# EE675A (2022) EndSem Solutions

**Lecturer**: Dr. S.S. Peruru **Scribe**:Kumar Rajnish, Ayush Yadav

February 4, 2024

## 1 Objective questions

(2 x 10 = 20 Marks)

1. If the environment is possibly non-Markovian, which algorithm do you suggest?
(i) MC
(ii) TD
Ans. (i) MC

2. Suppose we are using an off-policy method to predict the value function of a given policy target policy . Is it necessary for the behaviour policy b to have non-zero probability of selecting all actions?
(i) no
(ii) yes
Ans. (i) no

3. Which of the following are true about Double Q-learning?
(i) Helps to reduce maximization bias
(ii) Improves convergence speed
(iii) Converges to a better policy in the limit
(iv) It is an On-policy algorithm
Ans. (i), (ii)

4. In the TD ($\lambda$) algorithm, if $\lambda = 1$ and $\gamma = 1$, then which among the following are true?
(i) the method behaves like a Monte Carlo method for an undiscounted task
(ii) the eligibility traces do not decay
(iii) the value of all states are updated by the TD error in each episode
(iv) this method is not suitable for continuing tasks
Ans. (i),(ii),(iv)

5. Given the following sequence of states observed from the beginning of an episode: $s_2$, $s_1$, $s_3$, $s_2$, $s_1$, $s_2$, $s_1$, $s_6$,. What is the eligibility value, $e_7(s_1)$, of

state s1 at time step 7 given trace decay parameter $\lambda$ , discount rate $\gamma$ , and initial value,$e_0(s_1) = 0$, when accumulating traces are used?
(i) $\gamma^7\lambda^7$
(ii) $((\gamma\lambda)^7\ )+\ ((\gamma\lambda)^6\ )+((\gamma\lambda)^3\ )+((\gamma\lambda)\ )$
(iii) $(\gamma\lambda\ )(1+\gamma^2\lambda^2+\gamma^5\lambda^5\ )$
(iv)$\gamma^7\lambda^7\ +\gamma^3\lambda^3+\gamma\lambda$
Ans. (iii)

6. For a particular MDP, suppose we use function approximation and using the gradient descent approach converge to the value function that is the global optimum. Is this value function the same, in general, as the true value function of the MDP?
(i) no
(ii) yes
Ans. (i) no

7. Suppose that individual features, $\phi_i(s, a)$, used in the representation of the action value function are non-linear functions of s and a. Is it possible to use the LSTDQ method in such scenarios
(i) no
(ii) yes
Ans. (ii) yes

8. Using similar parametrisations to represent policies, would you expect, in general, MC policy gradient methods to converge faster or slower than actor-critic methods assuming that the approximation to $Q^\pi$ used in the actor-critic method satisfies the compatibility criteria?
(i) slower
(ii) faster
Ans. (i) slower

9. Value function based methods are oriented towards finding deterministic policies whereas policy search methods are geared towards finding stochastic policies. True or false?
(i) false
(ii) true
Ans. (ii) true

10. Suppose we are using a policy gradient method to solve a reinforcement learning problem. Assuming that the policy returned by the method is not optimal, which among the following are plausible reasons for such an outcome?
(i) the search procedure converged to a locally optimal policy
(ii) the search procedure was terminated before it could reach the optimal policy
(iii) the sample trajectories arising in the problem were very long
(iv) the optimal policy could not be represented by the parameterisation used to represent the policy

Ans. (i),(ii),(iv)

## 2 Basic algorithms and their relations

(5 x 5 = 25 Marks)

1. Suppose we want to predict the action-value function of a policy $\pi$ using Expected SARSA algorithm, write down the update equation for it. Explain its relation with the Bellman expectation update for $q_\pi$ with the help of back up diagrams.
**Ans:** $Q_t(S_t, A_t) = Q_t(S_t, A_t) + \alpha[R_{t+1} + \gamma \sum \pi(a', S_{t+1}) Q(S_{t+1}, a') - Q(S_t, A_t)]$

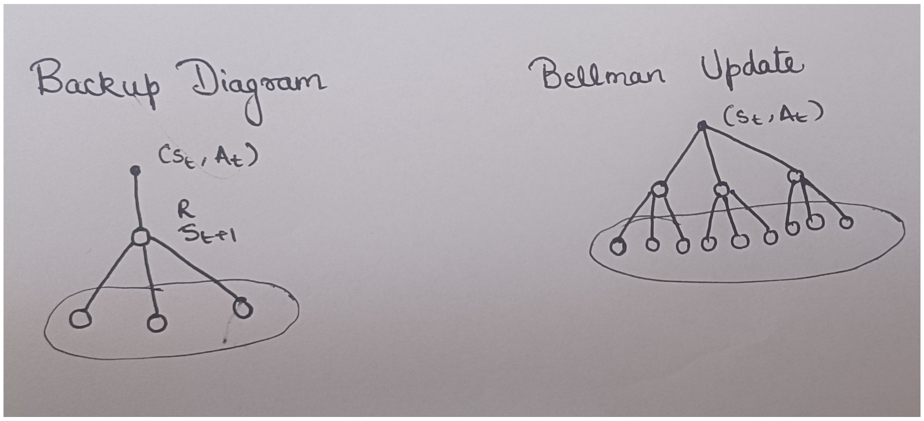

Figure 1:

As we can see from the diagrams , Expected SARSA samples the next state and takes expectation over the possibilities of next state whereas Bellman update takes expectation over the current states as well as the next steps.

2. While predicting the action-value function of a given policy $\pi$ , under what conditions on $\pi$ , the Expected SARSA will be same as using SARSA algorithm.

Ans: Two conditions should be enough:
1) Similar initialization. 2) $\pi$ is a deterministic policy.

3. We know that Q-learning is an off-policy method. Then, why do we not use an importance sampling based correction to account for the difference in behaviour and target policy?

Ans: The reason we use importance sampling based correction is because

we want to make sure the update is scaled according to the target policy rather than behavioural policy.
In Q-learning during the update step we choose the target as $R_{t+1}+\gamma \underset{a}{max}\, Q($S$_{t+1}$, a). Notice how the target is already acting according to the target policy[deterministic greedy] and not according to the behavioural policy. That is why there is no need for a correction on top of this we can also say that Q-learing has the correction built into it.

4. Suppose you are using Expected SARSA to learn the optimal actionvalue function of an MDP. Under what setting, Expected SARSA will be same as using Q-learning.
Ans: For SARSA to be same as Q-learning $\pi$ should be a deterministic and greedy policy for all off-policy behaviour.

5. Give an example of an RL algorithm for each of these twelve types: model-based, model-free, policy-based, value-based, actor-critic, onpolicy, off-policy, online, offline, batch, function-approximation, bootstrapping?
Ans: Model Based : Dyna-Q
Model Free: TD
Policy based: REINFORCE
Value based: TD(0)
Actor-Critic: Advantage Actor-Critic
ON-policy: SARSA
off-policy: Q-learning
online: TD(0)
Offline: Monte Carlo Control
Batch: LSTDQ, DQN
function-approximation: Deep Q Networks
boot-strapping: TD(0)

# 3 Q-learning and Function approximation

(5+10+5 = 20 Marks)

Consider an episodic, deterministic chain MDP with n = 7 states assembled in a line. The possible actions are a 1, 1, and the transition function is deterministic such that s' = s + a. Note that as an exception, taking a = -1 from s = 1 keeps us in s = 1, and taking a = 1 from s = 7 keeps us in s = 7. We have a special goal state, g = 4, such that taking any action from g ends the episode with a reward of r = 0. From all other states, any action incurs a reward of r = -1. We let $\gamma$ = 1. The chain MDP is pictured in Fig.2(below), with the goal state s4 shaded in.

By inspection, we see that V*(s) = $|s - 1|$.

1. (Tabular setting) We would like to perform tabular Q-learning on this chain MDP. Suppose we observe the following 4 step trajectory (in the form (state, action, reward)):
(3, 1, 1),(2, 1, 1),(3, 1, 1),(4, 1, 0) eqn (1)
Suppose we initialize all Q values to 0. Use the tabular Q-learning update to give updated values for
Q(3, 1), Q(2, 1), Q(3, 1) eqn (2)
assuming we process the trajectory in the order given from left to right. Use the learning rate $\alpha$ =1/2.

**Ans:** $\alpha$= 1/2 ; $\gamma$ = 1
$Q(s, a) = 0 \forall s \epsilon S_{und} \forall a \epsilon A$
The update step is $Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \underset{a}{max}[Q(S_{t+1}, A_t)] - Q(S_t, A_t)]$
Now, playing values for each experience
Q(3,-1) = 0 +1/2[-1 +1.0 -0] =-1/2
Q(2,1) = 0 +1/2[-1 +1.0 -0] =-1/2
Q(3,1) = 0 +1/2[-1 +1.0 -0] =-1/2

2. (Function approximation) Now, we are interested in performing linear function approximation in conjunction with Q-learning. In particular, we have a weight vector w = [w$_0$, $w_1$, $w_2$]$^T$ $\epsilon$ $R^3$. Given some state s and action $a \epsilon 1, 1$, the featurization of this state, action pair is: $[s, a, 1]^T$. To linearly approximate the Q-values, we compute $\hat{q}(s, a; w) = w_0 s + w_1 a + w_2$ .
Given the parameters w and a single sample $(s, a, r, s^{'})$, the loss function we will minimize is
$J(w) = (r + \gamma \underset{a}{max} \hat{q}(s^{'}, a; w^{-}) \hat{q}(s, a; w))^2$ (3)
where $\hat{q}(s^{'}, a; w^{-})$ is a target network parametrized by fixed weights $w^{-}$.
Suppose we currently have the weight vectors $w = [1, 1, 1]^T$ and $w^{=[1,1,2]^T}$ and we observe a sample $(s = 2, a = 1, r = 1, s = 1)$, perform a single gradient update to the parameters w given this sample. Use the learning rate $= \frac{1}{4}$. Write out the gradient $\nabla_w J(w)$ as well as the obtained new parameters w' after the update. Show all work.
**Ans:** X = $[s, a, 1]^T$; w = $[w_0, w_1, w_2]^T$
$J(w) = (r + \gamma \underset{a'}{max} \hat{q}(s', a'; w^{-}) \hat{q}(s, a; w))^2$

$w^{-} = [1, -1, -2]^T$

$w_c = [-1, 1, 1]^T$

$(s, a, r, s^{'})^{=(2,-1,-1,1);\alpha=\frac{1}{4}}$

$\hat{q}(s, a; w) = w_0 s + w_1 s + w_2$

Let, $r + \gamma \underset{a'}{max}\, \hat{q}(s', a'; w^{-}) =$ g

$\nabla_w J(w) = \nabla_w (g - \hat{q}(s, a; w))^2$

$2(g - \hat{q}(s, a, w))\nabla_w \hat{q}(s, a; w)$
value of $\hat{q}(s, a; w) = w_0 s + w_1 a + w_2 = -2 - 1 + 1$
$= -2$

value of g : $r + \gamma \underset{a'}{max}(q'(s'^{\hat{},a',w)}$

$r + \gamma \underset{a'}{max}[s - a' - 2]; \gamma = 1$

$-1 + 1\underset{a'}{max}[s^{'-a^{'-2]}}$

$-1 + 1\underset{a'}{max}[1 - a^{'-2]}$

$g = -1$

value o $\nabla_w \hat{q}(s, a; w) = (\frac{\partial \hat{q}}{\partial w_0}, \frac{\partial \hat{q}}{\partial w_1}, \frac{\partial \hat{q}}{\partial w_2}$)=(2,-1,1)

$\nabla_w J(W)$=2(-1+2) $\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

Weight Update:
$W_{new} = W_{old} + \alpha \nabla_w J(w)$
$\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 4 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ 1.5 \end{bmatrix}$

3. (Convergence) Suppose, the optimal Q function Q (s, a) is exactly representable by some neural network architecture N . Suppose we perform Q-learning on this MDP using the architecture N to represent the Q-values. Suppose we randomly initialize the weights of a neural net with architecture N and collect infinitely many samples with infinite exploration. Are we guaranteed to converge to the optimal Q function Q*(s, a)? Explain your answer.

Ans:

# 4 Policy Gradient Methods

(5+5+5=15 Marks)

1. Consider policy parameterized using the soft-max in action preferences $\pi(a|s, \theta) = \frac{\exp^{h(s,a,\theta)}}{\sum_b \exp^{s,b,\theta}}$ with linear action preferences $h(s, a, \theta) = \theta^T x(s, a)$. For this parameterization, prove that the eligibility vector is $\nabla ln\pi(a|s, \theta) = x(s, a) - \sum_b \pi(b|s, \theta)x(s, b)$,

2. Write the update equation for REINFORCE.
3. Consider the advantage function defined as $A_\pi(s,a) := Q_\pi(s,a)V_\pi(s)$. Can it used as baseline for REINFORCE algorithm to improve variance? Explain.

**Ans 1**:$\pi(a|s;\theta) = \frac{\exp^{(s,a;\theta)}}{\sum_b \exp^{h(s,b;\theta)}}$

b(s,a;$\theta$) $= \theta^T x(s,a)$

To Prove: $\nabla ln\pi(a|s;\theta) = X(s,a) - \sum_b \pi(b|s;\theta)X(s,b)$

Proof: $\nabla ln\pi(a|s,\theta) = \frac{\nabla\pi(a|s,\theta)}{\pi(a|s,\theta)}$

let m=$\sum_b \exp^{(s,b;\theta)}$ $\nabla\pi(a|s,\theta) = \nabla(\frac{\exp^{\theta^T X(s,a)}}{\sum_b \exp^{\theta^T X(s,b)}})$

Solving for ifh component=[showing for component i of $\theta$ i.e. $\theta_i$, other component will bare the same solution]
$m[X_i(s,a)\exp^{\theta^T X(s,a)}] - \exp^{\theta^T X(s,a)}[\sum_b X_i(s,b)\exp^{h(s,b;\theta)}]$

$\frac{X_i(s,a)\exp^{\theta^T X(s,a)}}{m} - \frac{\exp^{\theta^T X(s,a)}}{m}[\sum_b X_i(s,b)[\frac{\exp^{h(s,b;\theta)}}{m}]]$
$\pi(a|s;\theta)$
Now, $(\frac{\nabla\pi(a|s;\theta)}{\pi(a|s;\theta)}) = \frac{X_i(s,a)\pi(a|s;\theta)}{\pi(a|s;\theta)} - \frac{\pi(a|s;\theta)}{\pi(a|s;\theta)}[\sum_b X_i(s,b)[\frac{\exp^{h(s,b;\theta)}}{\sum_{b'} \exp h(s,b';\theta)]]}$
Notice that $\frac{\exp^{h(s,b,\theta)}}{\sum_{b'} \exp^{h(s,b',\theta)}}$
$X_i(s,a) - \sum_b X_i(s,b)\pi(b|s,\theta)$

writing this equation in vector form we get $X(s,a) - \sum_b \pi(b|s,\theta)X(s,b)$

therefore, $\frac{\nabla\pi(a|s;\theta)}{\pi(a|s;\theta)} = \nabla ln(\pi(a|s;\theta)) = X(s,a) - \sum_b \pi(b|s;\theta)X(s,b)$

**Ans 2:** $\theta < -\theta + \alpha\gamma^t G\nabla ln\pi(A_t|s_t,\theta)$

## 5 Bonus question on Model approximation

(Bonus marks: 20) Let $M = (S, A, R, P, )$ be an MDP with $|S| < \infty, |A| < \infty$ and $\gamma\epsilon[0,1)$. Let  be a modification of M to be specified below. We compare the optimal value functions and policies in M and . Let $\hat{M} = (S, A, \hat{R}, P, )$ where $|\hat{R}(s,a)R(s,a)|\epsilon$ for all $s\epsilon S$ and $a\epsilon A$. Besides the rewards, all other components of  stay the same as in M. Prove that $V^* - \hat{V}^* \leq \frac{\epsilon}{1\gamma}$ . Will M and  have the same optimal policy? Briefly explain. Note $V^*$ and $\hat{V}^*$ are optimal value functions in M and , respectively. The following questions may guide you to the proof.

1. Let $P_{ss'\pi,t}$ be the probability of being in state $s^{'}$ after t time-steps of following policy $\pi$ starting from state s, i.e., $P_\pi(S_t = s^{'|S_0=s)}.Express$V$^\pi(s)$

interms of $\{P^{\pi,t}_{ss'}\}^{\infty}_{t=1}$ and $R^{\pi}_{s} := E_{a} \ _{\pi}[R(s,a)]$

2. For a given policy $\pi$, show that $\hat{V}^{\pi}(s)V^{\pi}(s) \leq \frac{\epsilon}{1-\gamma}$.

3. Use the above result to obtain the bound on $\hat{V}^{*} - V^{*}$. Note that it is not known whether the optimal policy for M and  are the same.

4. Do M and  have the same optimal policies? Prove or give a counter example. Hint: Think of a simple grid world.

**Ans1:** Notice that $V^{*}(s) = \sum_{t=1}^{\infty} r^{t-1} \sum_{s_t^{'} \epsilon S} P^{\pi}_{ss_t^{'}} R_{\pi}(s_t^{'})$

**Ans2:**$\hat{V^{\pi}}(s) - V^{\pi}(s) \leq |\hat{V^{\pi}}(s) - V^{\pi}(s)|$
$|\hat{V^{\pi}}(s) - V^{\pi}(s)| = |\sum_{t=1}^{\infty} \gamma^{t-1} \sum_{s_t^{'} \epsilon S} P^{\pi}_{ss_t^{'}} \hat{R}_{\pi}(s) - \sum_{t=1}^{\infty} \sum_{s_t^{'} \epsilon S} P^{\pi}_{ss_t^{'}} R_{\pi}(s)|$

—$\sum_{t=1}^{\infty} \gamma^{t-1} \sum_{s_t^{'} \epsilon S} P^{\pi}_{ss_t^{'}} [\hat{R}_{\pi}(s) - R_{\pi}(s)]|$

$\leq \sum_{t=1}^{\infty} \gamma^{t-1} \sum_{s_t^{'} \epsilon S} P^{\pi}_{ss_t^{'}} [\hat{R}_{\pi}(s) - R_{\pi}(s)]$

Now, $\hat{R}_{\pi}(s) - R_{\pi}(s) = |E_{a} \ _{\pi}[\hat{R}(s,a)] - E_{a} \ _{\pi}[R(s,a)]|$
$|E_{a} \ _{\pi}[\hat{R}(s,a) - R(s,a)]|$ (Visiting linearity of expectation)

$\leq E_{a} \ _{\pi}[|\hat{R}(s,a) - R(s,a)|]$

$\leq E_{a} \ \pi[\epsilon]$

$\epsilon$

satisfying this we get, $|\hat{V^{\pi}}(s) - V^{\pi}(s)| \leq \sum_{t=1}^{\infty} \sum_{t}^{'} P^{\infty}_{ss_t^{'}} \epsilon \ (\epsilon) \sum_{t=1\infty} \gamma^{t-1} \sum_{s_t^{'}} P^{\pi}_{ss_t^{'}}$

Now, notice that for $s_t^{'} \epsilon S$
$\sum P_{ss_t^{'}=1}$

therefore, $|\hat{V^{\pi}}(s) - V^{\pi}(s)| \leq \epsilon \sum_{t=1}^{\infty} \gamma^{t-1} = \frac{\epsilon}{1-\gamma}$

$\hat{V^{\pi}}(s) - V^{\pi}(s) \leq |\hat{V^{\pi}}(s) - V^{\pi}(s)| \leq \frac{\epsilon}{1-\gamma}$

**Ans3:** Let $\pi^{*}$ be the optimal policy for M.
Let $\hat{\pi^{*}}$ be the optimal policy for $\hat{M}$.
$|\hat{V^{\pi^{*}}}(s) - V^{\pi^{*}}(s)| \leq \frac{\epsilon}{1-\gamma}$

$|\hat{V}^{\hat{\pi}^{*}}(s) - V^{\hat{\pi}^{*}}(s)| \leq \frac{\epsilon}{1-\gamma}$

$|\hat{V^{\pi^{*}}}(s) - V^{\pi^{*}}(s)| \leq |\hat{V}^{\hat{\pi}^{*}}(s) - V^{\hat{\pi}^{*}}(s) + V^{\hat{\pi}^{*}}(s) + \hat{V^{\pi^{*}}}(s) - V^{\pi^{*}}(s) - \hat{V^{\pi^{*}}}(s)|$

$\leq |\hat{V}^{\hat{\pi}^*}(s) - V^{\hat{\pi}^*}(s)| + |\widehat{V^{\pi^*}}(s) - V^{\pi^*}(s)| + |\widehat{V^{\pi^*}}(s) - V^{\hat{\pi}^*}(s)|$

$\leq \frac{2\epsilon}{1-\gamma} + |\widehat{V^{\pi^*}}(s) - V^{\hat{\pi}^*}(s)|$

$V^*(s) = \underset{\pi}{max} V^\pi(s)$
$\hat{V}^*(s) = \underset{\hat{\pi}}{max} \hat{V}^{\hat{\pi}}(s)$

$\hat{V}^*(s) - \hat{V}^{\hat{\pi}^*}(s) + \hat{V}^{\pi^*}(s) - V^*(s)$
$\leq \hat{V}^{\pi^*}(s) - V^*(s) \leq \frac{\epsilon}{1-\gamma}$

$\hat{V}^*(s) - V^*(s) \leq \frac{\epsilon}{1-\gamma}$

since, $\hat{V}^*(s) - \hat{V}^{\pi^*}(s) \leq 0$ $\pi^*$ may not be optimal for .

$\hat{V}^{\pi^*}(s) - V^*(s) \leq \frac{\epsilon}{1-\gamma}$ since , it is the same ploicy $\pi^*$

adding, $\hat{V}^*(s) - V^*(s) \leq \frac{\epsilon}{1-\gamma}$

similarly, $V^*(s) - V^{\hat{\pi}^*}(s) \leq 0$
$V^{\hat{\pi}^*}(s) - \hat{V}^{\hat{\pi}^*}(s) \leq \frac{\epsilon}{1-\gamma}$

$V^*(s) - \hat{V}^*(s) \leq \frac{\epsilon}{1-\gamma}$

$|V^*(s) - \hat{V}^*(s)| \leq \frac{\epsilon}{1-\gamma}$

**Ans4:** No, it is not necessary that M and  have the same optimal policies. A slight change in the reward for some action may have it more favourable and optimal compared to other actions.

Consider the following counterexample in fig 3(below): Two actions available from $S_1$ $R(s_1, L) = 1$ $R(s, L)\hat{} = 1 - \epsilon$
$R(s_1, D) = 1$ $R(s, D)\hat{} = 1 - \epsilon$

condition given $|\hat{R}(s, a) - R(s, a)| \leq \epsilon$ is satisfied.

But optimal policy $\pi$ for M favours both actions-taking either actions is optimal.

On the other hand the optimal policy $\hat{\pi}$ for $\hat{M}$ favours only one action- just the down actions os optimal. Even in his simple case the optimal policies are different.

let $V^* = V_1^\pi$ and $\hat{V}^* = \hat{V_2^\pi}$
Now, $\hat{V_2^\pi} - V_2^\pi \leq \frac{\epsilon}{1-\gamma}$
$\hat{V_1^\pi} - V_1^\pi \leq \frac{\epsilon}{1-\gamma}$

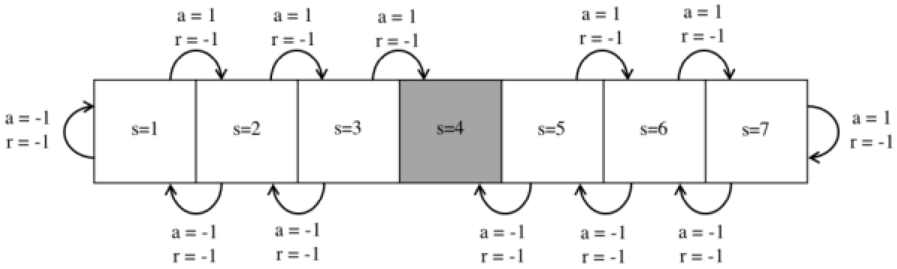

Figure 2: Chain MDP

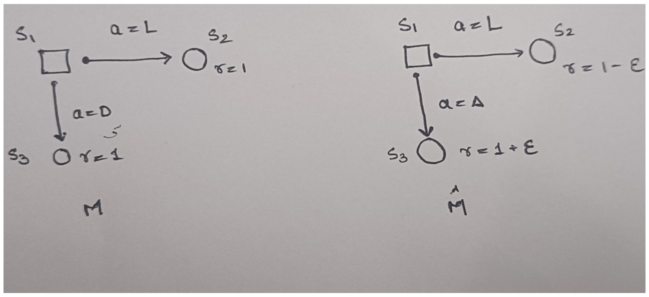

Figure 3: