} \\ -1 & \text{if the ball is in the green area} \\ -2 & \text{if } v_{putt} = -2 \\ -3 & \text{if } v_{putt} < -2 \text{ and we can put the ball somewhere where} \\ & q_*(s, \text{driver}) = -2 \\ -4 & \text{otherwise} \end{cases}$$

22. *Consider the continuing MDP shown on to the right. [Diagram.] 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 $\gamma = 0.5$?*

    If $\gamma = 0$, then only the next reward matters, so the optimal policy is π_{left}.

    If $\gamma = 0.9$, then the optimal policy is π_{right}, because $0 + 0.9 \cdot 2 > 1 + 0.9 \cdot 0$.

    If $\gamma = 0.5$, then all the policies are optimal, because $0 + 0.5 \cdot 1 = 1 + 0.5 \cdot 0$.

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

$$\begin{aligned}
q_*(h,s) &= p(h|h,s)(r(h,s,h) + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + p(l|h,s)(r(h,s,l) + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
&= \alpha(r_{search} + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + (1-\alpha)(r_{search} + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
q_*(h,w) &= p(h|h,w)(r(h,w,h) + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + p(l|h,w)(r(h,w,l) + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
&= r_{wait} + \gamma \max(q_*(h,s), q_*(h,w)) \\
q_*(l,s) &= p(h|l,s)(r(l,s,h) + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + p(l|l,s)(r(l,s,l) + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
&= (1-\beta)(-3 + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + \beta(r_{search} + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
q_*(l,w) &= p(h|l,w)(r(l,w,h) + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + p(l|l,w)(r(l,w,l) + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
&= r_{wait} + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
q_*(l,r) &= p(h|l,r)(r(l,r,h) + \gamma \max(q_*(h,s), q_*(h,w)) \\
&\quad + p(l|l,r)(r(l,r,l) + \gamma \max(q_*(l,s), q_*(l,w) + q_*(l,r)) \\
&= \gamma \max(q_*(h,s), q_*(h,w))
\end{aligned}$$

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.*

    If we are in this best state and follow the optimal policy, then we get a reward of 10 in the next step, then get 0 reward for the next 4 steps, and then 10 again and so on. So the return we can get is

$$\sum_{i=0}^{\infty} \gamma^{5i} 10 = \frac{1}{1-\gamma^5} = \frac{1}{1-0.9^5} \approx 2.442$$