Chapter 3

Exercise 3.1:

It is simple to come up with several examples. I will mention only one of them here.

An agent/robot learning to play Table Tennis. This is a complicated example given that a lot of mechanical work is also involved in this. However that is not our concern. The environment of this task is the entire Table Tennis setup including the opponent and the table. The action is the movement of the bat by the robot and the reward can be +1 every time the robot is successfully able to return the ball. The states can be the bat position in 3-D space, the bat angle, the ball position in 3-D space, spin of the ball etc. Theoretically, this agent should be able to train against a real opponent. A more realistic model(realistic because this is the information available to a human being learning) would be one which tracks the opponent's movement rather than simply sensing the spin and speed of the ball and kind of 'predict' the spin (magnitude as well as direction) based on that.
One interesting thing about this model is that it will learn to play defensively. To teach the robot aggressive play, some sort of high reward can be given when the opponent loses a point. This is definitely not going to be as simple as giving a reward but it is still an idea.

Exercise 3.2:

One important condition for an MDP framework to be adequate is the presence of a finite state set which accurately represents the effects of the agent's actions on the environment. In highly stochastic environments, where a lot of factors keep changing or new factors come into play randomly, it is impossible for the agent to keep track of the effect of its actions. The state set is therefore not clear in such situations. For example, in multiplayer games, other player's actions, which are unknown to the agent, can change the state which makes the agent's estimate of the effect of it's own action inaccurate.

Exercise 3.3:

The line between agent and environment should be drawn where the effect of the agent's actions on the environment (or states) are clearly visible. Here, it is appropriate to place the line where the body meets the machine because the agent cannot really control the dynamics of the vehicle (actual functioning of the motor and brakes etc). The agent does not have complete control over the vehicle which makes it inappropriate to draw the line where the tires meet the road. The very high level of 'where to drive' is also incorrect as it does not clearly tell the agent about its interaction with the environment.

Exercise 3.4:

S	A	R	S'	$P(S', R\|S, A)$
$High$	$Search$	R_{Search}	$High$	α
$High$	$Search$	R_{Search}	Low	$1-\alpha$
$High$	$Wait$	R_{Wait}	$High$	1
Low	$Search$	R_{Search}	Low	β
Low	$Search$	-3	$High$	$1-\beta$
Low	$Recharge$	0	$High$	1
Low	$Wait$	R_{Wait}	Low	1

Exercise 3.5:

For episodic tasks the state space should be modified to include the terminating states. The new state space is denoted by S^+.

Eq 3.3 for episodic tasks is then:

$$\sum_{S' \in S^+} \sum_{r \in R} P(S', R|S, A) = 1 \; for \; all \; S \in S^+ and \; A \in A(S)$$

Exercise 3.6:

If the episode terminates at $t = T$, then $R_1, R_2, ...R_{T-1} = 0 \; and \; R_T = -1$.

Therefore $G_t = -\gamma^{T-t} = -\gamma^K$ *where K is the time till failure from t*.

This return is the same as the continuing task return.

Exercise 3.7:

The problem is that the intermediate rewards are zero and hence not telling the robot/agent anything about how it is doing. This forces the episode to be very long. Even if the agent escapes the maze, it will not learn anything about how long it took to escape because it is happy with the fixed return of +1 it is getting with every successful run. An easy fix is to consider discounting or simply give a reward of -1 at every time step the agent is inside the maze.

Exercise 3.8:

$R_1 = -1,\ R_2 = 2,\ R_3 = 6,\ R_4 = 3,\ R_5 = 2$

$\gamma = 0.5$

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

$G_5 = 0$

$G_4 = 2 + 0.5 * 0 = 2$

$G_3 = 3 + 0.5 * 2 = 4$

$G_2 = 6 + 0.5 * 4 = 8$

$G_1 = 2 + 0.5 * 8 = 6$

$G_0 = -1 + 0.5 * 6 = 2$

Exercise 3.9:

$\gamma = 0.9$

$R_1 = 2$

$R_2,\ R_3 ..... = 7$

$$G_1 = R_2 + \gamma * G_2$$
$$G_1 = 7 + \gamma * (7 + \gamma * (....))$$
$$G_1 = 7 * (1 + \gamma + \gamma^2 + \gamma^3 .....)$$
$$G_1 = 7/(1 - \gamma)$$

$$G_1 = 7 * 10 = 70$$
$$G_0 = R_1 + \gamma * G_1$$
$$G_0 = 2 + 0.9 * 70 = 65$$

Exercise 3.10:

Let $S = \sum_{K=0}^{\infty} \gamma^K$

$$S = \gamma^0 + \sum_{K=1}^{\infty} \gamma^K$$
$$S = \gamma^0 + \gamma * \sum_{K=0}^{\infty} \gamma^K$$
$$S = 1 + \gamma * S$$
$$S(1 - \gamma) = 1$$

Therefore,

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

Exercise 3.11:

Current state $= S_t$

Current policy $= \pi$

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

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

Exercise 3.12:

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

Exercise 3.13:

$$q_\pi(s, a) = \sum_{r \in R} \sum_{s^1 \in S} p(s^1, r|s, a) * [r + \gamma * v_\pi(s^1)]$$

$$q_\pi(s, a) = \sum_{s^1, r} p(s^1, r|s, a) * [r + \gamma * v_\pi(s^1)]$$

Exercise 3.14:

$$v_\pi(s) = \sum_{a} \pi(a|s) \sum_{s^1, r} p(s^1, r|s, a) * [r + \gamma * v_\pi(s^1)]$$

$$v_\pi(Center) = 0.25 * (0 + 0.9 * 0.7) + 0.25 * (0 + 0.9 * 0.4) + 0.25 * (0 + 0.9 * 2.3) + 0.25 * (0 - 0.9 * 0.4)$$

$$v_\pi(Center) = 0.25 * 0.9 * (2.3 + 0.7)$$
$$v_\pi(Center) = 0.25 * 0.9 * 3 = 0.675 \approx 0.7$$

Exercise 3.15:

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

Adding c to each reward:

$$G_t = R_{t+1} + c + \gamma * (R_{t+2} + c) + \gamma^2 * (R_{t+3} + c) + ...$$
$$G_t = c + \gamma * c + \gamma^2 * c.... + R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + ...$$
$$G_t = c * \sum_{k=0}^{\infty} \gamma^k + \sum_{k=0}^{\infty} \gamma^k * R_{t+k+1}$$
$$G_t = c/(1-\gamma) + \sum_{k=0}^{\infty} \gamma^k * R_{t+k+1}$$

Therefore $v_c = c/(1-\gamma)$, a constant, is added to $v_\pi(s) \, for \, all \, s \in S$

Exercise 3.16:

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

Adding c to each reward:

$$G_t = R_{t+1} + c + \gamma * (R_{t+2} + c) + \gamma^2 * (R_{t+3} + c) + ... + \gamma^{T-t-1} * (R_T + c)$$
$$G_t = c * \sum_{k=0}^{T-t-1} \gamma^k + \sum_{k=0}^{T-t-1} \gamma^k * R_{t+k+1}$$
$$G_t = c * (1-\gamma^{T-t})/(1-\gamma) + \sum_{k=0}^{T-t-1} \gamma^k * R_{t+k+1}$$

We see that the value added to G_t is dependent on t and hence not constant. This will thus affect the relative difference between the values of two states.

Exercise 3.17:

$$q_\pi(s,a) = E_\pi[G_t|S_t = s,\ A_t = a]$$
$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma * G_{t+1}|S_t = s,\ A_t = a]$$
$$q_\pi(s,a) = E_\pi[R_{t+1}|S_t = s,\ A_t = a] + \gamma * E_\pi[G_{t+1}|S_t = s,\ A_t = a]$$
$$q_\pi(s,a) = E_\pi[R_{t+1}|S_t = s,\ A_t = a] + \gamma \sum_{s^1,\, r} \sum_{a^1} \pi(a^1|s^1) * p(s^1, r|s, a) * E_\pi[G_{t+1}|S_{t+1} = s^1,\ A_{t+1} = a^1]$$
$$q_\pi(s,a) = \sum_{s^1,\, r} p(s^1, r|s, a) * [r + \gamma * \sum_{a^1} \pi(a^1|s^1) * q_\pi(s^1, a^1)]$$

Exercise 3.18:

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

Exercise 3.19:

$$q_\pi(s,a) = E_{\pi,s^1}[R_{t+1} + \gamma * v_\pi(S_{t+1})]$$
$$q_\pi(s,a) = E_\pi[R_{t+1}] + \gamma * E_{s^1}[v_\pi(S_{t+1})]$$
$$q_\pi(s,a) = \sum_{s^1,\, r} p(s^1, r|s, a) * [r + \gamma * v_\pi(s^1)]$$

Exercise 3.20:

The optimal state value function $v_*(s)$ for Golf example will approximately be the maximum of $v_{putt}(s)$ and $q_*(s,\ driver)$. Everything in the grass region will have $v_*(s) = v_{putt}(s)$ while everything outside will have $v_*(s) = q_*(s,\ driver)$.

Exercise 3.21:

Let us say we are currently in region s, using the values of v_{putt} we can find the region we will reach after putting once. Let this region be s^1. $q_*(s,\ putt)$ can then be computed as the maximum of $q_*(s^1,\ driver)$ and $q_*(s^1,\ putt)$. If we work our way outwards from the hole, we can find the values of $q_*(s,\ putt)$.

Exercise 3.22:

Let the node at the top be $root$.
Node at bottom left be $dummy_l$ and node at bottom right be $dummy_r$.
Actions from these node are a_l and a_r.

$$q_*(root,\ left) = 1 + \gamma * q_*(dummy_l,\ a_l)$$
$$\Rightarrow q_*(dummy_l,\ a_l) = (q_*(root,\ left) - 1)/\gamma$$
$$q_*(root,\ right) = 0 + \gamma * q_*(dummy_r,\ a_r)$$
$$\Rightarrow q_*(dummy_r,\ a_r) = q_*(root,\ right)/\gamma$$
$$q_*(dummy_l,\ a_l) = 0 + \gamma * max[q_*(root,\ left),\ q_*(root,\ right)]$$
$$\Rightarrow max[q_*(root,\ left),\ q_*(root,\ right)] = q_*(dummy_l,\ a_l)/\gamma = (q_*(root,\ left) - 1)/\gamma^2$$
$$q_*(dummy_r,\ a_r) = 2 + \gamma * max[q_*(root,\ left),\ q_*(root,\ right)]$$
$$\Rightarrow max[q_*(root,\ left),\ q_*(root,\ right)] = (q_*(dummy_r,\ a_r) - 2)/\gamma = (q_*(root,\ right) - 2 * \gamma)/\gamma^2$$

Therefore we get:

$(q_*(root,\ left) - 1)/\gamma^2 = (q_*(root,\ right) - 2 * \gamma)/\gamma^2$

$q_*(root,\ left) = q_*(root,\ right) + 1 - 2 * \gamma)$

Which means that $q_*(root,\ left) > q_*(root,\ right)$ if $1 - 2 * \gamma > 0$

Thus for $\gamma = 0$ π_{left} will be optimal.

For $\gamma = 0.9$ π_{right} will be optimal.

for $\gamma = 0.5$, $q_*(root,\ left) = q_*(root,\ right)$. So both policies are equally optimal.

Exercise 3.23:

$q_*(high,\ search) = \alpha * (r_{search} + \gamma max_a[q_*(high,\ a)]) + (1 - \alpha) * (r_{search} + \gamma max_a[q_*(low,\ a)])$

$q_*(high,\ search) = \alpha * (r_{search} + \gamma * max[q_*(high,\ search)\ ,\ q_*(high,\ wait)])$

$+ (1 - \alpha) * (r_{search} + \gamma * max[q_*(low,\ search)\ ,\ q_*(low,\ wait), q_*(low,\ recharge)])$

$q_*(high,\ wait) = (r_{wait} + \gamma * max_a[q_*(high,\ a)])$

$q_*(high,\ wait) = (r_{wait} + \gamma * max[q_*(high,\ search)\ ,\ q_*(high,\ wait)])$

$q_*(low,\ search) = \beta * (r_{search} + \gamma max_a[q_*(low,\ a)]) + (1 - \beta) * (-3 + \gamma max_a[q_*(high,\ a)])$

$q_*(low,\ search) = \beta * (r_{search} + \gamma * max\ [q_*(low,\ search)\ , q_*(low,\ wait), q_*(low,\ recharge)])$

$+ (1 - \beta) * (-3 + \gamma * max[q_*(high,\ search)\ ,\ q_*(high,\ wait)]\ )$

$q_*(low,\ wait) = r_{wait} + \gamma * max_a[q_*(low,\ a)]$

$q_*(low,\ wait) = r_{wait} + \gamma * max\ [q_*(low,\ search)\ , q_*(low,\ wait), q_*(low,\ recharge)]$

$q_*(low,\ recharge) = 0 + \gamma * max_a[q_*(high,\ a)]$

$q_*(low,\ recharge) = \gamma * max\ [q_*(high,\ search)\ ,\ q_*(high,\ wait)]$

Exercise 3.24:

According to the optimal policy shown in Figure 3.5, the best move from position A is to make any move and go to A^1 and get a reward of +10 and then get back to A by going up 4 times.

Therefore,
$v_*(A) = 10 + \gamma * v_*(A^1)$
And
$v_*(A^1) = \gamma^4 * v_*(A)$ (this is easy to see)

Thus,
$v_*(A) = 10 + \gamma^5 * v_*(A)$
$v_*(A) = 10/(1 - \gamma^5)$
Putting $\gamma = 0.9$
$v_*(A) = 10/(1 - 0.9^5) = 24.4194281 \approx 24.419$

Exercise 3.25:

$v_*(s) = max_a[q_*(s, a)]$

Exercise 3.26:

$$q_*(s, a) = \sum_{s^1, r} p(s^1, r|s, a) * [r + \gamma * max_{a^1}(q_*(s^1, a^1))]$$

$$q_*(s,a) = \sum_{s^1,\, r} p(s^1, r|s,a) * [r + \gamma * v_*(s^1)]$$

Exercise 3.27:

$a_* = arg\ max_a[q_*(s,a)]$
$Policies\ for\ which\ \pi_*(a_*|s) = 1\ are\ optimal\ policies$
It is better to write a_* as a set of actions all of which give maximum value of $q_*(s,a)$. In that case any policy which assigns non-zero probability only to actions belonging to this set is π_*.

Exercise 3.28:

$a_* = arg\ max_a[\sum_{s^1,\, r} p(s^1, r|s,a) * [r + \gamma * v_*(s^1)]]$ Rest is the same as the previous exercise.

Exercise 3.29:

$$v_\pi(s) = E_\pi[G_t|S_t = s]$$
$$v_\pi(s) = E_\pi[R_{t+1} + \gamma * G_{t+1}|S_t = s]$$
$$v_\pi(s) = E_\pi[R_{t+1}|S_t = s] + \gamma * E_\pi[G_{t+1}|S_t = s]$$
$$v_\pi(s) = E_\pi[R_{t+1}|S_t = s] + \gamma * \sum_{s^1,\, a} p(s^1|s,a) * \pi_*(a|s) * E_\pi[G_{t+1}|S_{t+1} = s^1]$$
$$v_\pi(s) = \sum_a r(s,a) * \pi(a|s) + \gamma * \sum_{s^1,\, a} p(s^1|s,a) * \pi(a|s) * v_\pi(s^1)$$

$$v_*(s) = E_{\pi_*}[G_t|S_t = s]$$
$$v_*(s) = E_{\pi_*}[R_{t+1} + \gamma * G_{t+1}|S_t = s]$$
$$v_*(s) = E_{\pi_*}[R_{t+1}|S_t = s] + \gamma * E_{\pi_*}[G_{t+1}|S_t = s]$$

$$v_*(s) = E_{\pi_*}[R_{t+1}|S_t = s] + \gamma * \sum_{s^1,\, a} p(s^1|s,a) * \pi_*(a|s) * E_{\pi_*}[G_{t+1}|S_{t+1} = s^1]$$

$$v_*(s) = \sum_a r(s,a) * \pi_*(a|s) + \gamma * \sum_{s^1,\, a} p(s^1|s,a) * \pi_*(a|s) * v_*(s^1)$$

$$q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]$$
$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma * G_{t+1}|S_t = s, A_t = a]$$
$$q_\pi(s,a) = E_\pi[R_{t+1}|S_t = s, A_t = a] + \gamma * E_\pi[G_{t+1}|S_t = s, A_t = a]$$

$$q_\pi(s,a) = E_\pi[R_{t+1}|S_t = s, A_t = a] + \gamma * \sum_{s^1,\, a^1} p(s^1|s,a) * \pi(a^1|s^1) * E_\pi[G_{t+1}|S_{t+1} = s^1, A_{t+1} = a^1]$$

$$q_\pi(s,a) = r(s,a) + \gamma * \sum_{s^1,\, a^1} p(s^1|s,a) * \pi(a^1|s^1) * q_\pi(s^1,a^1)$$

$$q_*(s,a) = E_{\pi_*}[G_t|S_t = s, A_t = a]$$
$$q_*(s,) = E_{\pi_*}[R_{t+1} + \gamma * G_{t+1}|S_t = s, A_t = a]$$
$$q_*(s,a) = E_{\pi_*}[R_{t+1}|S_t = s, A_t = a] + \gamma * E_{\pi_*}[G_{t+1}|S_t = s, A_t = a]$$

$$q_*(s,a) = E_{\pi_*}[R_{t+1}|S_t = s, A_t = a] + \gamma * \sum_{s^1,\, a^1} p(s^1|s,a) * \pi_*(a^1|s^1) * E_{\pi_*}[G_{t+1}|S_{t+1} = s^1, A_{t+1} = a^1]$$

$$q_*(s,a) = r(s,a) + \gamma * \sum_{s^1,\, a^1} p(s^1|s,a) * \pi_*(a^1|s^1) * q_*(s^1,a^1)$$