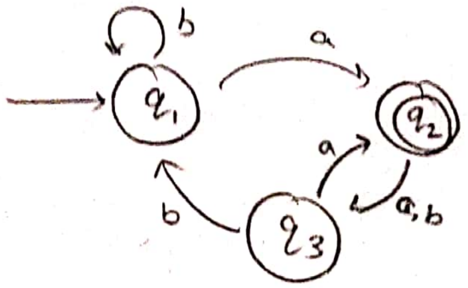
Roll No - 2019101053 Name - Archit Jain

Automata Phase I

Q 1 Give formal description of given machine



Sol

Since a finite automata is a tuple of 5 $(Q, \Sigma, \delta, q_0, F)$ where Q is finite set called states, Σ is finite set of alphabets, $\delta : Q \times \Sigma \to Q$ a transition function, $q_0 \in Q$ is start state and $F \subseteq Q$ which is set of accept state.

So by the machine diagram given we can say that

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

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

$\delta =$

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_3

$q_0 = q_1$ (indicated by arrow)

$F = \{q_2\}$ (indicated by double circle)

$\boxed{Q\ 2}$ The formal description of DFA M is $(\{q_1, q_2 \cdots q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$ where δ is given by following table Give state diagr

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

Sol)

Since Given

$$M = (Q, \Sigma, \delta, q, F)$$

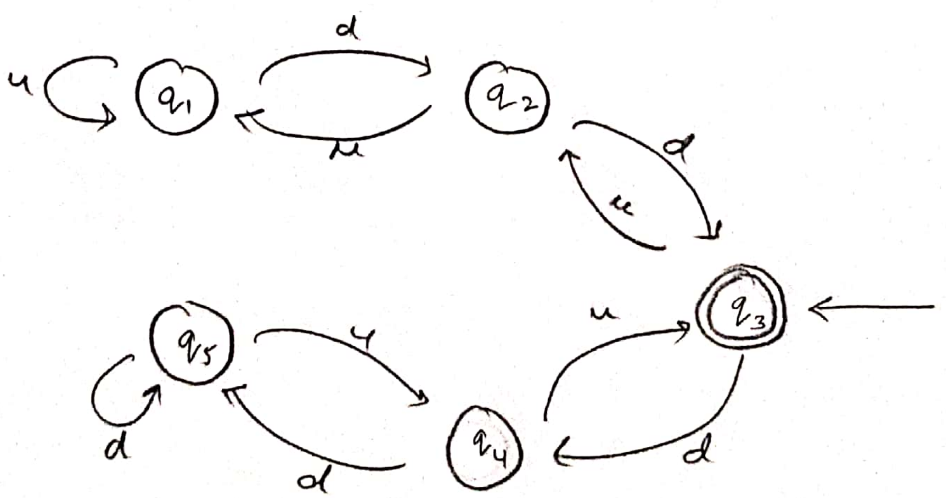
$$Q = \{q_1 \ldots q_5\}$$

$$\Sigma = \{u, d\}$$

δ : describe in table

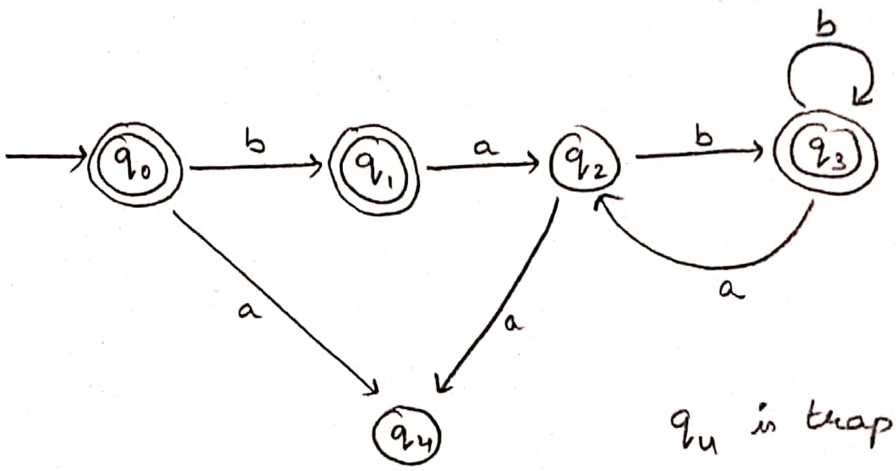
So, by this start state $= q_3$

accept state $= q_3$



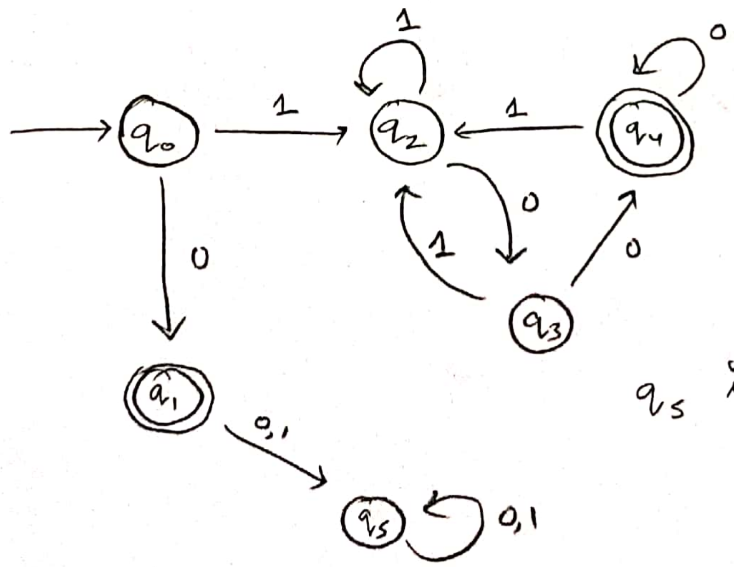
Q 3 Build deterministic finite automata for language $L = \{ w \in (a,b)^* :$ every a is w is immediately preceded and followed by b $\}$

Sol



q_4 is trap state

Q 4 Build DFA for $L = \{ w \in (0,1)^* :$ w corresponds to the binary encoding, without leading 0's of natural numbers that are evenly divisible by 4 $\}$

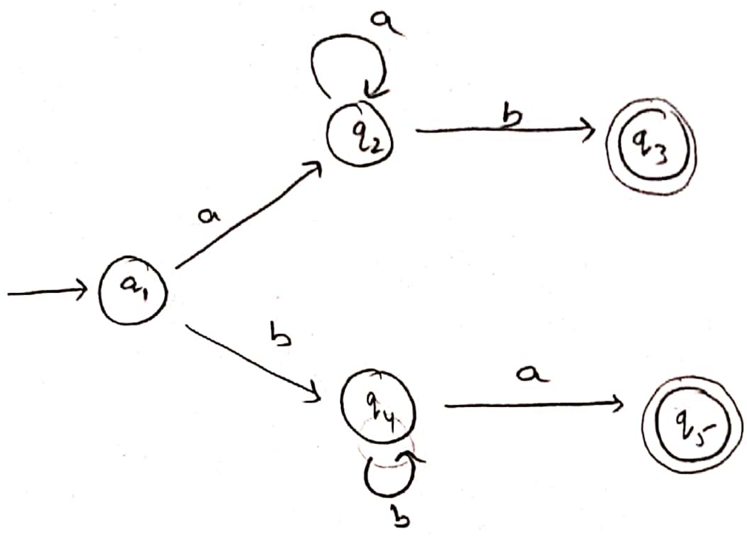


q_5 is trap state

Q 5 Build DFA for language

L = {w | w contains neither substring ab nor ba}

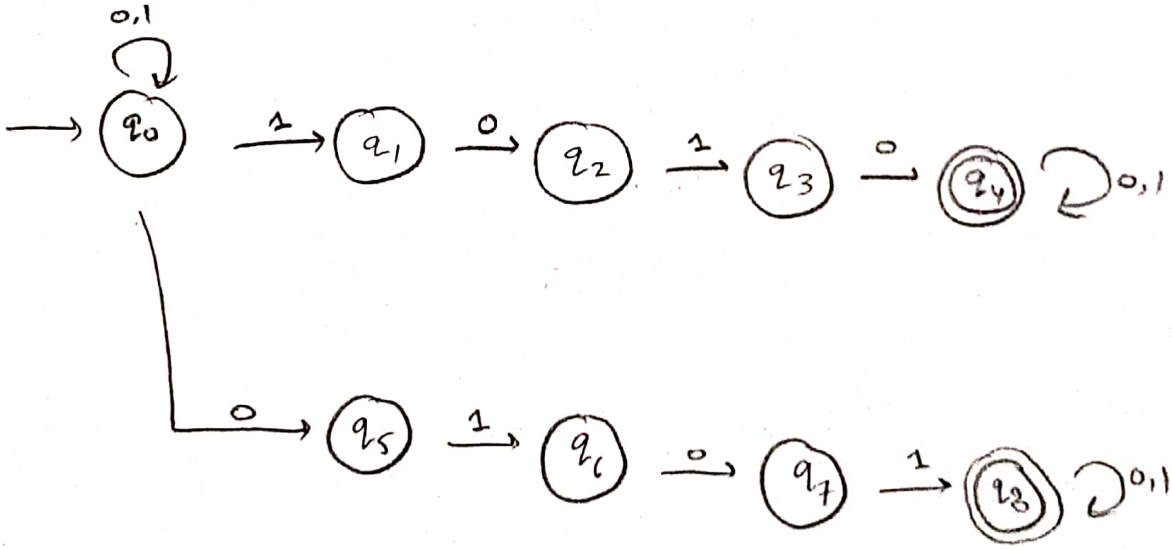
Sol



Q 6 Build NFA for language

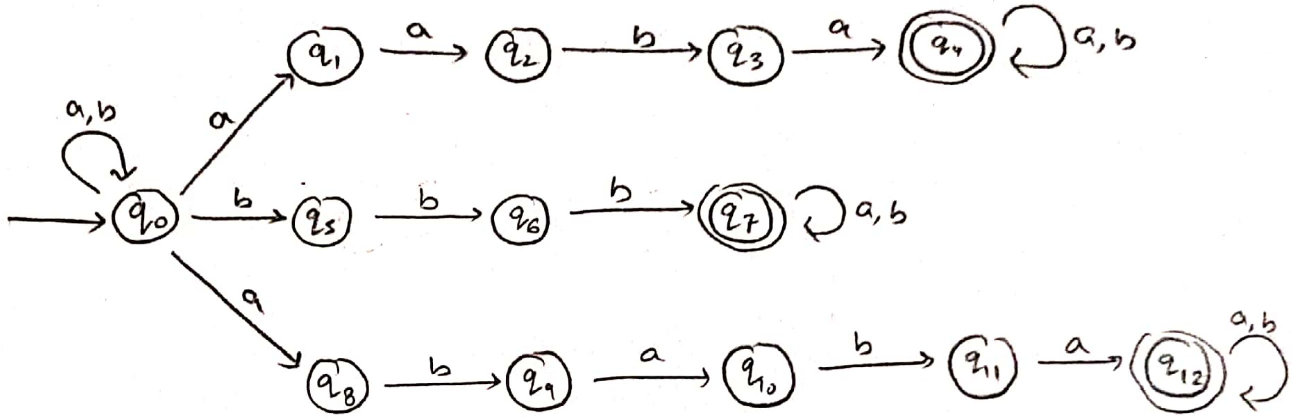
L = {w ∈ {0,1}* | w contains both 101 & 010 as substring}

Sol



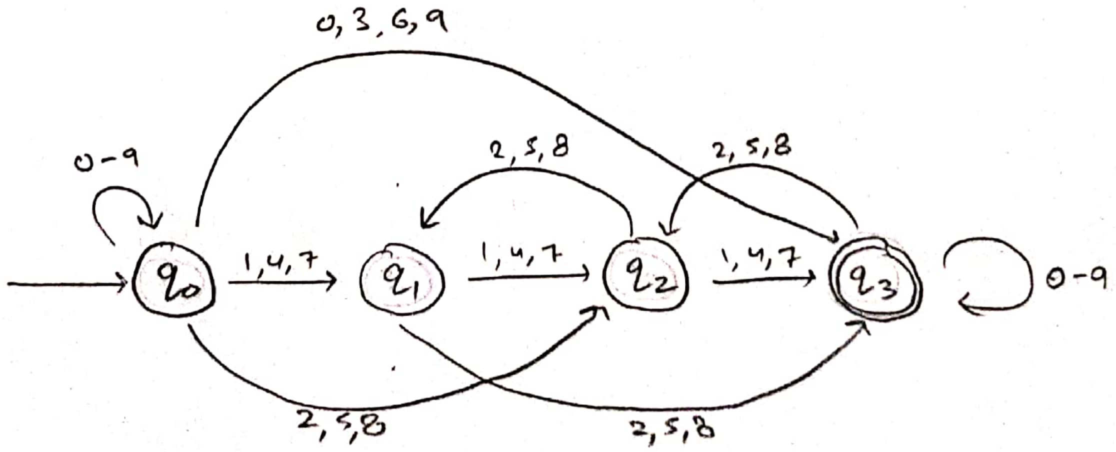
Q 7 Show NFA to the language $L = \{w \in \{a,b\}^* : w$ contains at least one instance of aaba, bbb or ababa $\}$

Sol



Q 8 Show NFA for $L = \{w \in \{0-9\}^* : w$ represents decimal encoding of a natural number whose encoding contains as a substring, ~~the~~ encoding of a natural number that is divisible by 3 $\}$

Sol



Q9) Prove every NFA can be converted to an equivalent one that has a single accept state.

Sol

Let M be a NFA and N be a another NFA with single accept state q_{final}

we take every accept state M and (i) make it non-accepting state, and (ii) add ε-transition from that state to q_{final}. Then we get NFA N

Let $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, then

$N = \langle Q \cup \{q_{final}\}, \Sigma, \delta', q_0, \{q_{final}\} \rangle$ for any $q \in Q$ & $a \in \Sigma$

$$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{if } a \neq \varepsilon \text{ or } q \notin F \\ \delta(q,a) \cup \{q_{final}\} & \text{if } a = \varepsilon \text{ and } q \in F \end{cases}$$

And $q'(q_{final}, a) = \phi$ for each $a \in \Sigma_\varepsilon$

Thus, M is equivalent to N

Hence, every NFA is converted to an equivalent one that has single accept state.

Q10 Prove that language $\{0^n 1^m 0^n \mid m,n \geq 0\}$ is not regular

Sol

Consider $L = \{0^n 1^m 0^n \mid m,n \geq 0\}$

Assuming L is a regular language and a string $S = 0^p 1 0^p$

Let us divide string S into 3 pieces x, y & z

$\therefore$ $S = 1^p 0 1^p = xyz$ where P is pumping length

Let us take $x = 0^{p-k}$, $y = 0^k$ and $z = 10^p$ $(k>0)$

Now $xy^0z \notin L$ because $P-k < P$

$\therefore$ Our assumption that L is regular is wrong

$\therefore$ By pumping lemma, it is proved that L is not regular.

Q 11 Give regular expressions generating following lang over alphabet $\Sigma = \{0,1\}$

(a) $L = \{w \mid w$ begins with a 1 & ends with 0$\}$

(b) $L = \{w \mid w$ containing substring 0101$\}$

(c) $L = \{w \mid w$ has length atleast 3 & 3rd symbol is 0$\}$

Sol (a) $R = 1\Sigma^*0 = 1(0+1)^*0$

(b) $R = \Sigma^* 0101 \Sigma^* = (0+1)^* 0101 (0+1)^*$

(c) $R = \Sigma\Sigma 0 \Sigma^* = (0+1)(0+1)0(0+1)^*$

Q12

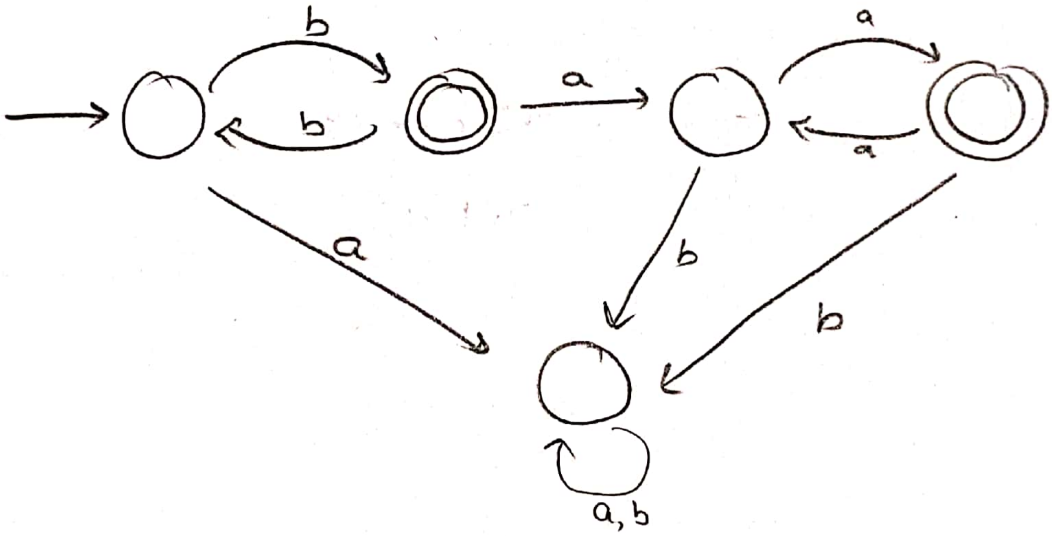
Let $D = \{w | w$ contains an even number of a's and an odd number of b's and doesn't contain the substring ab $\}$. Give DFA with five states that recognizes D & a regular expression that generates D.

Sol

So we can say that

$D = \{w | w$ contains odd numbers of b & even a $\}$

State Diagram



So by this we can say that the accept string are like $\{b, baa, bbbaaa \ldots\}$ and can be expressed as combination of 2 languages D_1 & D_2

$D_1 = \{w | w$ contain odd b's $\}$

$D_2 = \{w | w$ contain even a's $\}$

$D = D_1 \circ D_2$

Now, lets say

R_1 be regular expression generating D_1
R_2 be regular expression generaty D_2
R be regular expression generating D

$$R = R_1 \circ R_2$$

$$R_1 = b(bb)^*$$

$$R_2 = (aa)^*$$

$$R = b(bb)^* \circ (aa)^* = b(bb)^*(aa)^*$$

$\Rightarrow$ Therefore the regular expression generates language D is $b(bb)^*(aa)^*$

Q13 Show that if M is a DFA that recognizes language B, swapping accept & non-accept states in M yeilds a new DFA recognizing the complement of B.

Sol

→ M is a DFA that recognizes regular language B (Given)

→ Let say M' be a new DFA that has swapped accept and non-accept states in M

→ So if M' accepts a string x & if similar string when run on other machine (M) it will stops on a non-accepted state & vice versa with other string y accepted by M

→ So if $x \in B$ the $x \notin \bar{B}$ & vice versa

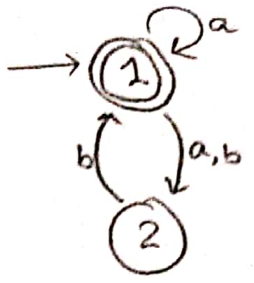
→ So M will not accept string accepted by M' & vice-versa

→ Therefore if M recognizes language B, there exists M' which recognizes complement of B which is also regular

Hence Proved

Q14 Convert given DFA to NFA

DFA:



Sol

$Q^1 = P(Q)$, where Q^1 is the subset of all sets of Q.

So, $Q^1 = \{\phi, \{1\}, \{2\}, \{1,2\}\}$

For an element R in Q^1 and a in set of alphabets Σ

Calculate $\delta'(R,a) = \{q \in Q \mid q \in \delta(r,a) \text{ for some } r \in R\}$

Here δ' performs the transition on r for some values of a

$$\delta^1(\phi, a) = \delta(\phi, a) = \phi$$

$$\delta^1(\phi, b) = \delta(\phi, b) = \phi$$

$$\delta^1(\{1\}, a) = \delta(1, a) = \{1,2\}$$

$$\delta^1(\{1\}, b) = \delta(1, b) = \{2\}$$

$$\delta^2(\{2\}, a) = \delta(2, a) = \phi$$

$$\delta^1(\{2\}, b) = \delta(2, b) = \{1\}$$

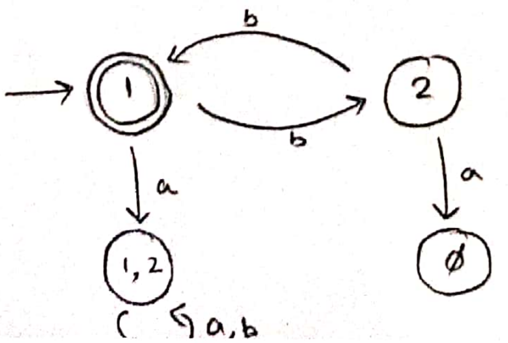
$$\delta^1(\{1,2\}, a) = \delta(\{1,2\}, a) = \delta(1,a) \cup \delta(2,a) = \{1,2\}$$

$$\delta^1(\{1,2\}, b) = \delta(\{1,2\}, b) = \delta(2,b) \cup \delta(2,b) = \{1,2\}$$

$q_0' = \{q_0\}$ where q_0 is start state in NFA

$F' = \{R \in Q^1 \mid R \text{ contains an accept state of NFA}\}$

NFA:



Q15 Given NFA recognizing the language

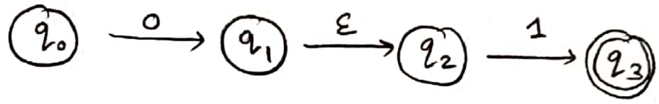
$(01 \cup 001 \cup 010)^*$

Sol

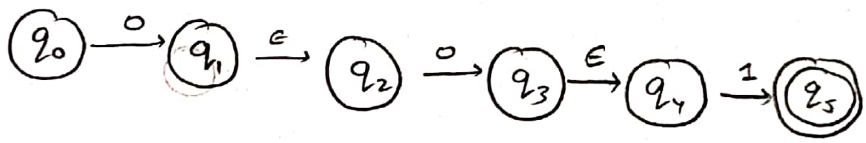
$L = ((01) \cup (001) \cup (010))^*$

Assume that M as NFA that recognizes language L

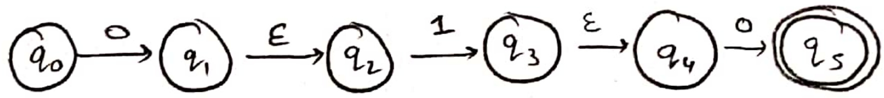
01 :



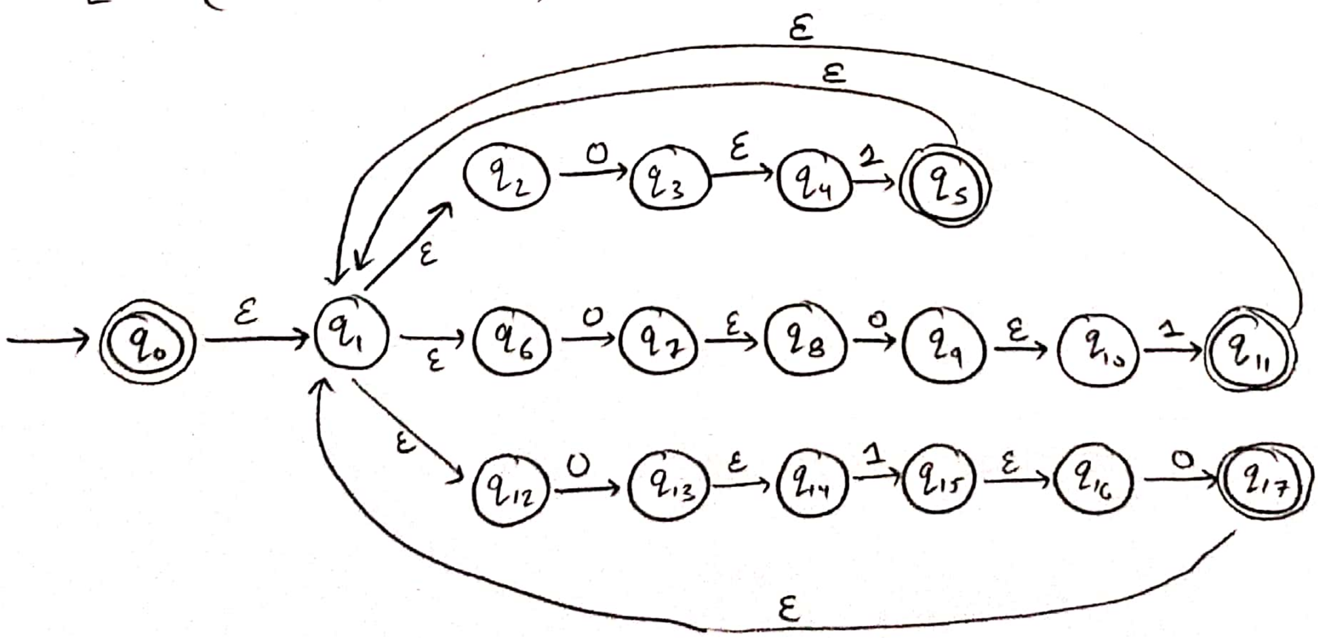
001 :



010 :



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

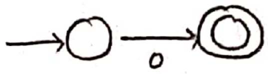


Q16 Convert regex $(0 \cup 1)^* \, 000 \, (0 \cup 1)^*$ to NFA

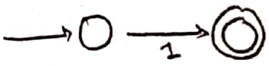
Sol

$R = (0 \cup 1)^* \, 000 \, (0 \cup 1)^*$

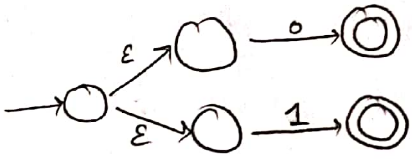
0 :



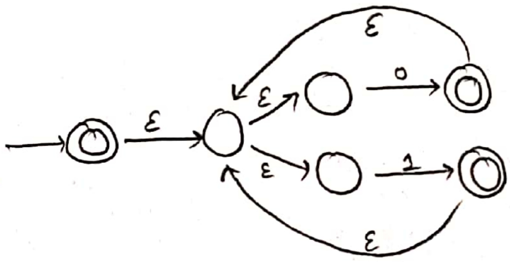
1 :



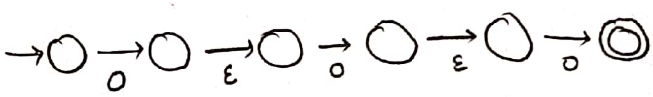
$0 \cup 1$:



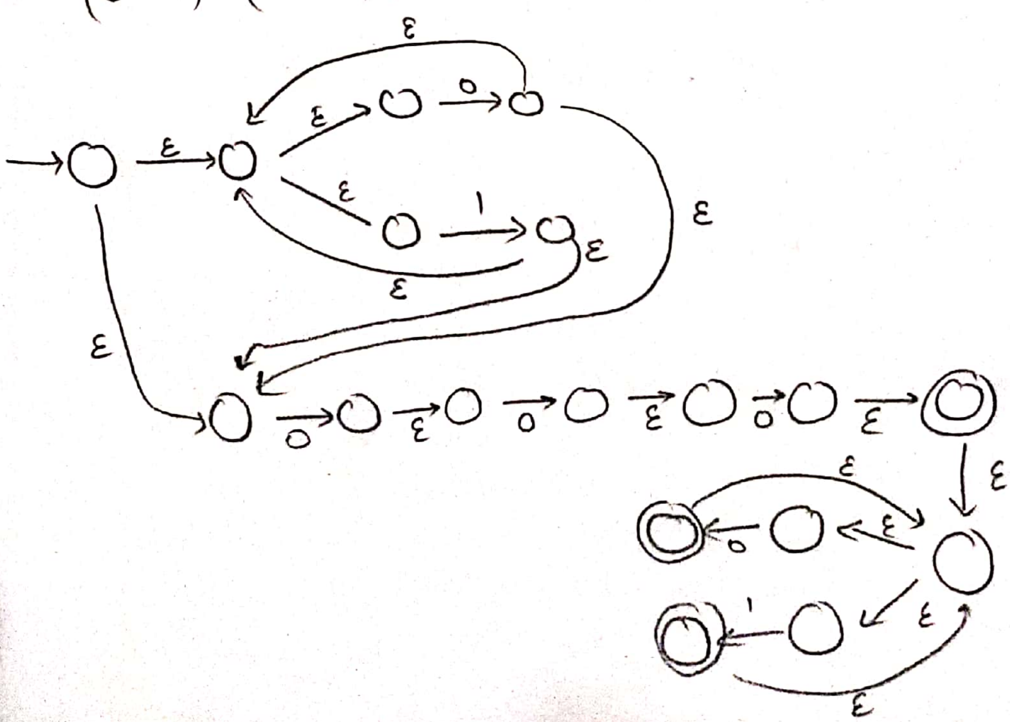
$(0 \cup 1)^*$:



000 :



$R = (0 \cup 1)^* \, (0 \; 00) (0 \cup 1)^*$



Q17 Give regular expression for given language
$L = \{w \mid w \text{ contain substring } 0101\}$

Sol

Consider the language $L = \{w \mid w \text{ contain substring } 0101\}$ over alphabet $\Sigma = \{0,1\}$

R be regular expression generates L

$$R = \Sigma^* 0101 \Sigma^*$$
$$= (0+1)^* 01\,01\,(0+1)^*$$

The strings accepted by regular expression are
0101, 001011, 1101010...

$\Rightarrow$ Therefore the regular expression is $(0+1)^* 0101 (0+1)^*$

Q18 Give regular expression for
$L = \{w \mid \text{every odd position of } w \text{ is } 1\}$

Sol

Let R be regular expression generates L

$$R = (1\Sigma)^* (\varepsilon+1)$$
$$= (1(0+1))^* (\varepsilon+1)$$

The strings accepted by regular expression are
101, 111, 1010, ...

$\Rightarrow$ Therefore the regular expression is $(1(0+1))^* (\varepsilon+1)$