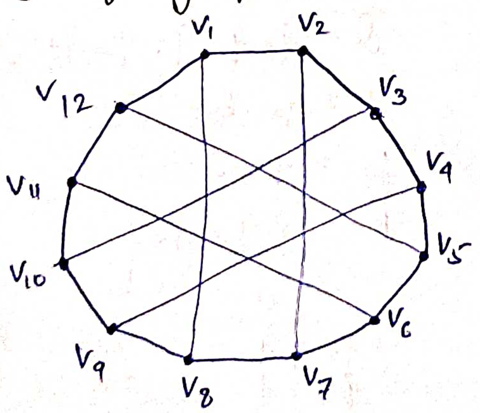
Q1. In theoretical computer science, how important are computing theory and Complexity theory? Describe their differences and how they are related to one another.

Ans. Derived from theoritical computer science, Theory of Computation deals with how efficiently problem can be solved on a model of computation using an algorithm. The components of it includes —
- Computability theory
- Complexity theory
- Automata theory

The theories of computability and complexity are closely related:
In Computability theory, the objective is to classify problems as ~~easy ones and hard ones whereas in Complexibility~~ solvable ones or unsolvable ones whereas in Complexbility theory, the classification of problems is by those that are easy ones and those that are hard ones.

2. A graph G is said to be k-regular if every node in the graph has degree k.
a) Construct a 3-regular graph G = (V,E) with 12 nodes. Display the vertex set V and edge set E of the graph G.
b) Write down the formula by using which you constructed the edges for graph G.

A?
a)



$V = \{V_1, V_2, V_3, \ldots V_9, V_{10}, V_{11}, V_{12}\}$

$E = \{(V_1, V_2)(V_1, V_8)(V_1, V_{12})$
$(V_2, V_7)(V_2, V_3)(V_3, V_{10})(V_3, V_4)$
$(V_4, V_9)(V_4, V_5)(V_5, V_{12})(V_5, V_6)$
$(V_6, V_{11})(V_6, V_7)(V_7, V_8)(V_8, V_9)$
$(V_9, V_{10})(V_{10}, V_{11})(V_{11}, V_{12})\}$

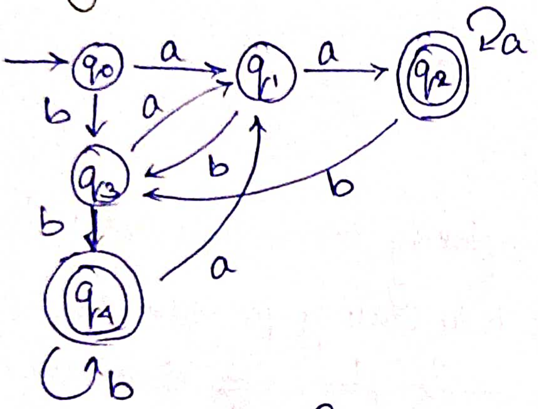
b) The total no. of edge of a k-regular with n vertices $= (n*k)/2$

Here, $n = 12$
$k = 3$

Thus $|E| = \frac{(12*3)}{2} = \frac{36}{2} = 18$

3. Construct the DFA for the following languages –

a) The language accepting all the strings such that last two symbols must be same over input alphabets $\Sigma = \{a, b\}$



min # state = 5.

$L: \{xaa, xbb \mid x \in \Sigma^*\}$

DFA: $\{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

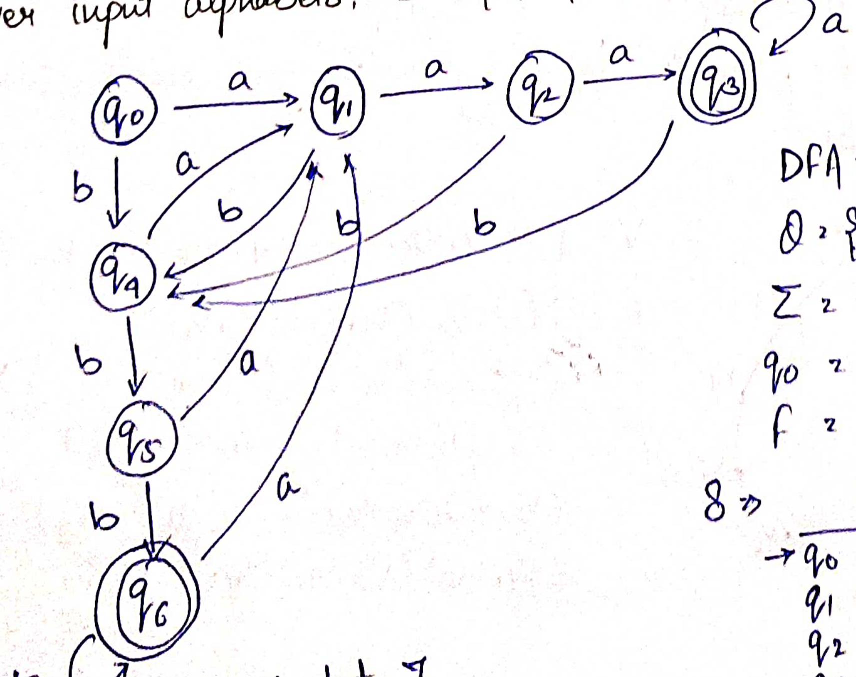
$q_0 = q_0$

$F = \{q_2, q_4\}$

$\delta =$

	a	b
q_0	q_1	q_3
q_1	q_2	q_3
q_2	q_2	q_3
q_3	q_1	q_4
q_4	q_1	q_4

b) The language accepting all the string that ends with 3a's or 3b's over input alphabets, $\Sigma = \{a, b\}$



min # state 7

DFA $= \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6$

$\Sigma = \{a, b\}$

$q_0 = q_0$

$F = \{q_3, q_6\}$

$\delta \Rightarrow$

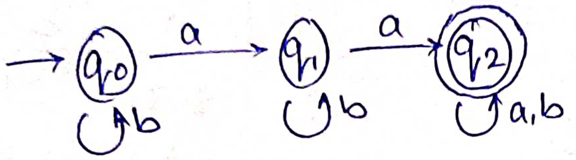
	a	b
$\rightarrow q_0$	q_1	q_4
q_1	q_2	q_4
q_2	q_3	q_4
q_3	q_3	q_4
q_4	q_1	q_5
q_5	q_1	q_6
q_6	q_1	q_6

4. Draw the state transition diagram and show the state transition table for the following DFA's

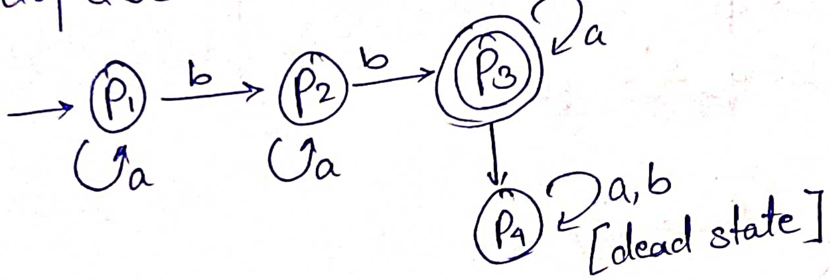
a) DFA for the language accepting all strings that contains atleast 2a's and exactly 2b's over input alphabet $\Sigma = \{a,b\}$.

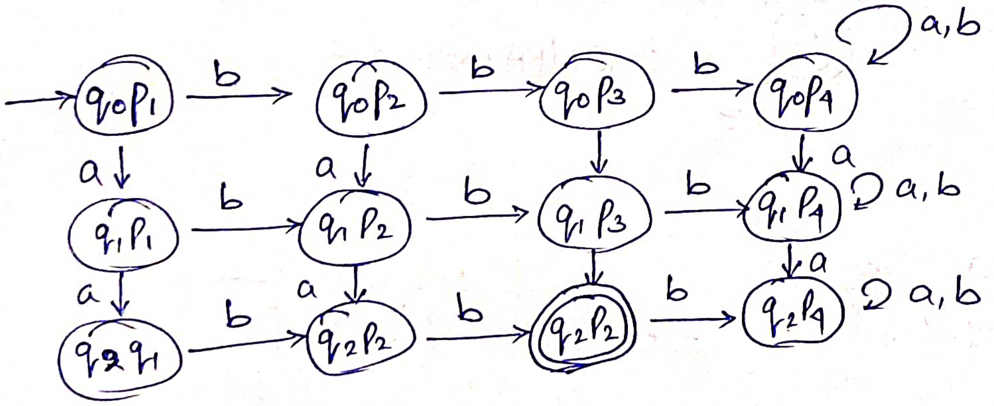
b) DFA for the language strings containing neither '00' nor '11' as substring over input alphabets $\Sigma = \{0,1\}$.

a) Atleast 2a's

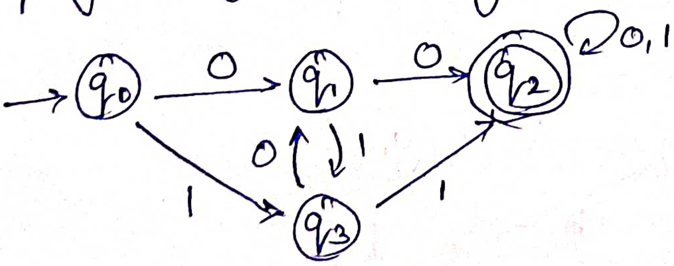


Exactly 2b's

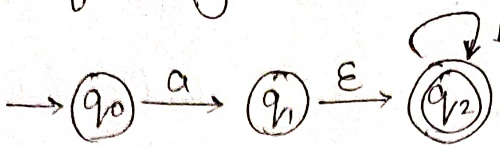




b) L' : Accepting strings containing '00' or '11' as substring



5. a) Convert the following NFA with ε to NFA without ε.



b) Convert the obtained NFA to its equivalent DFA.

⇒ a)

$q_0 \xrightarrow{\varepsilon^*} q_0 \xrightarrow{a} q_1 \xrightarrow{\varepsilon^*} q_1, q_2$

$q_0 \xrightarrow{\varepsilon^*} q_0 \xrightarrow{b} \phi$

$q_1 \xrightarrow{\varepsilon^*} q_1 \xrightarrow{a} \phi$

$q_1 \xrightarrow{\varepsilon^*} q_1 \xrightarrow{b} q_2 \xrightarrow{\varepsilon^*} q_2$

$q_2 \xrightarrow{\varepsilon^*} q_2 \xrightarrow{a} \phi$

$q_2 \xrightarrow{\varepsilon^*} q_2 \xrightarrow{b} q_2 \xrightarrow{\varepsilon^*} q_2$

	a	b
q_0	$\{q_1, q_2\}$	{ }
q_1	{ }	$\{q_2\}$
q_2	{ }	$\{q_2\}$

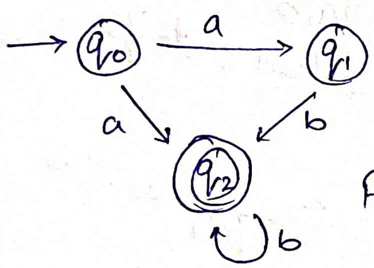


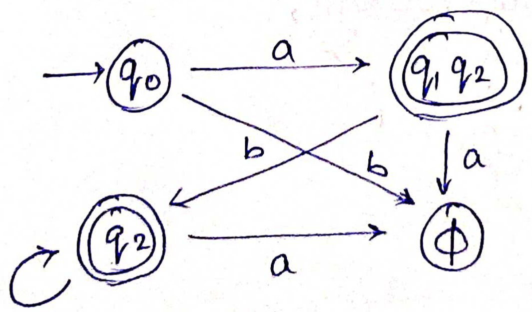
Fig. NFA without ε

b) NFA to DFA

	a	b
q_0	$\{q_1, q_2\}$	{ }
q_1	{ }	q_2
q_2	{ }	q_2

→

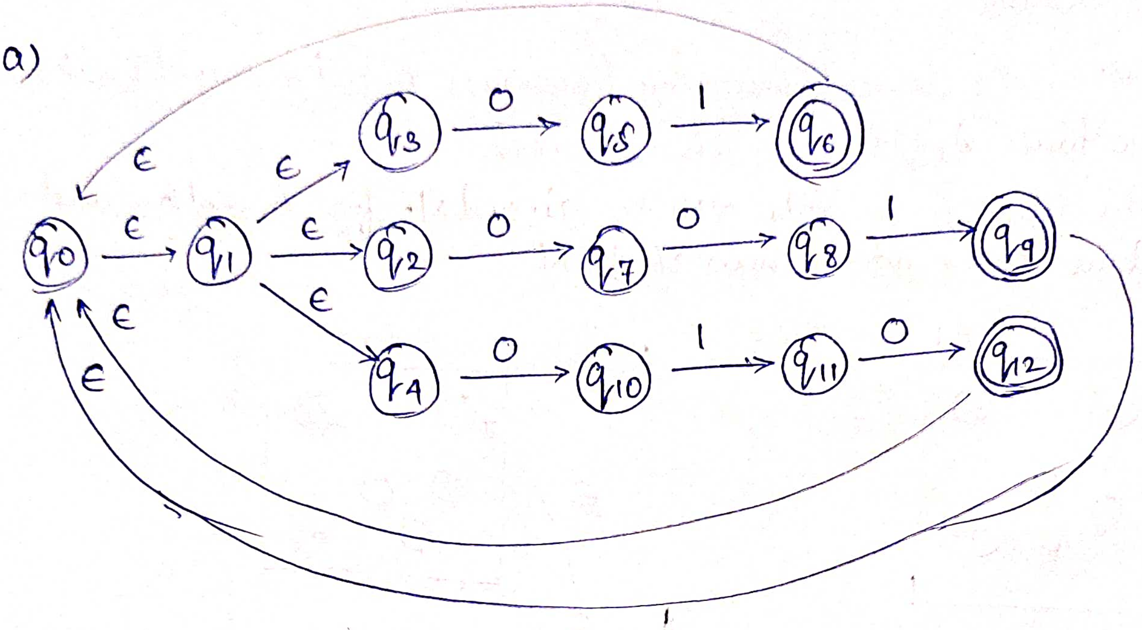
	a	b
q_0	$[q_1, q_2]$	ϕ
q_1	ϕ	$[q_2]$
q_2	ϕ	$[q_2]$



5. a) Design an NFA that recognizing the language $(01 \cup 001 \cup 010)^*$

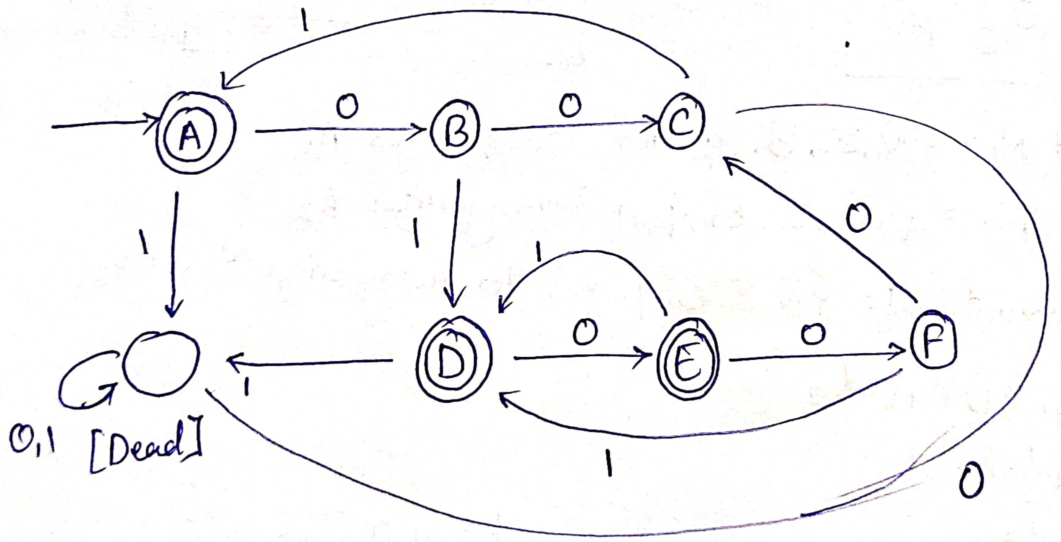
b) Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the states.

a)



b) NFA to DFA

$L = (01 \cup 001 \cup 010)^*$



7. Prove that the class of Regular Language is closed under

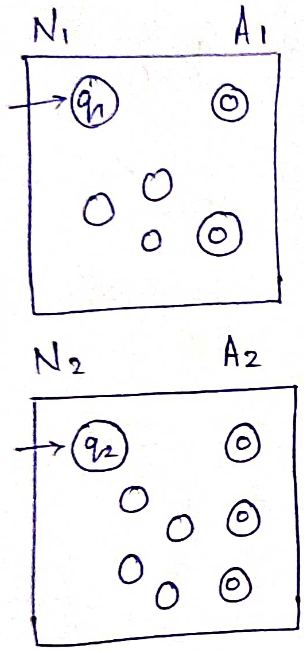
a) Union

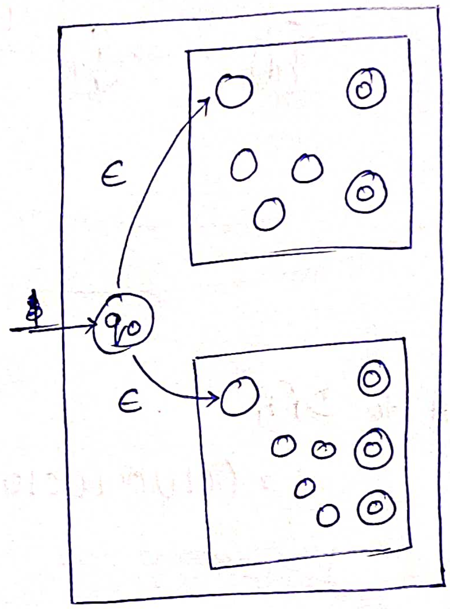
b) Concatenation and

c) Kleene Closure

a) Prove Idea: We have 2 regular languages A_1 and A_2 and need to prove that $A_1 \cup A_2$ is also Regular.

The idea is to take 2NFAs N_1 and N_2 for A_1 and A_2 and Combine them into 1 new NFA, N





Proof → Let $N_1 = \{Q_1, \Sigma, \delta_1, q_1, F\}$ recognizes A_1

$N_2 = \{Q_2, \Sigma, \delta_2, q_2, F\}$ recognizes A_2

Construct $N = \{Q, \Sigma, \delta, q_0, F\}$ to recognize $A_1 \cup A_2$

$Q = \{q_0\} \cup Q_1 \cup Q_2$

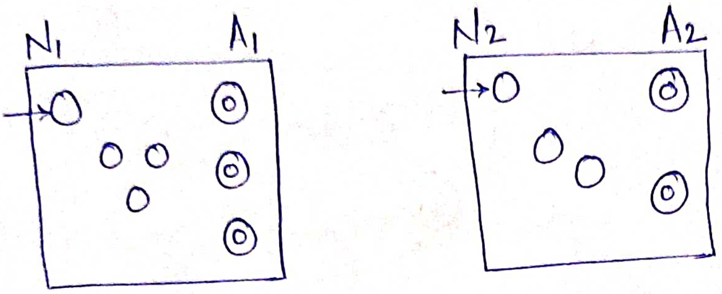
$q_0 =$ initial state

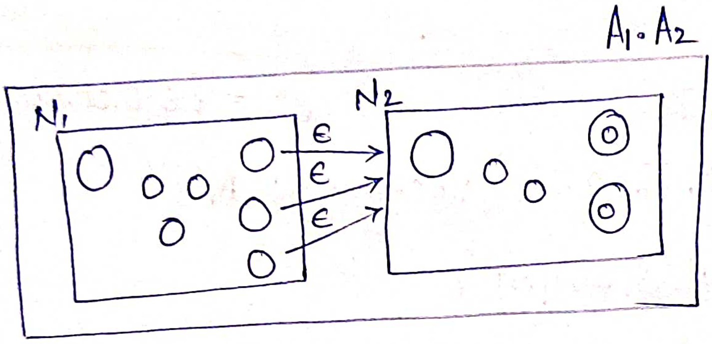
$F = F_1 \cup F_2$

Define δ so that for any $q \in Q$ and any $a \in \Sigma$

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q \\ \delta_2(q,a) & q \in Q \\ (q_1, q_2) & q = q_0 \text{ and } a = \epsilon \\ \phi & q = q_0 \text{ and } a \phi \epsilon \end{cases}$$

ii) Prove Idea: We have regular languages A_1 and A_2 and want to prove that $A_1 \circ A_2$ is regular.
The idea is to take two NFAs N_1 and N_2 for A_1 and A_2 and combine them into a new NFA 'N' as shown in figure





Proof: $N_1 = \{Q_1, \Sigma, \delta, q_1, F_1\}$ recogonizes A_1
$N_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2\}$ recogonizes A_2

Construct $N = \{Q, \Sigma, \delta, q_1, F_2\}$ to recogonizes $A_1 \cdot A_2$

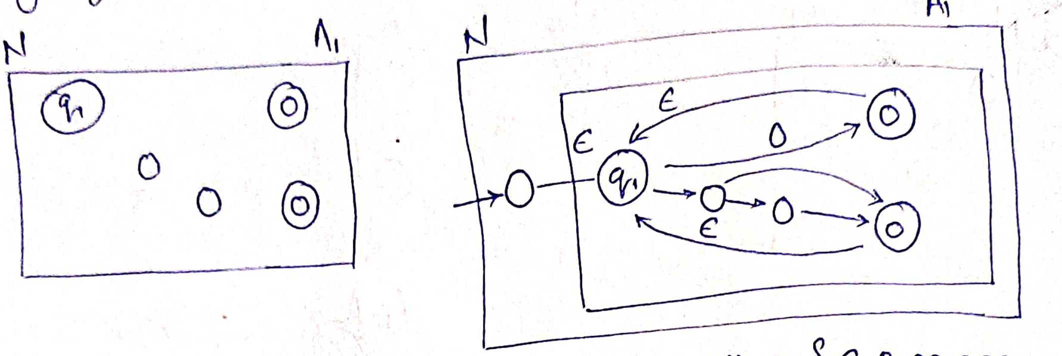
I. $Q = Q_1 \cup Q_2$
the state q_1 is the start state of N_1 and also the start state of N

II. The accept state F_2 are the same as the accept state of N_2 and also the accept state of N.

III. Define δ so that $q \in Q$ and $A \in \Sigma$

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } q \in \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \text{ and } a = \epsilon \\ \delta_2(q,a) & q \in Q \end{cases}$$

c) Prove idea - We have a regular language A_1 and want to prove the A_1^* also is a regular. We take an NFA N_1 for A_1 and modify it to recogonize A_1^*



$0^* = \{\epsilon, 0, 00, 000, \dots\}$

Prove: Let $N_1 = \{Q_1, \delta_1, \Sigma, q_1, F_1\}$ recogonizes A_1

Construct $N = \{Q, \delta, \Sigma, q_0, F\}$

$Q = \{q_0\} \cup Q_1$

$F = \{q_0\} \cup F_1$

$q_0 =$ initial state

Define δ so that for any $q \in Q$ for any $A \in \Sigma$

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin q_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \notin \epsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a = \epsilon \\ \{q_1\} & q = q_0 \text{ and } a = \epsilon \\ \phi & q = q_0 \text{ and } a \neq \epsilon \end{cases}$$

8 a) Let $\Sigma = \{a,b\}$. Write regular expression to define language consisting of strings w such that, w of length even.

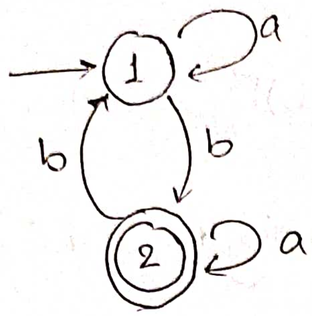
b) Let $\Sigma = \{a,b\}$. Write regular expression to define language consisting of strings w such that, w of length odd.

b) $R = (a+b)[(a+b)(a+b)]^*$

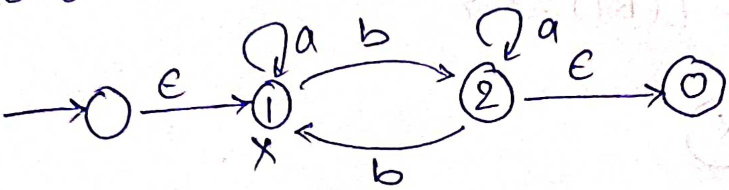
a) $R = [(a+b)(a+b)]^*$

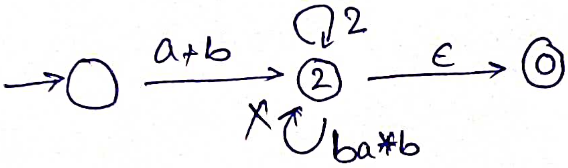
9. Convert the following finite automata to regular expression.

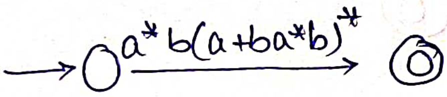
a)



⇒ a) State elimination method

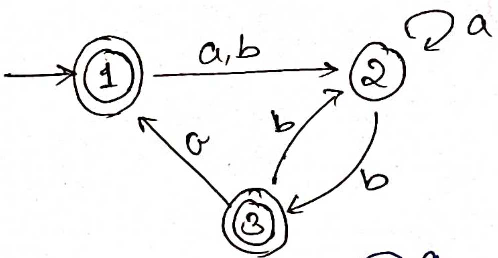


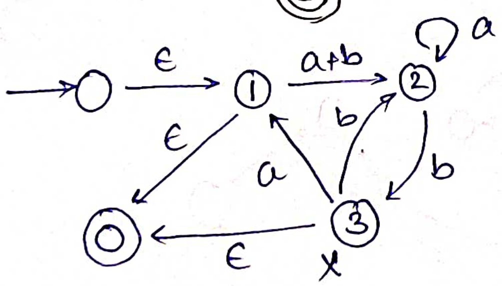


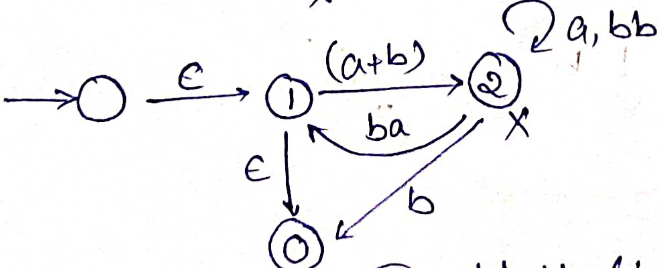


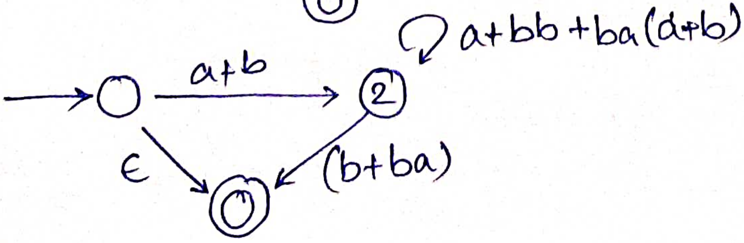
$\boxed{\therefore RE = a^*b(a+ba^*b)^*}$

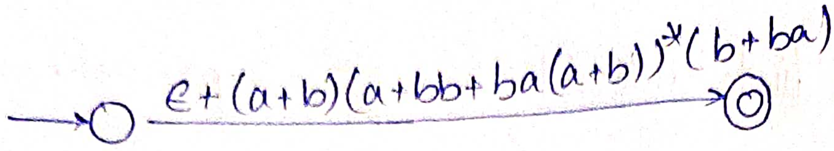
b)









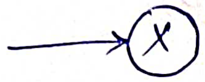


$RE = \epsilon + (a+b)(a+bb+ba(a+b))^*(b+ba)$

10. Design the Finite Automata for the following Regular Expressions:

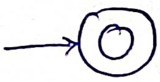
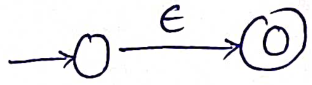
i) $R1 = \phi$

~~L(R) = { }~~ $L(R1) = \{\ \}$

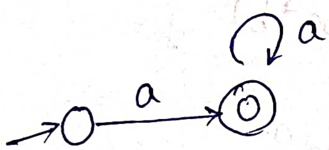


ii) $R2 = \epsilon$

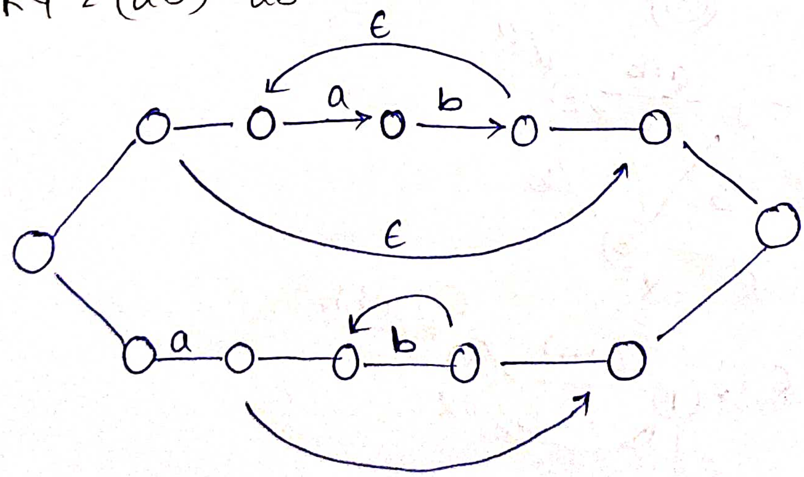
$L(R2) = \{\ \}$

 or 

iii) $R3 = a^+$



iv) $R4 = (ab)^* ab^*$



e) 0 $0^* 1^*$