Siddharth Sadhwani
2018313

Question 1

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

Question 3

Ex 3.15

We know that :

$$V_\pi = E_\pi [G_t | S_t = s]$$

$$V_\pi = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

Now if we add a constant c to each reward R_t then

$$R_{new} = R_{old} + c$$

and $R'_{t+k+1} = R_{t+k+1} + c$

Thus :

$$V'_\pi = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + c \sum_{k=0}^{\infty} \gamma^k | S_t = s \right]$$

$$V_\pi' = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + c\left(\frac{1}{1-\gamma}\right) \mid S_t = s\right]$$

(using infinite GP sum)

Thus,

$$V_\pi' = V_\pi + c\left(\frac{1}{1-\gamma}\right)$$

$$V_\pi' = V_\pi + v_c$$

Thus adding a constant 'c' to each reward does not affect the relative values of any state under any policy. A term of v_c gets added to the value of each state whose value is:

$$v_c = c\left(\frac{1}{1-\gamma}\right)$$

Thus, the signs of these rewards are more important than the intervals between them.

Now

~~Similarly~~, ~~add~~ for episodic task:

$$V_\pi' = E_\pi\left[\sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} + \sum_{k=0}^{T-t-1} \gamma^k c \mid S_t = s\right]$$

$$V_\pi' = E_\pi\left[\sum_{k=0}^{T-t-1} \gamma^k R_{t+k} + c\left(\frac{1-\gamma^{(T-t-1)}}{1-\gamma}\right) \mid S_t = s\right]$$

$$V_{t\pi}' = V_{t\pi} + c\left[E_{t\pi}\left[\frac{1-\gamma^{(T-t-1)}}{1-\gamma} \mid S_t = s\right]\right]$$

$$V_{t\pi}' = V_{t\pi} + c\left(\frac{1-\gamma^{(T-t-1)}}{1-\gamma}\right)$$

$$\boxed{V_{t\pi}' = V_{t\pi} + v_c(t)}$$

in $v_c(t)$

Here c remains constant for all states but $v_c(t)$ is dependent on $(T-t-1)$ as well and as we reach towards the terminal state, the value of γ is the smallest as compared to initial states and thus a relative change in the state values can be seen in episodic tasks.

Question 5

For expressing v_* interms of q_*, we need to ensure that q_* is calculated for the optimal 'a' in the very first step as afterwards, it itself follows π_*. Thus we need to select the action that gives the max reward for the first step. Therefore,

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

Question 8

Given:

1) $p(s', r | s, a) = P[S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a]$

2) $p(s' | s, a) = \sum_{r \in R} p(s', r | s, a)$

$= P[S_{t+1} = s' \mid S_t = s, A_t = a]$

3) $p(r | s, a) = \sum_{s'} p(s', r | s, a)$

$= P[R_{t+1} = r \mid S_t = s, A_t = a]$

4) $\pi(s) = P[A_t = a \mid S_t = s]$

5) Other notations:

a) $A_t = a$ b) $A_{t+1} = a'$ c) $R_{t+1} = r$

d) $R_{t+2} = r'$ e) $S_t = s$ f) $S_{t+1} = s'$

g) $S_{t+2} = s''$

To calculate:

$$p[R_{t+2} = r' \mid S_t = s, A_t = a]$$

Answer:

1) we already know $p(r' \mid s', a')$ from (eq 3, for $t+2$) that gives probability of $R_{t+2} = r'$ for all s' and a' pairs.

2) Now we need to calculate probability of transition to $S_{t+1} = s'$ given $S_t = s$ and $A_t = a$ (using eq 2).

$$p(s' \mid s, a)$$

3) We also need probability of all action $A_{t+1} = a'$ for all s' (using eq 4) for s'.

$$\pi(a' \mid s')$$

4) Step 2 and 3 should be done for all s and $a' \in A(s')$ thus we need to sum over all s' and a'.

Hence

$$P[R_{t+2}=r' \mid S_t=s, A_t=a] = \left[\sum_{s',a'} p(s' \mid s,a)\pi(a' \mid s')\right] \cdot p(r' \mid s',a')$$

$$\Rightarrow \left[\sum_{s',a'} P[S_{t+1}=s' \mid S_t=s, A_t=a] P[A_{t+1}=a' \mid S_{t+1}=s'] \times P[R_{t+2}=r' \mid S_{t+1}=s', A_{t+1}=a']\right]$$

$$\Rightarrow \boxed{\left[\sum_{s',a',r} p[s',r \mid s,a) \cdot \pi(a' \mid s')\right] \sum_{s''} p(s'', r' \mid s', a')}$$

Question 9

$$E[X] = \sum_{x \in S_x} x P_x(x)$$

Thus using our result from previous question.

$$E[R_{t+2} \mid S_t=s, A_t=a] = \sum_{r \in R} r \times \left[\left[\sum_{s',a',r} p(s',r \mid s,a)\pi(a' \mid s')\right] \times \sum_{s''} p(s'', r' \mid s', a')\right]$$

Question 10

$$V_\pi(s) = E_\pi [G_t | S_t = s]$$

$$\Rightarrow E_\pi [R_{t+1} + \gamma G_{t+1} | S_t = s]$$

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

~~$\sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) [r + \gamma E_\pi$~~

$$\Rightarrow \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) [r + \gamma E_\pi [G_{t+1} | S_{t+1} = s']]$$

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

Bellman equation ↑

Question 11

$R_1 = 2$

$R_2 = -1$

$R_3 = 10$

$R_4 = -3$

$\gamma = 0.5$

$$G_t = R_{t+1} + \gamma R_{t+2} - - - - \gamma^{(T-t-1)} R_T \quad \text{— (1)}$$

$$\boxed{G_t = R_{t+1} + \gamma G_{t+1}} \quad \text{— (2)}$$

So,

$G_3 = R_4 + \gamma G_4$ ($G_4 \to 0$)

$\boxed{G_3 = -3}$

Now

$G_2 = R_3 + \gamma G_3$

$G_2 = 10 + (0.5)(-3)$

$\boxed{G_2 = 8.5}$

Now

$G_1 = R_2 + \gamma G_2$

$G_1 = -1 + (0.5)(8.5)$

$\boxed{G_1 = 3.25}$

Now

$G_0 = R_1 + \gamma G_1$

$G_0 = 2 + (0.5)(3.25)$

$\boxed{G_0 = 3.625}$

- Now if constant reward 'c' is received at every time step, for $\gamma < 1$.

Then infinite horizon discounted return is:

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

$\boxed{R_{t+k+1} = c}$ — given

$$G_t = \sum_{k=0}^{\infty} \gamma^k c$$

$$= c\left[\sum_{k=0}^{\infty} \gamma^k\right]$$

Now this forms an infinite G.P, with $r = \gamma$ and $r < 1$ thus the return is.

$$\boxed{G_t = c\left(\frac{1}{1-\gamma}\right) = \left(\frac{c}{1-\gamma}\right)}$$

Hence proved.

Question 12

Given:

$v_*(s)$, for all $s \in S$

For each state, according to bellman equation, we would need to find the best / greedies one-step reward and the action that gives us this greedy one-step reward would be part of the optimum policy. Thus:

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \; E[R_{t+1} + \gamma v_*(s_{t+1}) \mid S_t = s, A_t = a]$$

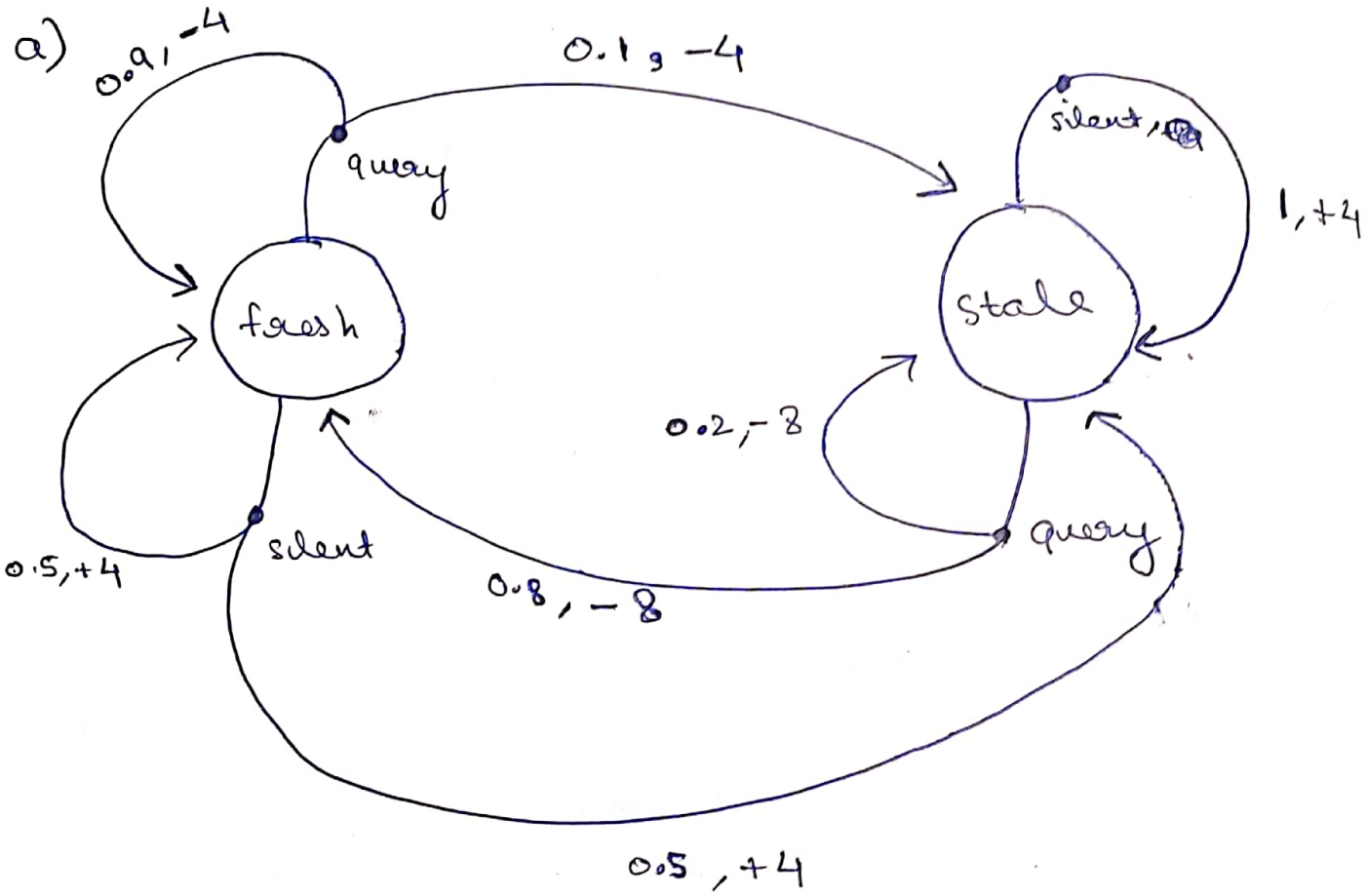
for all states

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_r \sum_{s'} (r + \gamma v_*(s')) \, p(s', r \mid s, a)$$

$\forall s \in S$

Question 13

a)



s	a	s'	$p(s'\|s,a)$	$r(s,a,s')$	$p(s',r\|s,a)$
fresh	query	fresh	0.9	-4	0.9
fresh	query	stale	0.1	-4	0.1
fresh	silent	fresh	0.5	+4	0.5
fresh	silent	stale	0.5	+4	0.5
stale	query	fresh	0.8	-8	0.8
stale	query	stale	0.2	-8	0.2
stale	silent	stale	1	+4	1

b) $\gamma = 0.5$

$V_0(\text{stale}) = 0$

$V_0(\text{fresh}) = 0$

1st Iteration

$$V_0(\text{fresh}) = \max\left[(-4+0.5(0))(0.9) + (-4+0.5(0))0.1),\ (+4+0.5(0))(0.5) + (+4+0.5(0))0.5\right]$$

$$= -3.6 - 0.4 + 2 + 2$$

$$= \max(-4, +4)$$

$$\Rightarrow +4$$

$$V_0(\text{stale}) = \max\left[(-3+0.5(0))0.8 + (-3+0.5(0))0.2,\ (+4+0.5(0))\times 1\right]$$

$$\Rightarrow \max\left[(-0.64 - 1.6),\ +4\right]$$

$$\Rightarrow +4$$

2nd Iteration

$$V_{0,1}(\text{fresh}) = \max\Big[(-4+0.5(4))(0.9)+(-4+0.5(4))(0.1),\ (4+0.5(4))(0.5)+(+4+0.5(4))(0.5)\Big]$$

$$= \max\Big[(-1.8-0.2),\ 3+3\Big]$$

$$\Rightarrow \max[-2, 6]$$

$$\Rightarrow 6$$

$$V_{0,1}(\text{stale}) = \max\Big[(-3+0.5(4))0.3+(-3+0.5(4))0.7+\big[4+0.5(4)\big]1\Big]$$

$$= \max(-6, 6)$$

$$= 6.$$

3rd Iteration

$$V_2(\text{fresh}) = \max\Big[\big(6 + 0.5(6)\big)(0.9) + \big(-14 + 0.5(6)\big)0.1\Big]$$

(+10 reward included; -10 inc)

$$\big(14 + 0.5(6)\big)0.5 + \big(-6 + 0.5\times 6\big)0.5\Big]$$

(9 → +10 inc; -10 inc)

$$\Rightarrow \max\Big[(8.1 - 1.1)\ 9\ (8.5 - 1.5)\Big]$$

$$\Rightarrow \max(7, 9, 7)$$

$$\boxed{V_2(\text{fresh}) \Rightarrow +7}$$

$$V_2(\text{stale}) = \max\Big[\big(2 + 0.5(6)\big)0.8 + \big(-18 + 0.5(6)\big)0.2$$

(+10 inc; -10 inc)

$$(-6 + 0.5(6))\Big]$$

(-10 inc; 9)

$$\Rightarrow \max[4 - 3\ 9\ -3]$$

$$\boxed{V_2(\text{stale}) \Rightarrow +4}$$

To Calculate the optimal costs and the optimal policy, we use the approach of value iteration (sort of).

We calculate V_i (stale) and V_i (fresh) for 3 iterations and on the last iteration we include the ± 10 reward as mentioned in the question. Since we are calculating $V(s)$ at all the 3 iterations thus we get an idea of all $8 (2^3)$ possible paths to follow to reach the state at $t = 3$.

Consistently, we see that silent action gives us the maximum expected return for both states 'stale' and 'fresh' thus we can say that optimal policy =

$$\pi_*(s) = \text{silent} \quad \forall \; s \in S.$$

Also, since we used the property of lazy evaluation, thus the optimal cost found at the end are approximate and can be improved if more iterations are done.

c) $\gamma = 0.5$

$V_0(\text{stale}) = 0$

$V_0(\text{fresh}) = 0$

1st Iteration

$$V_0(\text{fresh}) = \max\left[(-4+0.5(0))(0.9) + (-4+0.5(0))0.1),\ (+4+0.5(0))(0.5) + (+4+0.5(0))0.5\right]$$

$$= -3.6 - 0.4 + 2 + 2$$

$$= \max(-4, +4)$$

$$\Rightarrow +4$$

$$V_0(\text{stale}) = \max\left[(-3+0.5(0))0.8 + (-3+0.5(0))0.2,\ (+4+0.5(0))\times 1\right]$$

$$\Rightarrow \max\left[-6.4 - 1.6,\ +4\right]$$

$$\Rightarrow +4$$

2nd Iteration

$$V_{0,1}(\text{fresh}) = \max\Big[(-4 + 0.5(4))(0.9) + (-4 + 0.5(4))(0.1),$$
$$(4 + 0.5(4))(0.5) + (+4 + 0.5(4))(0.5)\Big]$$

$$= \max[(-1.8 - 0.2), 3+3]$$

$$\Rightarrow \max[-2, 6]$$

$$\Rightarrow 6$$

$$V_{0,1}(\text{stale}) = \max\Big[(-3 + 0.5(4))0.3 + (-3 + 0.5(4))0.2$$
$$+ [4 + 0.5(4)]1\Big]$$

$$= \max(-6, 6)$$

$$= 6.$$

3rd Iteration

$$V_2(\text{fresh}) = \max\Big[\big(-4 + 0.5(6)\big)\,0.9 + \big(-4 + 0.5(6)\big)\,(0.1),\ \big(4 + 0.5(6)\big)(0.5) + \big(+4 + 0.5(6)\big)\,0.5\Big]$$

$$= \max\big(-0.9 - 0.1,\ 3.5 + 3.5\big)$$

$$= 7$$

$$V_2(\text{stale}) = \max\Big[\big(-8 + 0.5(6)\big)\,0.8 + \big(-8 + 0.5(6)\big)\,0.2,\ \big(4 + 0.5(6)\big)\Big]$$

$$= \max(-4 - 1,\ 7)$$

$$= 7$$

4th iteration

$$V_3(\text{fresh}) = \max\Big[(-4 + 0.5(7))0.9 + (-4+0.5(7))\,0.1,\ (4+0.5(7))0.5 + (4+0.5(7))\,0.5,\ 7.5\Big]$$

$$= \max[-0.45,\ -0.05,\ 7.5]$$

$$= \boxed{7.5}$$

$$V_3^{\text{stal}}(\text{fresh}) = \max\Big[(-8+0.5(7))0.8 + (-8+0.5(7))\,0.2,\ (4+0.5(7))\,1\Big]$$

$$= \max[-4.5,\ 7.5]$$

$$= \boxed{7.5}$$

Therefore;

$V_0(s) = 0$
$V_1(s) = 4$
$V_2(s) = 7$
$V_3(s) = 7.5$

$V_0(f) = 0$
$V_1(f) = 4$
$V_2(f) = 7$
$V_3(f) = 7.5$

Policy Iteration

$V_0(\text{fresh}) = 0$

$V_0(\text{stale}) = 0$

$\pi(\text{fresh}) = [0.5, 0.5]$

$\pi(\text{stale}) = [0.5, 0.5)$

↑ away.

1st Iteration

⇒ Evaluation

$$V_1(\text{fresh}) = 0.5\Big[(-4 + 0.5(0))0.9 + (-4 + 0.5(0))0.1 + (+4 + 0.5(0))0.5 + (+4 + 0.5(0))0.5\Big]$$

$$\Rightarrow 0$$

$$V_1(\text{stale}) = 0.5\Big[(-8 + 0.5(0))0.8 + (-8 + 0.5(0))0.2 + (4 + 0.5(0))1\Big]$$

$$\Rightarrow 0.5(-8 + 4)$$

$$\Rightarrow -2$$

⇒ Improvement

old $\pi(\text{fresh}) = [0.5, 0.5]$

$$\pi(\text{fresh}) = \text{argmax}\Big[(-4 + 0.5(0))0.9 + (-4 + 0.5(-2))0.1),\ (+4 + 0.5(0))0.5) + (4 + 0.5(-2))0.5\Big)\Big]$$

$$= \text{argmax}[-3.6 - 0.5,\ 2 + 1.5]$$

= 1 (action = silent)

new π (fresh) = [0, 1] ← prob. of silent

Now

old π (stale) = [0.5, 0.5]

π(stale] = argmax [(−3 + 0.5(0)) 0.3 + (−3 + 0.5(0)) 0.7
(4 + 0.5(−2))]

= argmax [−6.4 − 1.89 + 3]

⇒ 1 (action = silent)

new π (stale) = [0, 1] ← prob. of silent

Iteration 2

Evaluation

V_2 (fresh) = 0 + 1 [(+4 + 0.5(0)) 0.5
+(4 + 0.5(−2)) 0.5]

= 2 + 1.5

= 3.5

V_2 (stale) = 0 + 1 [(+4 + 0.5(−2)) 1]

⇒ +3

Improvement

old π (fresh) = [0, 1]

$$\pi(\text{fresh}) = \text{argmax}\left[(-4+0.5(3.5))^{0.9} + (-4+0.5(3))^{0.1}\right]$$

$$(+4+0.5(3.5))0.5 + [4+0.5(3))0.5]$$

$$= \text{argmax}\left[-2.025 \quad -0.25 \quad 2.875 \quad +2.75\right].$$

= 1 [action = silent].

new π (fresh) = [0, 1]

old π (stale) = [0, 1]

$$\pi(\text{stale}) = \text{argmax}\left[(-8+0.5(3.5))0.8 + (-8+0.5(3))0.2 \quad 4+0.5(3)\right]$$

$$= \text{argmax}\left[-5 \quad -1.3 \quad 5.5\right]$$

= 1 (action = silent)

new π (stale) = [0, 1]

∴ Since our old $\pi(s)$ and new $\pi(s)$ are equal hence we can say that we have converged to the optimal policy and there is no need to continue the same iteration for two more steps.

Optimal policy $\pi(s) = \text{silent} \quad \forall s \in S$

Note: We only do evaluation using the previous time step values in accordance with generalised policy iteration as it says that even though we choose $M = 1$ (the lazy evaluation) for $V_{\pi_k}(s)$ $= T_{\pi_k}^{M}(V_{\pi_{k-1}}(s))$, eventually this also converges to V_*.

Question 14

To show: 1) Policy improvement either improves policy or leaves it unchanged.

2) If it leaves policy unchanged, then the policy is the optimal policy.

Proof: For 1) we can prove that π_{k+1} is better than π_k if $V_{\pi_{k+1}} \geq V_{\pi_k}$

From policy improvement step, we get:

$$T_{\pi_{k+1}} V_{\pi_k} = T V_{\pi_k}$$

Now, In TV_{π_k} we are picking the greedy action that gives the max reward with reward-to-go as V_{π_k}. This would clearly be greater than or equal to picking any other action or the greedy action respectively. Thus:

$$T_{\pi_{k+1}} V_{\pi_k} = T V_{\pi_k} \geq T_{\pi_k} V_{\pi_k} = V_{\pi_k}$$

Since V_{π_k} is the fixed point of $T_{\pi_k} V_{\pi_k}$

Thus

$$T_{\pi_{k+1}} V_{\pi_k} \geq V_{\pi_k} \quad \text{---(1)}$$

Now we know that

$$T_{\pi_{k+1}} \left(T_{\pi_{k+1}} V_{\pi_k}(s) \right) \geq T_{\pi_{k+1}} V_{\pi_k} \quad \text{---(2)}$$

as now we use $T_{\pi_{k+1}} V_{\pi_k}$ as the reward to go function and the return can be either greater or equal to our reward to go.

So $$T_{\pi_{k+1}}\left(T_{\pi_{k+1}} V_{\pi_k}(s)\right) \geq V_{\pi_k}(s)$$

...from (1) & (2)

This can be done N times thus.

$$T^{N}_{\pi_{k+1}}\left(V_{\pi_k}(s)\right) \geq V_{\pi_k}(s) \quad \text{— (3)}$$

Now since $V_{\pi_{k+1}}(s)$ is the fixed point for $T_{\pi_{k+1}}$ thus we know that L.H.S of (3) will converge to $V_{\pi_{k+1}}(s)$

$$\lim_{N\to\infty} T^{N}_{\pi_{k+1}}\left(V_{\pi_k}(s)\right) = V_{\pi_{k+1}}(s) \quad \text{— (4)}$$

hence, using eqn (3) & (4)

$$V_{\pi_{k+1}}(s) \geq V_{\pi_k}(s)$$

Hence proved.

For proving 2).

Suppose $V_{\pi_{k+1}}(s)$

From policy improvement, if we get.

$$T_{\pi_{k+1}} V_{\pi_k}(s) = V_{\pi_k}(s) \quad \text{— (5)}$$

It is guaranteed that $T_{\pi_{k+1}}$ has a unique fixed point $V_{\pi_{k+1}}$

thus,

$$V_{\pi_k}(s) = V_{\pi_{k+1}}(s) \quad \forall s$$

Also $T_{\pi_{k+1}} V_{\pi_k}(s) = T V_{\pi_k}(s)$ — (6)

(policy improvement step)

Thus from (5) & (6),

$$V_{\pi_k}(s) = T V_{\pi_k}(s)$$

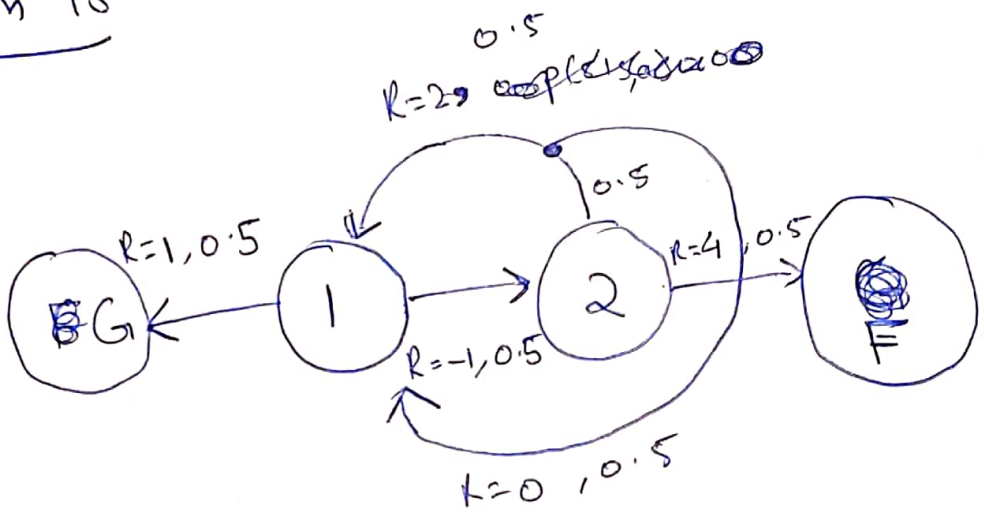
This means that,

$$V_{\pi_k}(s) = \max E[R_{t+1} + \gamma V_{\pi_k}(S_{t+1}) \mid S_t = s, A_t = a]$$

This implies that V_{π_k} is the solution of the Bellman optimality equations.

Thus $\boxed{V_* = V_{\pi_k} = V_{\pi_{k+1}}}$

Question 15



s	a	s'	$p(s' \mid s,a)$	$r(s,a,s')$	$p(s', r \mid s,a)$
1	des	G	0.5	1	1
1	asc	2	0.5	−1	1
2	des	1	0.5	2	0.5
2	des	1	0.5	0	0.5
2	asc	F	0.5	+4	1

Note: We only do Evaluation using the previous time step values in accordance with generalized policy iteration as it says that even though we choose $M=1$ (the lazy evaluation) for $v_{\pi_k}(s) = T^{M}_{\pi_k}(v_{\pi_{k-1}}(s))$ Eventually this also converges to v_*.

Policy Iteration

$V_0(1) = 0$

$V_0(2) = 0$

$V_\pi(G) = 0, \quad V_\pi(F) = 0$

asc des

$\pi_0(1) = [0.5, 0.5]$

$\pi_0(2) = [0.5, 0.5]$

1st Iteration

Evaluation

$$V_1(1) = 0.5\left[(1 + 1(0)) + (-1 + 1(0))\right]$$

$$\Rightarrow 0.5[1 - 1]$$

$$= 0$$

$$V_1(2) = 0.5\left[(2 + 1(0))0.5) + (0 + 1(0))0.5 + (4 + 1(0))1\right]$$

$$\Rightarrow 0.5[1 + 4]$$

$$= 2.5$$

Improvement :

$\pi_0(1) = [0.5, 0.5]$

$\pi_1(1) = \text{argmax}[(1 + 0), -1 + 2.5]$

$= \text{argmax}[1, 1.5]$

$\Rightarrow 1 \quad (\text{action} = \text{asc})$

$$\pi_1(1) = [1, 0]$$

$\pi_0(2) = [0.5, 0.5]$

$\pi_1(2) = \text{argmax}\left[(2+0)0.5 + (0+0)0.5 \;,\; (4+0)\right]$

$= \text{argmax}\,[1, 4]$

$= 1 \quad (\text{action} = \text{asc})$

$\boxed{\pi_1(2) = [1, 0]}$

2nd iteration

Evaluation

$V_2(1) = 0 + 1[-1 + 2.5]$

$= +1.5$

$V_2(2) = 0 + 1[4 + 0]$

$= 4$

Improvement

$\pi_1(1) = [1, 0]$

$\pi_2(1) = \text{argmax}\left[(1+0) \;,\; (-1+4)\right]$

$= \text{argmax}\,[1, 3]$

$= 1 \quad (\text{action} = \text{asc})$

$\boxed{\pi_2(1) = [1, 0]}$

$\pi_1(2) = [1, 0]$

$\pi_2(2) = \text{argmax}\left[(2+1.5)0.5 + (0+1.5)0.5 \;,\; +4+0\right]$

$= \text{argmax}\left[1.75 + 0.75 \;,\; +4\right]$

$= 1 \text{ (action} = \text{asc)}$

$\pi_2(2) = [1, 0]$

Now since $\pi_1 = \pi_2$, thus we have converged to the optimal policy. Thus policy iteration gets terminated.

Optimal policy = $\pi_*(1) =$ ascend

$\pi_*(2) =$ ascend.

*Also as n increases, number of iterations increases