Automata theory -

Ayush sharma (2019101004)

Q.1 = From the given state diagram of two DFAs, M_1 & M_2. Answer the following.

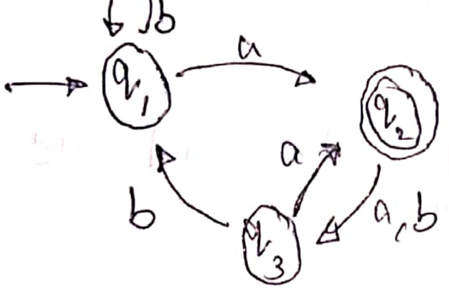
(a) What is the start state?

(b) What are set of accept states?

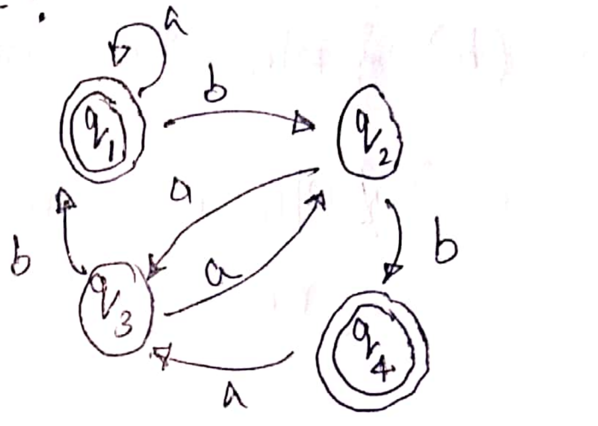
(c) Sequence of states does machine go on input aabb?

(d) Does machine accept 'aabb'?

(e) Does machine accept ε?



M_1



M_2

Q.2 = Give formal description of above machine M_1 & M_2.

Q.3 = For $M = \langle \{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\} \rangle$.

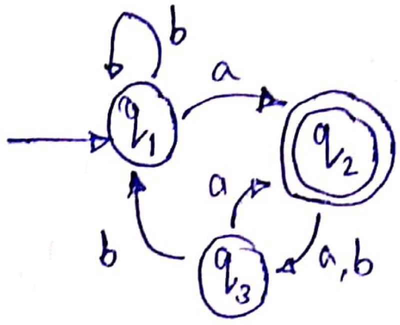
where δ :

	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

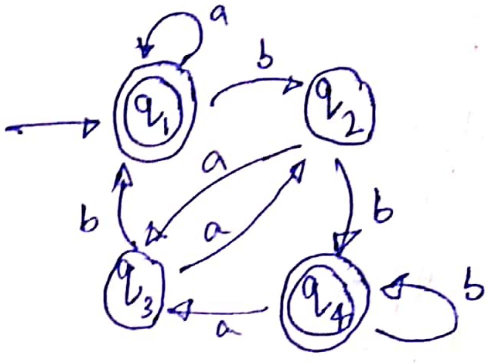
Find state diagram.

Sol^n
(1)

Solutions



M_1



M_2

(a) Start state is state 'q_1' for M_1 & M_2 both.

(b) Accept state set in M_1 = $\{q_2\}$
Accept state set in M_2 = $\{q_1, q_4\}$

(c) On input sequence 'aabb' sequence of states :-

M_1 : $q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_1 \xrightarrow{b} q_1$

M_2 : $q_1 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_4$

(d) Machine M_2 will accept the string while M_1 do not.

(e) When input is empty word ε, the machine just enters the start state & stays there. Hence, we have to answer whether start state is accept state or not.
M_1: Rejects while M_2: Accept.

Sol
(2) A DFA machine is formally 5-tuple
$(Q, \Sigma, \delta, q_0, F)$, where;
$Q \rightarrow$ finite set of states
$\Sigma \rightarrow$ finite alphabet
$\delta \rightarrow$ transition funcⁿ i.e $Q \times \Sigma \rightarrow Q$.
$q_0 \rightarrow$ start state, $q_0 \in Q$
$F \rightarrow F \subseteq Q$, set of accept states.

for M_1,

$Q = \{q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$

δ :

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

start state is q_1.
$F = \{q_2\}$

for M_2,

$Q = \{q_1, q_2, q_2, q_4\}$
$\Sigma = \{a, b\}$, $F = \{q_1, q_4\}$

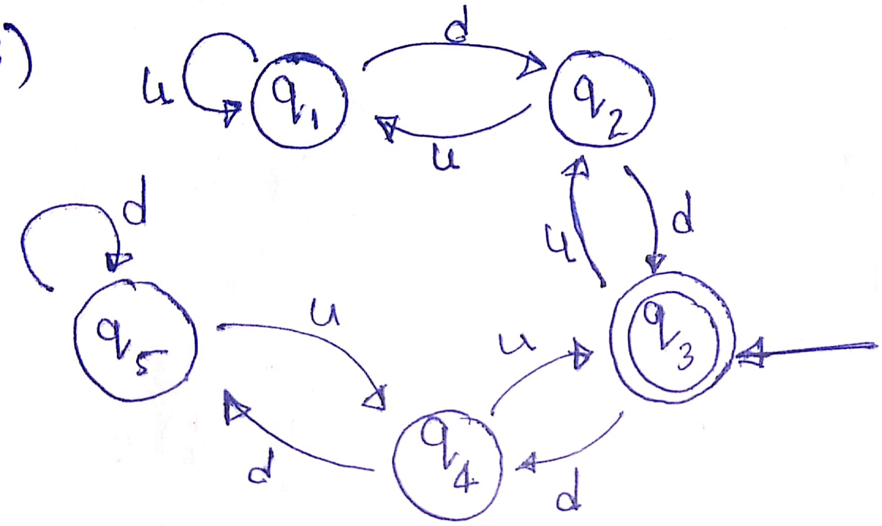
δ :

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

start state is q_1.

Scanned with CamScanner

Sol
(1.3)



State diagram for DFA machine M given in the Question.

Q.4 = Each of the following language is intersection of two simple languages. In each part construct DFAs for the simpler languages & then combine them. In all part $\Sigma = \{a, b\}$.

(a) $\{w \mid w$ has atleast 3 a's & atleast 2 b's$\}$

(b) $\{w \mid w$ has exactly 2 a's & atleast 2 b's$\}$

(c) $\{w \mid w$ has even number of a's and one or two b's$\}$.

Sol

(04) Formally, DFA for intersection of language which gets recognised by DFAs, $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ is written as

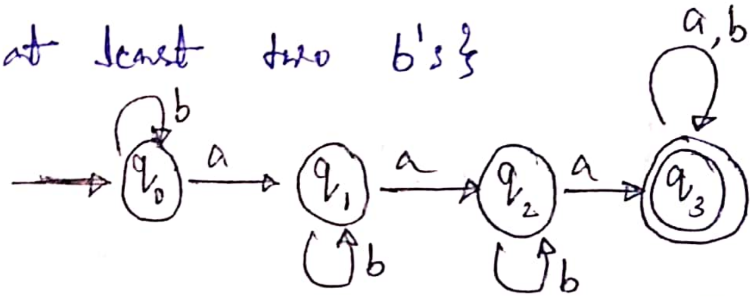
$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

where $\delta(q', q'') = (\delta_1(q'), \delta_2(q''))$

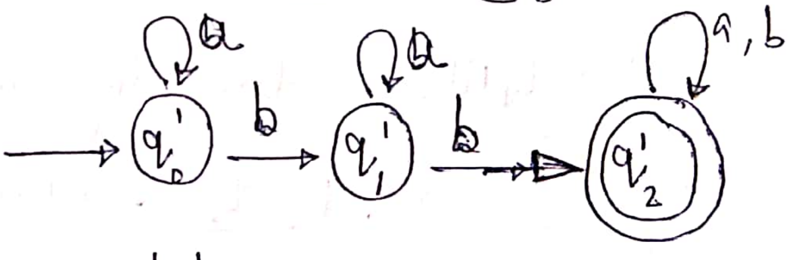
(a) $L_1 = \{ w \mid w$ has at least 3 a's $\}$

$L_2 = \{ w \mid w$ has at least two b's $\}$

L_1 state diagram :

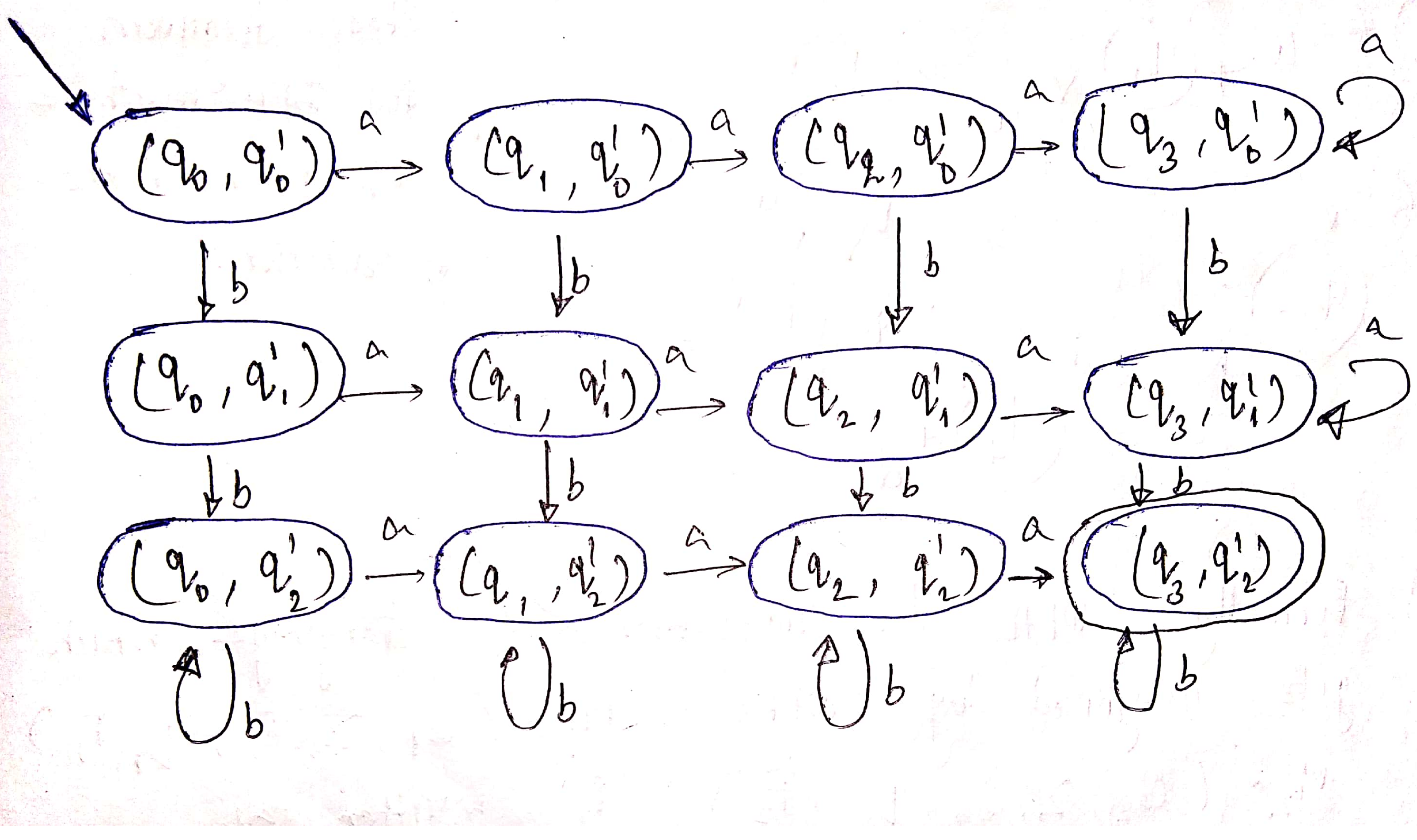


L_2 state diagram :



So, for $L = L_1 \cap L_2$, state diagram ;

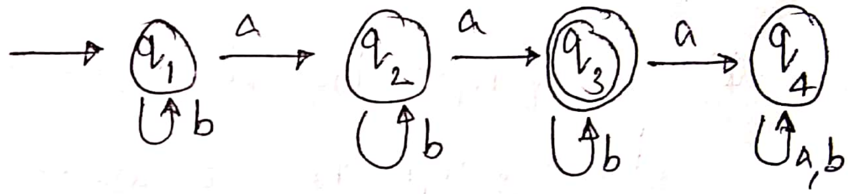
i.e. $L = \{ w \mid w$ has at least 3 a's and at least 2 b's $\}$


(q_0, q'_0)
(q_1, q'_0)
(q_2, q'_0)
(q_3, q'_0)
(q_0, q'_1)
(q_1, q'_1)
(q_2, q'_1)
(q_3, q'_1)
(q_0, q'_2)
(q_1, q'_2)
(q_2, q'_2)
(q_3, q'_2)
a
b

(b) $L_1 = \{ w \mid w$ has exactly 2 a's $\}$

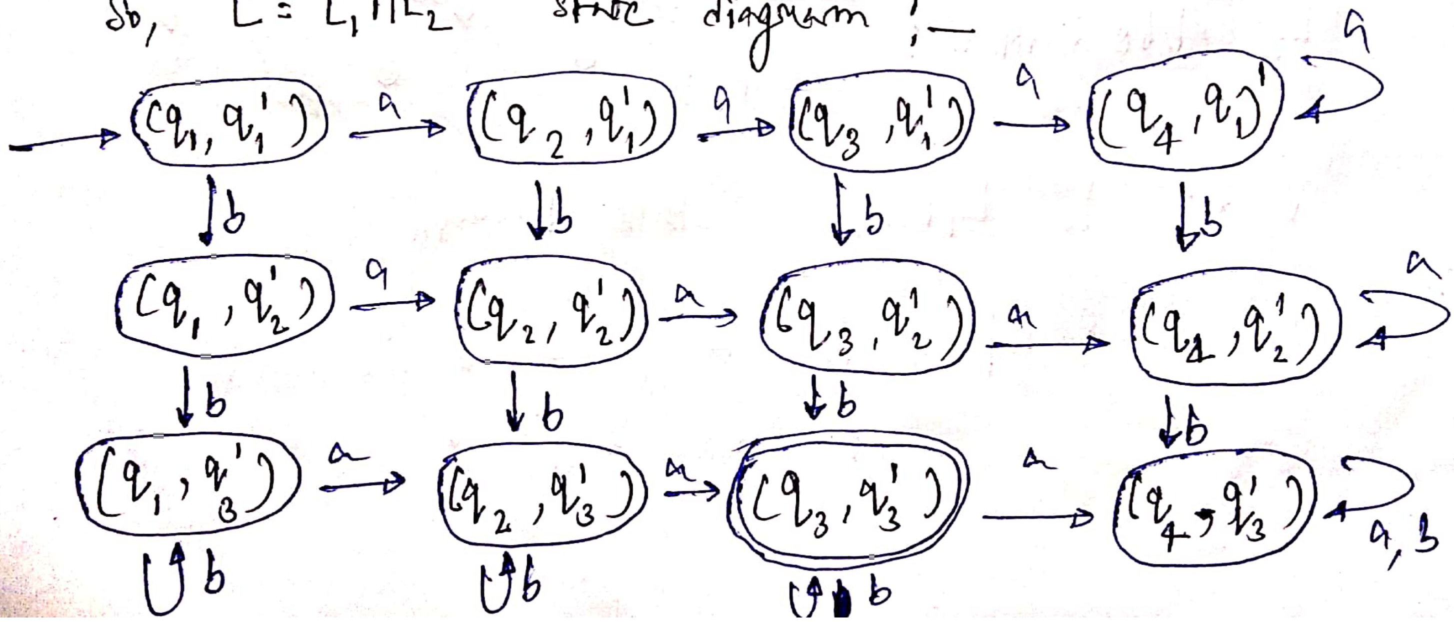
$L_2 = \{ w \mid w$ has at least two b's $\}$

state diagram L_1 :



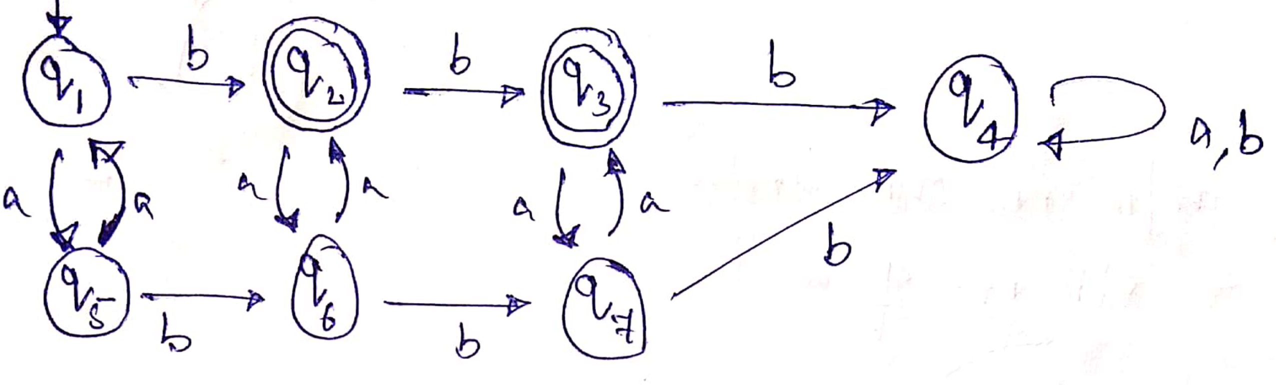
L_2 :

So, $L = L_1 \cap L_2$ state diagram :—



(c) Using the similar analysis ;

state diagram :— $L = \{w \mid w$ has an even number of a's & one or two b's $\}$



Q.5 = Prove that every NFA can be converted to an equivalent one that has a single accept state.

Sol 5
([illegible])

<u>To prove</u>: every NFA can be converted to an equivalent one that has a single accept state.

<u>Approach</u>: Make a new accept state i.e. q' & add ε-transitions from every accept state to this new accept state & remove older accept state.

So, say $M = \{Q, \Sigma, \delta, q_0, F\}$ be any NFA.

We construct required M' as follows;

$$M' = \{Q \cup \{q'\}, \Sigma, \delta', q_0, \{q'\}\}$$

where q' is new accept state & $\delta' = \delta \cup \bigcup_{q \in F} (q, \varepsilon, q')$.

Q.6 = Let $D = \{ w | w$ contains an even number of a's and an odd number of b's and doesnot contain the substring ab $\}$. Given a DFA with 5 states that recognizes D and a regular expression that generates D.

Soln ⇒ We can say D is a language which have strings like $b^k a^m$; k is odd and m is even.

For e.g. ⇒ b, baa, bbbaaaa, bbbaa etc.

So, $D = \{ w | w$ contains odd b's followed by even a's $\}$

So, the DFA will be :- $\langle Q, \Sigma, \delta, Q_0, F \rangle$

Scanned with CamScanner

where

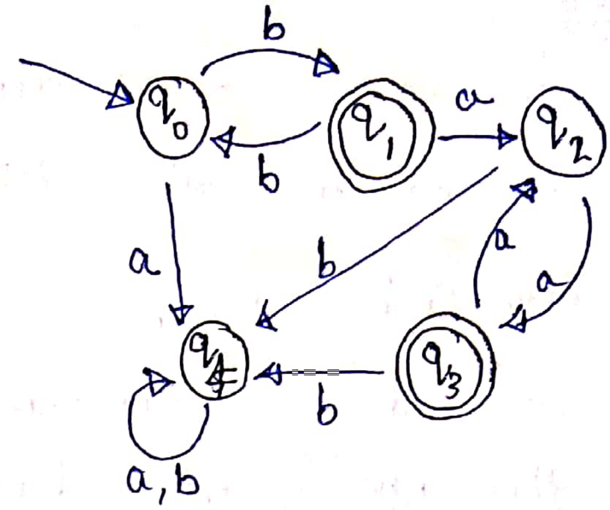
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

$Q_0 = \{q_0\}$

$F = \{q_1, q_3\}$

δ :



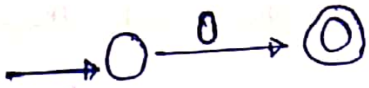
	q_0	q_1	q_2	q_3	q_4
a	q_4	q_2	q_3	q_2	q_4
b	q_1	q_0	q_4	q_4	q_4

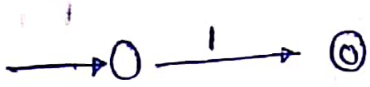
Now, we can say $D = \alpha \circ \beta$; 'o' is concatenation operator,
$\alpha = \{w \mid w$ has odd b's$\}$ and $\beta = \{w \mid w$ has even a's$\}$
Let R_D, R_α & R_β be reg. exp. generation D, α & β then,
$R_D = R_\alpha \circ R_\beta$
And as $R_\alpha = b(bb)^*$, $R_\beta = (aa)^*$
So, $R_D = b(bb)^* \circ (aa)^*$ $\Rightarrow$ Reg. exp. for $D = b(bb)^*(aa)^*$

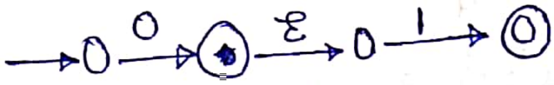
Q.7 = (a) Given an NFA that recognizes language $(01 \cup 001 \cup 010)^*$.
(b) Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

Sol^n $\Rightarrow$ (a) We will exploit the property of regular language that they are closed under union, concatenation and star operation.

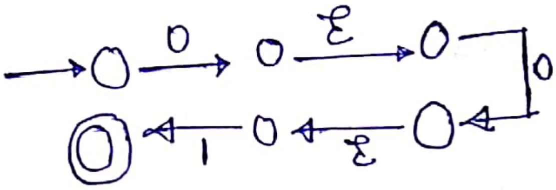
Reg. Exp. = $(01 \cup 001 \cup 010)^*$

NFA for accepting '0' ⇒ 

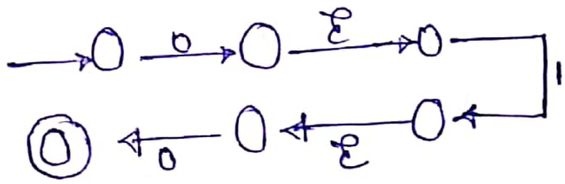
" " " '1' ⇒ 

✓ " " " '01' ⇒ 

{ using closure concatenation property.

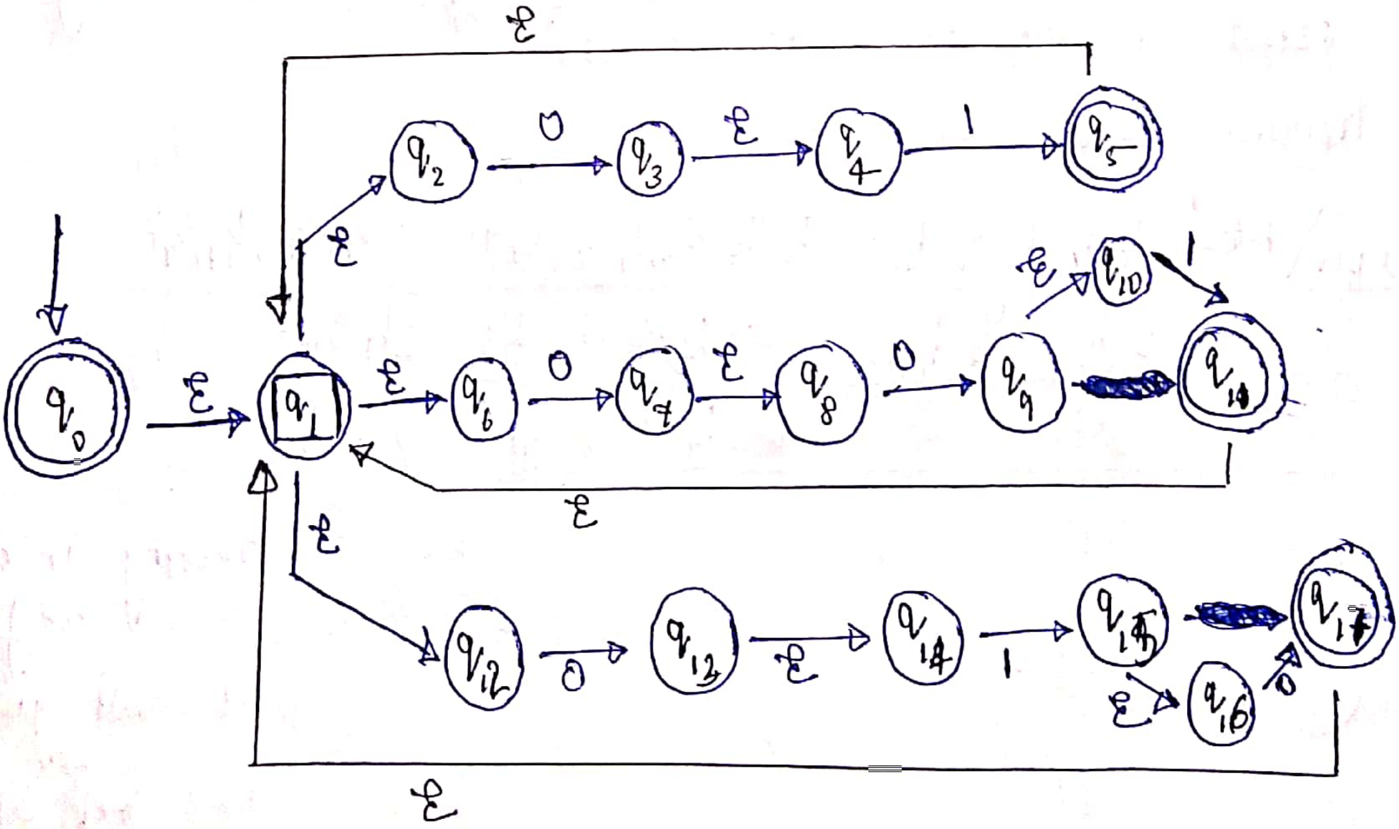
✓ " " " '001' ⇒ 

{ "

✓ " " " '010' ⇒ 

{ "

Now, using closure under union property & combining '✓' and then joining acceptable state with '□' for accounting '*'.



(b) Transition table for the obtained NFA :-

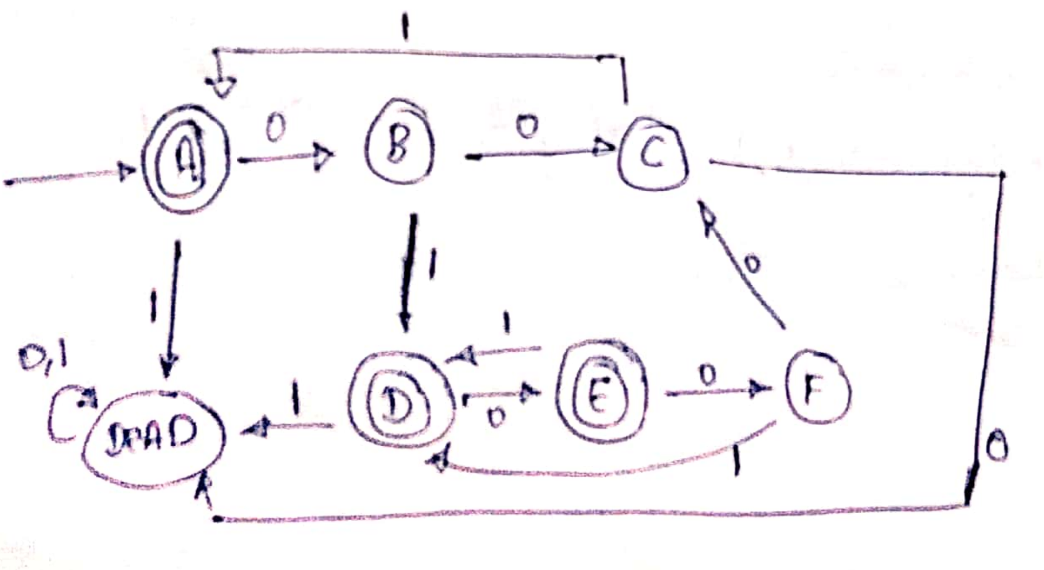
State	ε	0	1
→ q_0	q_1	×	×
q_1	$\{q_2, q_6, q_{12}\}$	×	×
q_2	×	q_3	×
q_3	q_4	×	×
q_4	×	×	q_5
q_5	q_1	×	×
q_6	×	q_7	×
q_7	q_8	×	×
q_8	×	q_9	×

State	ε	0	1
q_9	q_{10}	×	×
q_{10}	×	×	q_{11}
q_{11}	q_1	×	×
q_{12}	×	q_{13}	×
q_{13}	q_{14}	×	×
q_{14}	×	×	q_{15}
q_{15}	q_{16}	×	×
q_{16}	×	q_{17}	×
q_{17}	q_1	×	×

For DFA, ε closure$(q_0) = \{q_2, q_6, q_{12}, q_0, q_1\}$, Acceptable state are all state combination of q_i's of NFA having atleast 1 acceptable state of NFA.

Transition table for DFA :-

Input\State	$\{q_0, q_1, q_2, q_6, q_{12}\}$	$\{q_3, q_7, q_{13}, q_4, q_8, q_{14}\}$	$\{q_4, q_8, q_{14}, q_9\}$
0	$\{q_3, q_7, q_{13}, q_4, q_8, q_{14}\}$	$\cancel{\phi}\{q_4, q_8, q_{14}, q_9\}$	$\{q_9, q_{10}\}$
1	$\cancel{\phi}$	$\cancel{\phi}\{q_4, q_5, q_8, q_{14}, q_{15}\}$	$\{q_5, q_{10}, q_{15}\}$



Creating in this way we will get our final DFA be ⇐

Q.8 = Let B be any language over the alphabet Σ. Prove that $B = B^+$ iff $BB \subseteq B$.

Sol^n = Given :- B is any language over alphabet Σ.

To prove :- (a) If $B = B^+$ then $BB \subseteq B$

(b) If $BB \subseteq B$ then $B = B^+$

Proof :- (a) Let $B = B^+$. And as for every language B, we can claim $BB \subseteq B^+$.

Therefore, $BB \subseteq B$. Hence proved (a).

(b) Let $BB \subseteq B$. And we know for every language B, $B \subseteq B^+$. We have to only show $B^+ \subseteq B$.

Let $w \in B^+$, then $w = m_1 m_2 \cdots m_k$; $m_i \in B$ & $k \geq 1$.

Since, $m_1, m_2 \in B$ & $BB \subseteq B \Rightarrow x_1 x_2 \in B$.

Similarly, as $m_1 m_2 \in B$ & $m_3 \in B$ & $BB \subseteq B \Rightarrow m_1 m_2 m_3 \in B$

Continuing this way, $m_1 m_2 \cdots m_k \in B \Rightarrow w \in B \Rightarrow B^+ \subseteq B$.

Hence, as $B \subseteq B^+$ & $B^+ \subseteq B \Rightarrow B^+ = B$.

Therefore, proved (b)

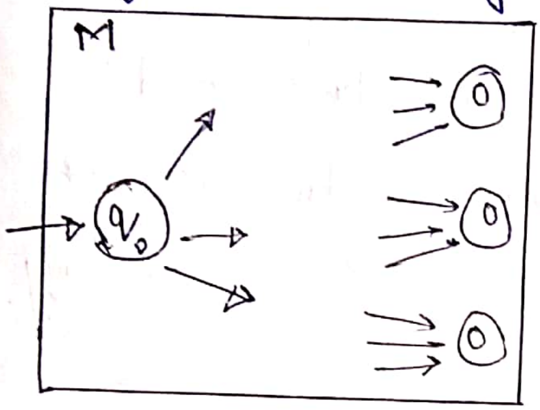
Hence proved $B = B^+$ iff $BB \subseteq B$.

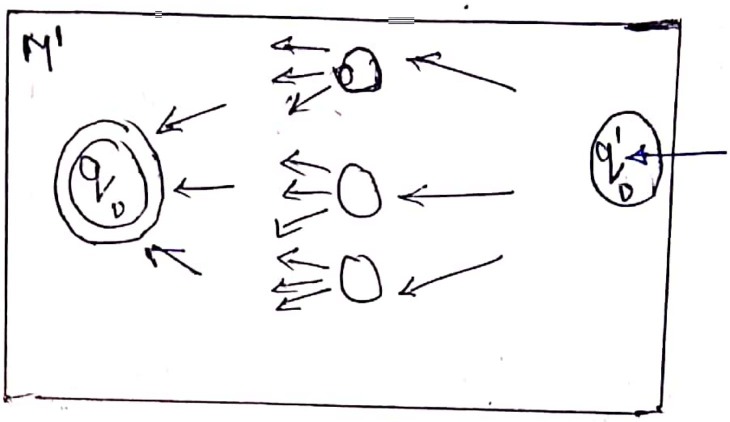
Q.9 = For any string $w = w_1 w_2 \ldots w_n$, the reverse of w written w^R, is the string w in reverse, $w_n \ldots w_2 w_1$. For any language A, let $A = \{ w^R \mid w \in A \}$. Show if A is regular, so is A^R.

Sol^n $\Rightarrow$

Let DFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ such that $L(M) = A$.

We will build NFA M' s.t. $L(M') = A^R$ by reversing arrow of A, converting start state of M to be only accept state & adding a new state q'_{accept} for M' & from q'_0 add ϵ-transitions to each state of M' corresponding to accept states of M.





Here $q'_0 = q'_{accept}$. For any $w \in \Sigma$, E path from start state to accept in M iff E path in w^R from q'_0 to q'_{accept} in M'. Thus, $w \in A$ iff $w^R \in A^R$.

8.10 = Prove that two expression are equal i.e.

$$a^*a(ba^*a)^* = (a+ab)^*a.$$

Sol^n $\Rightarrow$ $(a+ab)^*a = [a^*((ab)a^*)^*]a$ $\quad \{(R+S)^* = R^*(SR^*)^*\}$

$= a^*[((ab)a^*)^*a]$ $\quad \{\cdot$ is associative$\}$

$= a^*[(a(ba^*))^*a]$ $\quad \{(RS)^*R = R(SR)^*\}$

$= a^*[a((ba^*)a)^*]$ $\quad \{\cdot$ is associative$\}$

$= a^*a(ba^*a)^*$ Hence proved.

Scanned with CamScanner

Q.11 = Let $B_n = \{ a^K \mid K$ is a multiple of $n \}$. Show that for each $n \geq 1$, the language B_n is regular.

Sol^n = A set is always regular if a DFA/NFA can be drawn for it.

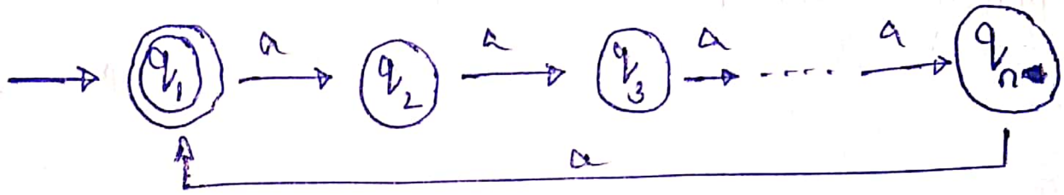
In case $n=1 \rightarrow B_1 = \{ a, a^2, a^3, a^4, \ldots \infty \}$

We can make DFA for B_1 s.t. $L(M) = B_1$ then

M =

. Hence, B_1 is regular.

Now for general n, DFA can also be made s.t. $L(M_n) = B_n$ then M_n =



q_1 is initial & final state.

Therefore, B_n is regular for all $n \geq 1$.

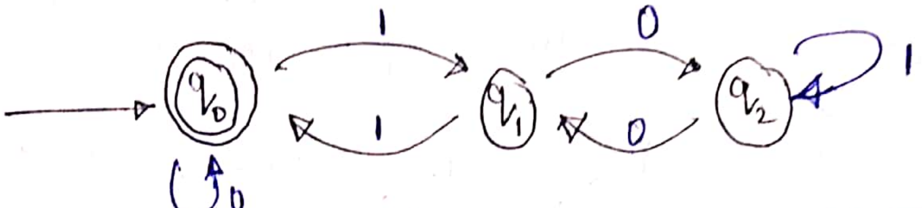
Q.12 = Let $C_n = \{ x \mid x$ is binary no. that is multiple of $n \}$ show that for each $n \geq 1$, the language C_n is regular.

Sol^n = A set is always regular if a DFA/NFA can be drawn for it. Assume $n=3$.

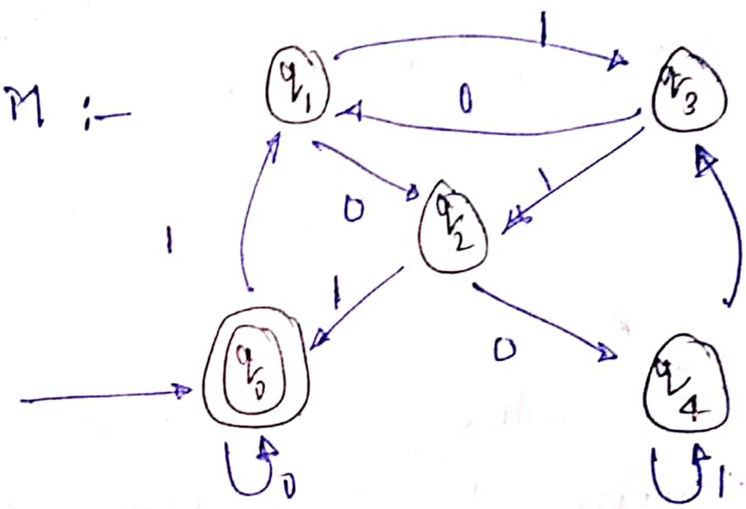
To check whether x, $x \in C_3$ is multiple of 3 or no, we will find $x \% 3$. If 0 then yes, otherwise no.

Everytime when we read a digit, the preceeding string is shifted left 1 position thereby doubling its value x.
If current digit is 0, new value $= 2x \pmod 3$
otherwise $2x+1 \pmod 3$.

Thus, DFA M s.t. $L(M) = C_3$:—



Similarly for $n=5$, M :—



Also, the other cases can be proved similarly. Like
$C_6 = C_3 0$, $C_{12} = C_6 00$, $C_{15} = C_3 \cap C_5$ ......

So, $M = (\{q_0, q_1, \ldots q_n\}, \{0,1\}, \delta, q_0, \{q_0\})$

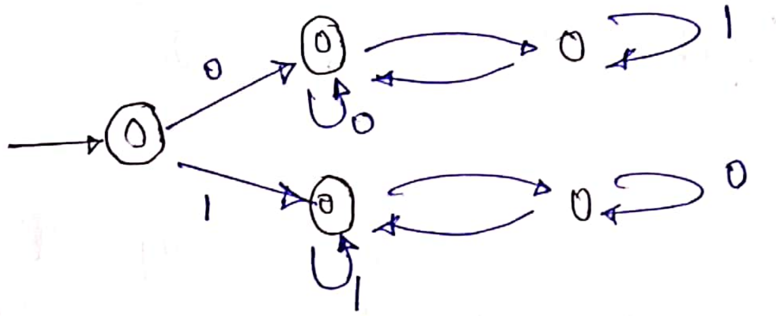
Thus, $L(M) = C_n$. Hence proved that C_n is regular $\forall\ n \geq 1$.

Q.13 = Let $\Sigma = \{0,1\}$ & Let $D = \{ w \mid w$ contains an equal no. of occurrences of the substrings 01 & 10$\}$.

thus $101 \in D$ because 1010 contains one 10 & one 01. show that D is regular.

Sol^n ⇒ We have to state M i.e. DFA s.t. $L(M) = D$.

We can draw M :—



this DFA recognizes D. Hence D is regular.

Q.14 = Prove that no FST can output w^R for every input w if the input and output alphabets are $\{0,1\}$.

Sol^n ⇒ Let $w = x_1 x_2 \cdots x_n \longrightarrow w^R = x_n \cdots x_2 x_1$

To prove :— no FST (Finite state Transducer) can output w^R $\forall$ w, over $\{0,1\}$.

Proof :— (Proof by contradiction) Let us assum FST T, output w^R on w as input.

Consider two input strings 00 & 01.

a) If $w = 00 \longrightarrow w^R = 00$ And If

$w = 01 \longrightarrow w^R = 10$.

In both w^R first input bit is same i.e. 0 but first output bit differs i.e 0 & 1 resp. But from actual definition of FST, FST can output its first output before it reads second Input. thus, FST cannot produce $w^R = 10$ for $w = 01$. So, our assumption that FST T outputs w^R on input w is wrong.

Hence, no FST outputs w^R on input w.

Q.15 = Consider a DFA over $\Sigma = \{a, b\}$ accepting all strings which have number of a's divisible by 6 & number of b's divisible by 8. What is the minimum states that DFA posses?

Soln ⇒ We construct a DFA for strings divisible by 6. It requires minimum 6 states a length of string mod 6 = 0,1,2,3,4,5. Also, we do this for string divisible by 8, which require minimum 8 states as length of string mod 8 = 0,1,2,3,4,5,6,7.

If first DFA is minimum & second DFA is minimum then merging both will be having minimum result DFA. Such DFA are called compound Automata.

So, minimum states = $6 \times 8 = 48$.