Exercise 3.1. Devise three example tasks of your own that fit into the MDP framework, identifying for each, it's 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.

Solution

For example, the sequential decision making process described by MDP can be used to solve routing problems such as the Travelling salesman problem. In this case, the agent is the salesman, the actions available are the routes available to take from the current state, the rewards in this case are the costs of taking each route, and the goal is to determine the optimal policy that minimizes the cost function over the duration of the trip.

Another application example is maintenance and repair problems, in which a dynamic system such as a vehicle will deteriorate over time due to its actions and the environment, and the available decisions at every time epoch is to do nothing, repair, or replace a certain component of the system. This problem can be formulated as an MDP to choose the actions that minimize cost of maintenance over the life of the vehicle.

Consider a thermostat responsible for temperature control in a residential home. The state of the device is fully characterized by temperature which can take three possible values: x = 0, 1, 2 corresponding to Low, Medium, High temperatures, respectively. Each temperature is assumed to be operationally acceptable. For simplicity, we assume that the thermostat uses an electric heater to modify the temperature (i.e. the outside temperature is low). The device can choose between the following three actions, a^0 = Cool, leaves the heater idle for the forthcoming interval. Since there is some base consumption associated with the thermostat operation we assume that $E_{cool} > 0$. The next action, a^1 = Keep, maintains the temperature at the current level and requires some energy for heater operation: $E_{keep} > E_{cool} > 0$. Finally, a^2 = Heat corresponds to intensive heating that raises the temperature and requires the largest amount of energy, E_{heat}, and $E_{heat} > E_{keep} > E_{cool} = 0$. Assuming that all levels of temperature are equally comfortable, the reward function depends only on the price and energy consumption associated with the action.

Exercise 3.2. Is the MDP framework adequate to usefully represent all goal-directed learning tasks? Can you think of any clear exceptions?

Solution

No. The MDP framework is not adequate for goal-directed learning when the task is Non-Markov.

https://www.quora.com/What-is-an-example-of-a-system-process-or-activity-that-cannot-be-modeled-by-a-Markov-Decision-Process

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4405578/

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?

Solution

You should define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine.

In drawing the line between agent and environment we follow the general rule that anything that cannot be changed arbitrarily by the agent is considered to be outside of it and thus part of the environment.

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(s’, r | s, a) > 0.

Solution

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).

Solution

For episodic tasks we should get the next state not from the set of nonterminal states, S, but from set of all states, S^+.

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

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?

Solution

In an episodic task, the time of termination, T, is a random variable that varies from episode to episode.

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

The return, G_t, can be written as above, or setting K = T - t - 1, as below

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

Again setting all rewards to zero, and the reward at time of failure (T), to -1, we obtain

$$G_t = 0 + \gamma 0 + \gamma^2 0 + ... + \gamma^{T-t-1}(-1)$$

Which again, by setting K = T - t - 1, can be written as below

$$G_t = 0 + \gamma 0 + \gamma^2 0 + ... - \gamma^K$$

So this formulation is exactly the same as that in the discounted, continuing formulation of this task.

$$G_t = -\gamma^{T-t-1} = -\gamma^{K}$$

Where K is again the number of time steps before failure.

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?

Solution

You have not effectively communicated to the robot what you want it to do. Give the robot a reward of -1 at all other times and +1 for escaping from the maze.

Scenario 1: reward of +1 for escaping from the maze and a reward of zero at all other times.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**			**T**	**G_t**
E_1	0	0	0	0	0	0	0	0	0	1			10	1
E_2	0	0	0	0	1								5	1
E_3	0	0	1										3	1
E_4	0	1											2	1

Table above simulates 4 episodes (E_1 through E_4). Episode, E_1 lasts for 10 time steps (T = 10), Episode, E_2 lasts for 5 time steps (T = 5), Episode E_3 lasts for 3 time steps (T = 3), and Episode E_4 lasts for 2 time steps (T = 2). The reward at each time step is shown in the corresponding column and row of the table above. The return per episode, G_t, computed according to 3.7 is shown in the column G_t of the table above. You can see that all episodes have the same return irrespective of the length (T) of the episode. In this scenario the robot agent really has nothing to maximize as the return is constant irrespective of how many time steps it takes to escape from the maze.

Scenario 2: reward of +1 for escaping from the maze and a reward of -1 at all other times.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**			**T**	**G_t**
E_1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1			10	-8
E_2	-1	-1	-1	-1	1								5	-3
E_3	-1	-1	1										3	-1
E_4	-1	1											2	0

Table above simulates 4 episodes (E_1 through E_4). Episode, E_1 lasts for 10 time steps (T = 10), Episode, E_2 lasts for 5 time steps (T = 5), Episode E_3 lasts for 3 time steps (T = 3), and Episode E_4 lasts for 2 time steps (T = 2). The reward at each time step is shown in the corresponding column and row of the table above. The return per episode, G_t, computed according to 3.7 is shown in the column G_t of the table above. You can see that each episode has a different return depending on the length (T) of the episode. In this scenario the robot agent can maximize the return by reducing the time steps it takes to escape from 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, …, G_5? Hint: Work backwards.

Solution

$G_5 = G_T = 0$

$G_4 = R_5 + \gamma . G_5 = 2 + (0.5 . 0) = 2$

$G_3 = R_4 + \gamma . G_4 = 3 + (0.5 . 2) = 4$

$G_2 = R_3 + \gamma . G_3 = 6 + (0.5 . 4) = 8$

$G_1 = R_2 + \gamma . G_2 = 2 + (0.5 . 8) = 6$

$G_0 = R_1 + \gamma . G_1 = -1 + (0.5 . 6) = 2$

Exercise 3.9. Suppose γ = 0.9 and the reward sequence is R_1 = 2 followed by an infinite sequence of 7s. What are G_1 and G_0?

Solution

$G_1 = 7 + \gamma . 7 + \gamma^2 . 7 + \gamma^3 . 7 + \dots$

$G_1 = 7 . (1 + \gamma + \gamma^2 + \gamma^3 + \dots)$

$G_1 = 7 . (1 / (1 - 0.9))$

$G_1 = 70$

$G_0 = R_1 + \gamma . G_1$

$G_0 = 2 + (0.9 . 70)$

$G_0 = 65$

Exercise 3.10. Prove the second equality in (3.10).

Solution

$G_t = 1 + \gamma + \gamma^2 + \gamma^4 + \ldots + \gamma^n$

Consider the partial sum, s, given by

$s = 1 + \gamma + \gamma^2 + \gamma^3 + \ldots + \gamma^n$
$\gamma s = \gamma + \gamma^2 + \gamma^3 + \ldots + \gamma^n + \ldots + \gamma^{n+1}$

Subtracting

$s - \gamma s = 1 - \gamma^{n+1}$
$s(1 - \gamma) = 1 - \gamma^{n+1}$

Then

$s = (1 - \gamma^{n+1}) / (1 - \gamma)$

As n approaches infinity, the absolute value of r must be less than one for the series to converge. The sum then becomes

$s = 1 / (1 - \gamma)$

See wikipedia: https://en.wikipedia.org/wiki/Geometric_series#Sum

Exercise 3.11. 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)?

Solution

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

Exercise 3.12. Give an equation for vπ in terms of qπ and π.

Solution

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

Exercise 3.13. Give an equation for qπ in terms of vπ and the four-argument p.

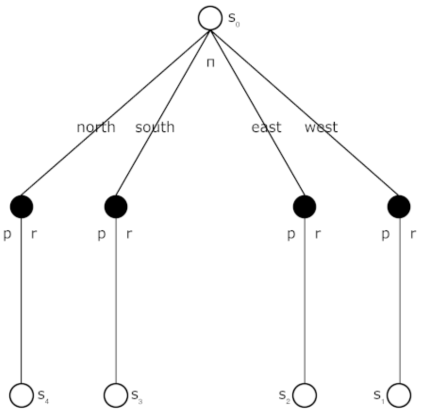
Solution

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

Exercise 3.14. The Bellman equation (3.14) must hold for each state for the value function vπ 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).

Solution

The backup diagram for this scenario is shown below where s_0 is the center state and s_1, s_2, s_3, s_4 are the set of possible succeeding states reachable from s_0 when taking any of the four actions, east, west, north or south in accordance with policy π, which is the equiprobable random policy.



The bellman equation (3.14) is repeated below

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

Substituting values in for $v\pi(s_0)$, $\pi(a|s) = 0.25$, $p(s',r \mid s,a) = 1$, $r = 0$, and $\gamma = 0.9$

$v\pi(s_0) = (0.25)((1)[0 + (0.9)(0.7)])$
$+ (0.25)((1)[0 + (0.9)(0.4)])$
$+ (0.25)((1)[0 + (0.9)(-0.4)])$
$+ (0.25)((1)[0 + (0.9)(0.7)])$

$v\pi(s_0) = (0.25)(0.9)(0.7)[0.7 + 0.4 - 0.4 + 2.3]$

$v\pi(s_0) = (0.25)(0.9)(3.0)$

$v\pi(s_0) = 0.675$

$v\pi(s_0) = 0.7$ (to 1 decimal place)

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?

Solution

The signs of these rewards are not important. Only the intervals between them are important. Let us review equation 3.8 below

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

Let's add a constant c to the reward R_{t+k+1}

$$G_t^{'} = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + c)$$

Expanding the term (R_{t+k+1} + c) in the bracket above

$$G_t^{'} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \sum_{k=0}^{\infty} \gamma^k c$$

This is now of the form

$$G_t^{'} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \frac{c}{1-\gamma}$$

Which is basically

$$G_t^{'} = G_t + \frac{c}{1-\gamma}$$

So v_c is equal to

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

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.

Solution

Recall the maze running example from Exercise 3.7 (above).

Scenario 2: reward of +1 for escaping from the maze and a reward of -1 at all other times.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**			**T**	**G_t**
E_1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1			10	-8
E_2	-1	-1	-1	-1	1								5	-3
E_3	-1	-1	1										3	-1
E_4	-1	1											2	0

Let us add a constant c = 10 to the rewards in the above table.

Scenario 3: reward of +11 for escaping from the maze and a reward of 9 at all other times.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**			**T**	**G_t**
E_1	9	9	9	9	9	9	9	9	9	11			10	92
E_2	9	9	9	9	11								5	47
E_3	9	9	11										3	29
E_4	9	11											2	20

If you look at the G_t (episode return) of each table above (Scenario 2 and Scenario 3), you will see that Scenario 2 is maximized at Episode 4 (E_4) while Scenario 3 is maximized at Episode 1 (E_1). So we can conclude that the task changes, since the task defined by the rewards of Scenario 2 and that defined by the rewards of Scenario 3 achieve different maximums. Were the task the same the relative rewards would be the same across both scenarios and the maximum would definitely be the same episode.

The reason is that the constant added to the return v_c (v_c = (T - t - 1) * c) is different for each episode of an episodic task and depends on the length of the episode, when the return, G_t, is computed according to equation 3.7.

Exercise 3.17. What is the Bellman equation for action values, that is, for qπ? It must give the action value qπ(s, a) in terms of the action values, qπ(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.

Solution

In Exercise 3.13 we derived an expression for qπ(s, a) in terms of vπ and p as follows

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

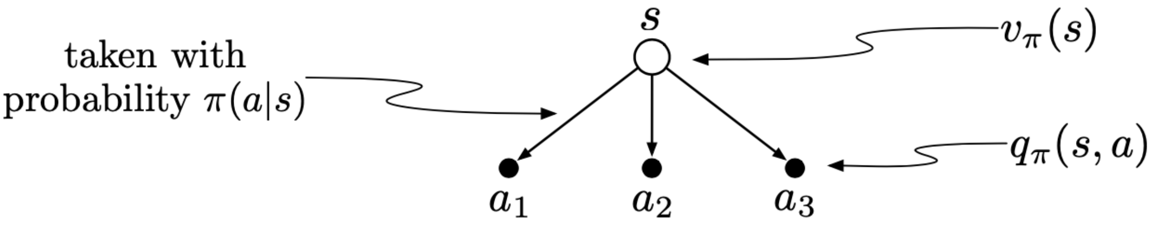
In Exercise 3.12 we derived an expression for vπ(s) in terms of qπ(s, a) and π as follows

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

Substituting vπ(s) from Exercise 3.12 in qπ(s, a) from Exercise 3.13 gives us the sequence of equations corresponding to the backup diagram as follows

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

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:



Give the equation corresponding to this intuition and diagram for the value at the root node, vπ (s), in terms of the value at the expected leaf node, qπ(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 π(a|s) such that no expected value notation appears in the equation.

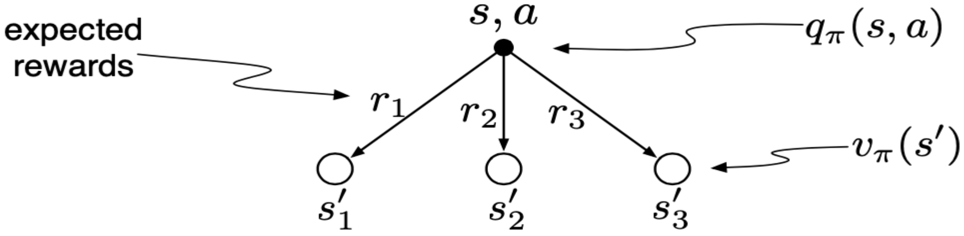
Solution

$$v_\pi(s) = E_\pi[q_\pi(s,a)|S_t = s, A_t = a]$$

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

$$v_\pi(s) = \pi(a_1|s)q_\pi(s,a_1) + \pi(a_2|s)q_\pi(s,a_2) + \pi(a_3|s)q_\pi(s,a_3)$$

Exercise 3.19. The value of an action, qπ(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:



Give the equation corresponding to this intuition and diagram for the action value, qπ(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.

Solution

$$q_\pi(s,a) = E[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s, A_t = a]$$

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

$$q_\pi(s,a) = p(s'_1, r_1|s,a)[r_1 + \gamma v_\pi(s'_1)] + p(s'_2, r_2|s,a)[r_2 + \gamma v_\pi(s'_2)] + p(s'_3, r_3|s,a)[r_3 + \gamma v_\pi(s'_3)]$$

Exercise 3.20. Draw or describe the optimal state-value function for the golf example.

Solution

Todo

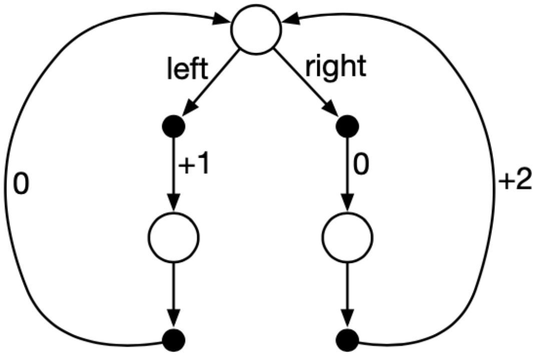
Exercise 3.21. Draw or describe the contours of the optimal action-value function for putting, q*(s, putter), for the golf example.

Solution

Todo

Exercise 3.22. Consider the continuing MDP shown on the right. 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 $\gamma = 0.5$?

Solution



From equation (3.8)

$G_t = R_{t+1} + \gamma.R_{t+2} + \gamma^2.R_{t+3} + \gamma^3.R_{t+4} + \gamma^4.R_{t+5} + \gamma^5.R_{t+6} + \dots$

For π_{left}

$G_t = 1 + \gamma.0 + \gamma^2.1 + \gamma^3.0 + \gamma^4.1 + \gamma^5.0 + \dots$
$G_t = 1 + \gamma^2.1 + \gamma^4.1 + \dots$
$G_t = 1 + \gamma^2 + \gamma^4 + \dots$

This is an infinite series with common ratio $r = \gamma^2$ and formula

$G_t = 1 / (1 - \gamma^2)$

For π_{right}

$G_t = 0 + \gamma.2 + \gamma^2.0 + \gamma^3.2 + \gamma^4.0 + \gamma^5.2 + \dots$
$G_t = \gamma.2 + \gamma^3.2 + \gamma^5.2 + \dots$
$G_t = \gamma.2.(1 + \gamma^2 + \gamma^4 + \dots)$

This is the product of $\gamma.2$ and an infinite series with common ratio $r = \gamma^2$ and formula

$G_t = 2.\gamma / (1 - \gamma^2)$

Now we can create a table showing the results of evaluating the formulas for G_t for both π_{left} and π_{right} and different values of γ

	γ = 0.0	**γ = 0.9**	**γ = 0.5**
$\pi_{\mathbf{left}}$	1	5.26315789474	1.33333333333
$\pi_{\mathbf{right}}$	0	9.47368421053	1.33333333333
π_{optimal}	$\pi_{\mathbf{left}}$	$\pi_{\mathbf{right}}$	$\pi_{\mathbf{left}} = \pi_{\mathbf{right}}$

From the table above we can see that the policy that is optimal when γ = 0.0 is π_{left}. When γ = 0.9, π_{right} is optimal. When γ = 0.5 then both policies are equal and optimal.

Exercise 3.23. Give the Bellman equation for q* for the recycling robot.

Solution

$q^*(h,s) = p(h \mid h,s)[r(h,s,h) + \gamma.\max\{q^*(h,s), q^*(h,w)\}] + p(l \mid h,s)[r(h,s,l) + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$
$q^*(h,s) = \alpha.[r_s + \gamma.\max\{q^*(h,s), q^*(h,w)\}] + (1 - \alpha).[r_s + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$

$q^*(l,s) = p(h \mid l,s)[r(l,s,h) + \gamma.\max\{q^*(h,s), q^*(h,w)\}] + p(l \mid l,s)[r(l,s,l) + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$
$q^*(l,s) = (1 - \beta).[-3 + \gamma.\max\{q^*(h,s), q^*(h,w)\}] + \beta.[r_s + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$

$q^*(h,w) = p(h \mid h,w)[r(h,w,h) + \gamma.\max\{q^*(h,s), q^*(h,w)\}]$
$q^*(h,w) = 1.[r_w + \gamma.\max\{q^*(h,s), q^*(h,w)\}]$
$q^*(h,w) = r_w + \gamma.\max\{q^*(h,s), q^*(h,w)\}$

$q^*(l,w) = p(l \mid l,w)[r(l,w,l) + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$
$q^*(l,w) = 1.[r_w + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}]$
$q^*(l,w) = r_w + \gamma.\max\{q^*(l,s), q^*(l,w), q^*(l,re)\}$

$q^*(l,re) = p(h \mid l,re)[r(l,re,h) + \gamma.\max\{q^*(h,s), q^*(h,w)\}]$
$q^*(l,re) = 1.[0 + \gamma.\max\{q^*(h,s), q^*(h,w)\}]$
$q^*(l,re) = \gamma.\max\{q^*(h,s), q^*(h,w)\}$

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.

Solution

From equation (3.8) we have

$G_t = R_{t+1} + \gamma.R_{t+2} + \gamma^2.R_{t+3} + \gamma^3.R_{t+4} + \gamma^4.R_{t+5} + \gamma^5.R_{t+6} + \gamma^6.R_{t+7} + \gamma^7.R_{t+8} + \gamma^8.R_{t+9} + \gamma^9.R_{t+10} + \gamma^{10}.R_{t+11} + \gamma^{11}.R_{t+12} + \gamma^{12}.R_{t+13} + \gamma^{13}.R_{t+14} + \gamma^{14}.R_{t+15} + \gamma^{15}.R_{t+16} + \gamma^{16}.R_{t+17} + \gamma^{17}.R_{t+18} + \gamma^{18}.R_{t+19} + \gamma^{19}.R_{t+20} + \gamma^{20}.R_{t+21} + \dots$

Substituting values while following the optimal policy from figure 3.5

$G_t = 10 + \gamma.0 + \gamma^2.0 + \gamma^3.0 + \gamma^4.0 + \gamma^5.10 + \gamma^6.0 + \gamma^7.0 + \gamma^8.0 + \gamma^9.0 + \gamma^{10}.10 + \gamma^{11}.0 + \gamma^{12}.0 + \gamma^{13}.0 + \gamma^{14}.0 + \gamma^{15}.10 + \gamma^{16}.0 + \gamma^{17}.0 + \gamma^{18}.0 + \gamma^{19}.0 + \gamma^{20}.10 + \dots$

Dropping the terms in 0

$G_t = 10 + \gamma^5.10 + \gamma^{10}.10 + \gamma^{15}.10 + \gamma^{20}.10 + \dots$

Factorizing

$G_t = 10.(1 + \gamma^5 + \gamma^{10} + \gamma^{15} + \gamma^{20} + \dots)$

This is the product of 10 and an infinite series with common ratio, $r = \gamma^5$ and formula

$G_t = 10 / (1 - \gamma^5)$

$G_t = 10 / (1 - (0.9)\text{^}5)$

$G_t = 24.419428097$

$G_t = 24.419$ (to 3 decimal places)

An alternative view of the same problem using also the optimal value function from Figure 3.5

$G_t = R_{t+1} + \gamma.G_{t+1}$
$G_t = 10 + (0.9).(16.0) = 24.4$
$G_t = R_{t+1} + \gamma.(R_{t+2} + \gamma.G_{t+2})$
$G_t = 10 + (0.9).(0 + (0.9).(17.8)) = 24.418$

This is 3 decimal places as required. Just for illustration, I will do the expansion one more time

$G_t = R_{t+1} + \gamma.(R_{t+2} + \gamma.(R_{t+3} + \gamma.G_{t+3})))$
$G_t = 10 + (0.9).(0 + (0.9).(0 + (0.9).(19.8))) = 24.4342$

Exercise 3.25. Give an equation for v* in terms of q*.

Solution

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

Exercise 3.26. Give an equation for q* in terms of v* and the four-argument p.

Solution

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

Exercise 3.27. Give an equation for π* in terms of q*.

Solution

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

Exercise 3.28. Give an equation for π* in terms of v* and the four-argument p.

Solution

$$\pi_*(s) = \arg \max_{a \in A(s)} \sum_{s' \in S} \sum_{r \in R} p(s', r | s, a)[r + \gamma v_*(s')]$$

Exercise 3.29. Rewrite the four Bellman equations for the four value functions (vπ, v*, qπ, and q*) in terms of the three argument function p (3.4) and the two-argument function r (3.5).

Solution

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma v_\pi(s')]$$

$$v_*(s) = \max_{a \in A(s)} (\sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma v_*(s')])$$

$$q_\pi(s,a) = \sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma \sum_{a' \in A(s')} \pi(a'|s') q_\pi(s',a')]$$

$$q_*(s,a) = \sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma \max_{a' \in A(s')} q_*(s',a')]$$

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

$$q_*(s,a) = \sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma v_*(s')]$$