TOC.

Language (L) = collection of string

- DFA

$M = (Q, \Sigma, \delta, q_0, F)$ ← Transition fun^n

$\delta : \underset{\text{current state}}{Q} \times \underset{\text{input symbol}}{\Sigma} \rightarrow \underset{\text{next state}}{Q}$

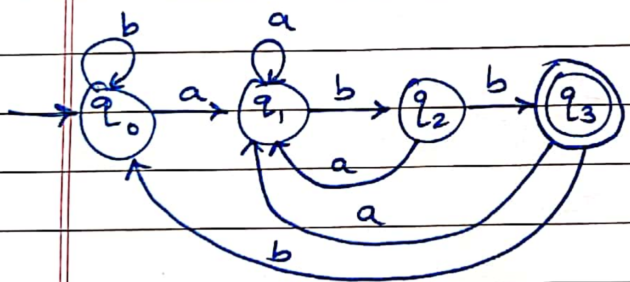
- Type 1

→ ends with

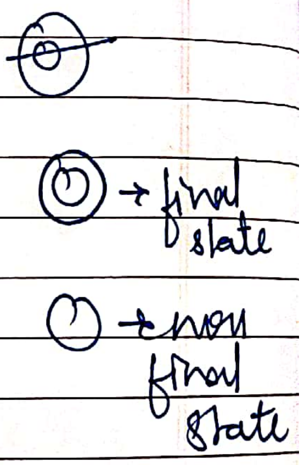
Design DFA over alphabet a, b $\Sigma = \{a, b\}$ which accept the string which ends with abb

$\rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} \underline{\underline{q_3}}$

$L = \{ abb, aabb, babb \; .... \}$



Transition diagram.



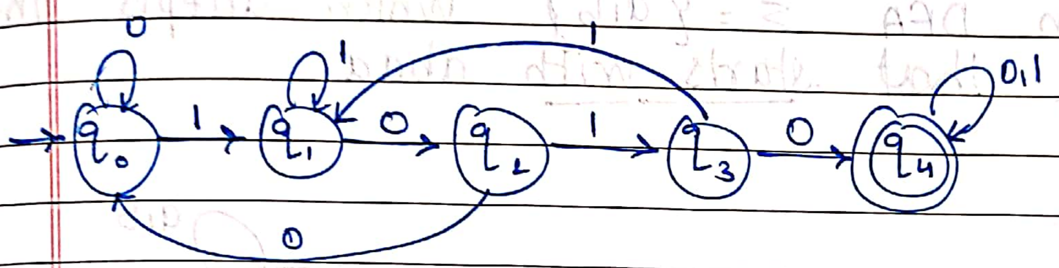
$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_1	q_0

Transition table

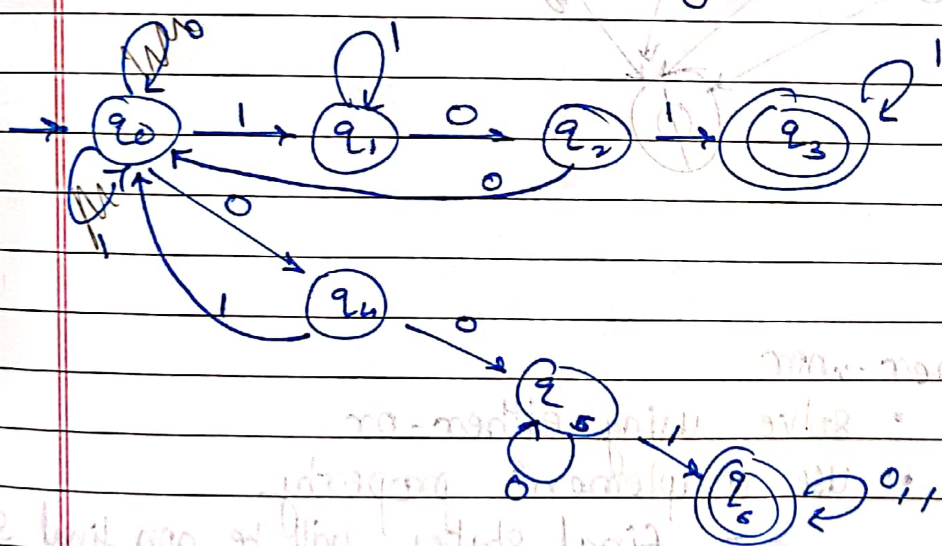
- Type 2
  substring.

substring always gives self loop at the end.

* Design DFA $\Sigma = \{0,1\}$ which accept those string which contains 1010



* Accept string that either contains 101 or 001 as substring.

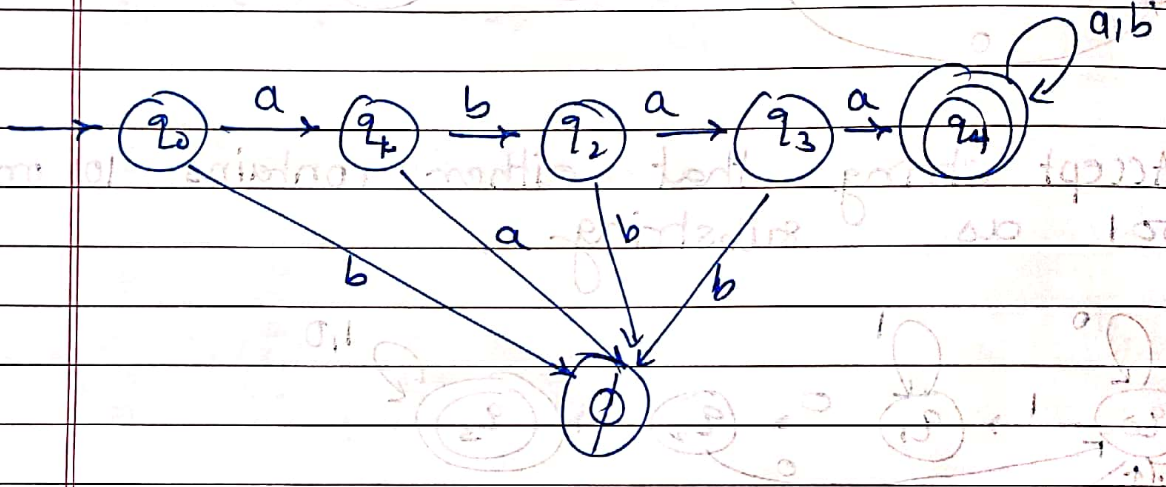


Type 3

→ Starts with.

- add dead state (ϕ)
- add self loop at the end.

* Design DFA Σ = {a, b} which accepts the string that starts with abaa

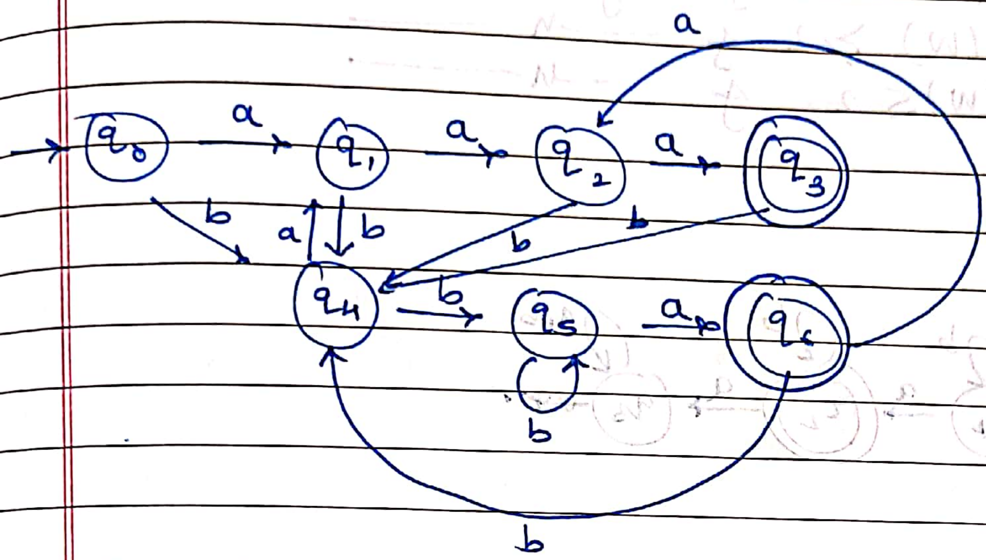


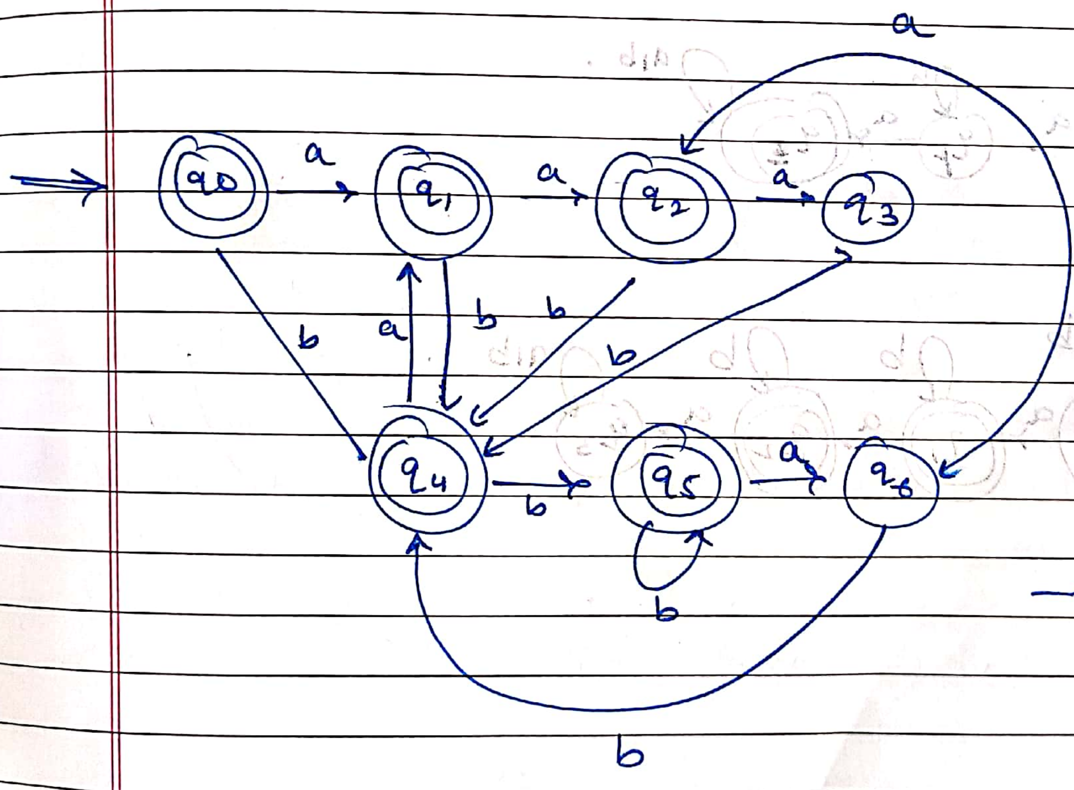
type 4

→ Neither .. nor

- solve using either-or
- use complement property
  - final states will be non final state
  - non — " — final state
  - Do not change transition

* Design DFA to accept string which neither ends with aaa nor bba.



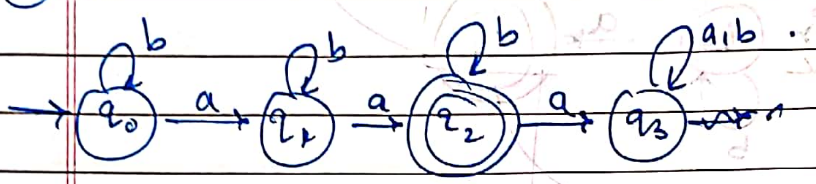


—Ans//

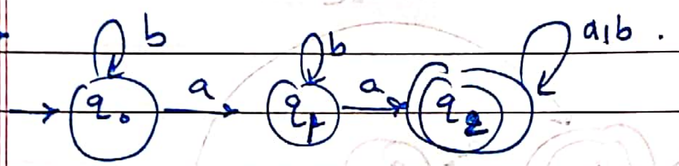
* Design DFA $\Sigma = \{a, b\}$ which accepts

1. $n_a(W) = 2$ & any no. of b
2. $n_a(W) \geq 2$ & — " —
3. $n_a(W) \leq 2$ & — " —

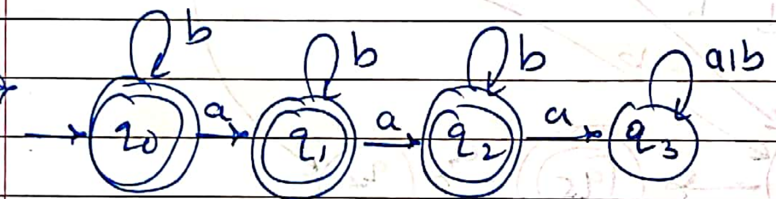
1 →



2 →



3 →



* Design DFA that accepts binary no. $\Sigma = \{0,1\}$ divisible by 3

(OR)

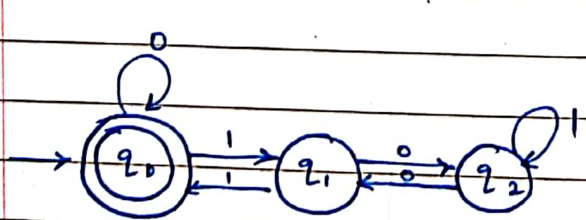
Divisiblity by 3 testor for binary no.

Ans:-

$Q \backslash \Sigma$	0	1
$*00(q_0)$	$\frac{000}{3} = q_0$	$\frac{001}{3} = q_1$
$01(q_1)$	$\frac{010}{3} = q_2$	$\frac{011}{3}$ q_0
$10(q_2)$	$\frac{100}{3}$ q_1	$\frac{101}{3}$ q_2

Transition table

	Binary no.	Remainder ($\div 3$)
0	000	00 $\rightarrow q_0$
1	001	01 $\rightarrow q_1$
2	010	10 $\rightarrow q_2$
3	011	00 $\rightarrow q_0$
4	100	01 $\rightarrow q_1$
5	101	10 $\rightarrow q_2$
6	110	00 $\rightarrow q_0$
7	111	01 $\rightarrow q_1$



Transition diagram.

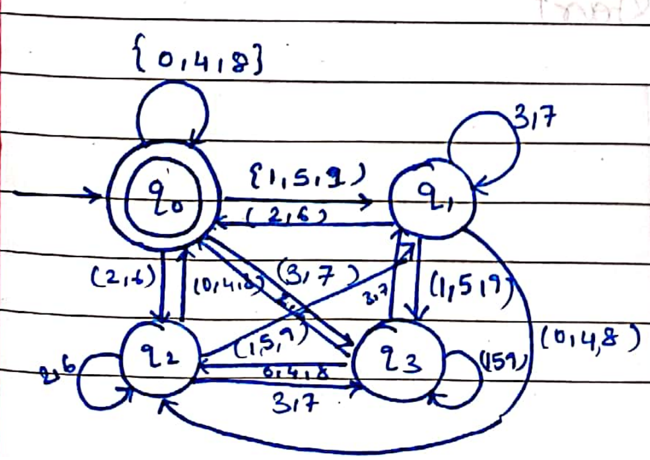
* DFA for decimal no. divisible by 4

Ans:- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Decimal no.	Remainder ($\div 4$)	State
0	0	q_0
1	1	q_1
2	2	q_2
3	3	q_3
4	0	q_0
5	1	q_1
6	2	q_2
7	3	q_3
8	0	q_0
9	1	q_1

- transition table

$q \backslash \Sigma$	q_0 (0,4,8)	q_1 (1,5,9)	q_2 (2,6)	q_3 (3,7)
*0 (q_0)	$\frac{00, 04, 08}{4}$ q_0	q_1	q_2	q_4
1 (q_1)	$\frac{10, 14, 18}{4}$ q_2	q_3	q_0	q_1
2 (q_2)	$\frac{20, 24, 28}{4}$ q_0	q_1	q_2	q_3
3 (q_3)	$\frac{30, 34, 38}{4}$ q_2	q_3	q_0	q_1

- 

* NFA $M = (Q, \Sigma, \delta, q_0, F)$ $\delta : Q \times \Sigma = 2^Q$

It resides in multiple state at the same time

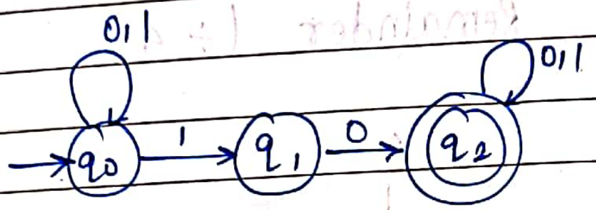
Q = total no. of finite state

Σ = i/p symbol

q_0 = starting state

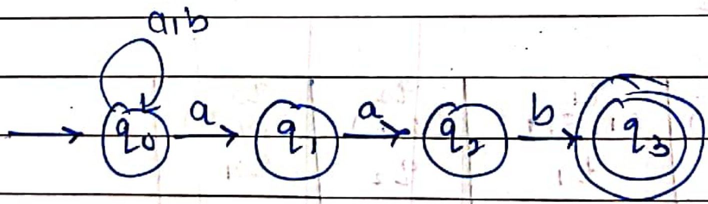
F = set of final states

eg.



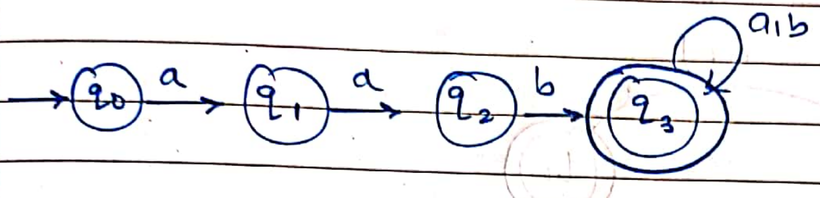
- Type 1 (ends with)
- Type 2 (starts with)
- Type 3 (substring)

(1) ends with aab.



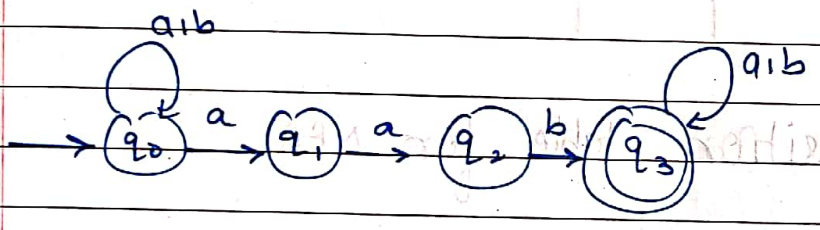
- Add self loop at start

(II) Starts with 'aab'



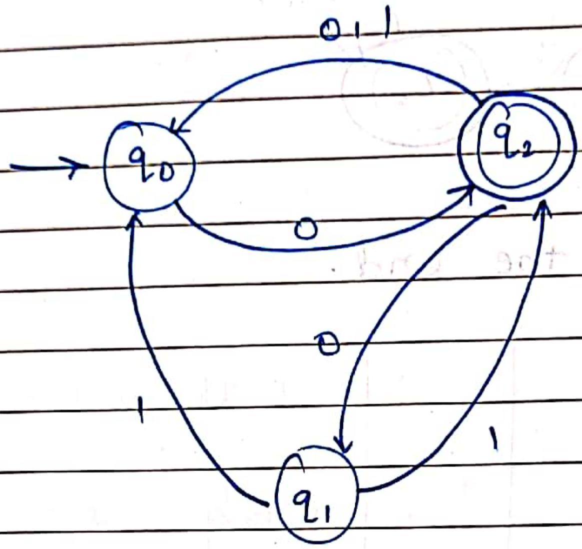
∘ add self loop at the end.

(III) substring 'aab'



∘ add self loop at start & at the end.

* Conversion [NFA to DFA]



Ans Step 1

transition table for NFA

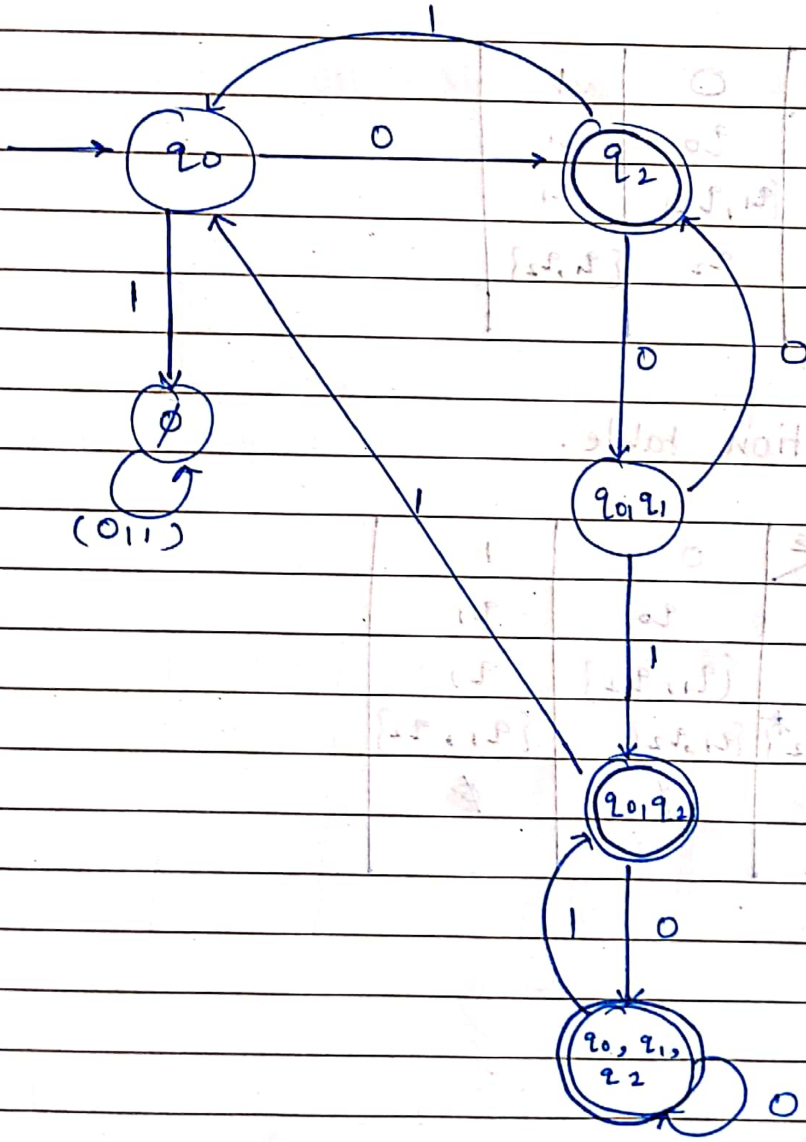
q \ Σ	0	1
→ q0	q2	ϕ
q1	ϕ	q0, q2
q2*	q0, q1	q0

Transition table for DFA.

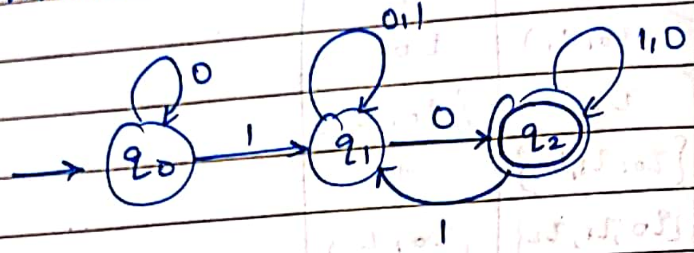
q \ Σ	0	1
q0	q2	ϕ

Date ______
Page No. ______

$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_2	ϕ
q_2^*	$\{q_0, q_1\}$	q_0
$\{q_0, q_1\}$	q_2	$\{q_0, q_2\}$
$\{q_0, q_2\}^*$	$\{q_0, q_1, q_2\}$	q_0
$\{q_0, q_1, q_2\}^*$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
ϕ	ϕ	ϕ



* convert NFA to DFA

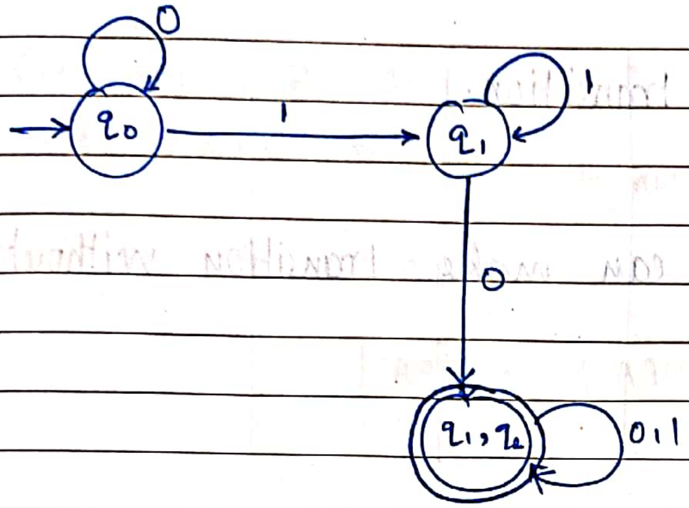


NFA transition table.

$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
q_2^*	q_2	$\{q_1, q_2\}$

- DFA transition table.

$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
$\{q_1, q_2\}^*$	$\{q_1, q_2\}$	$\{q_1, q_2\}$
$\cancel{\phi}$	$\cancel{\phi}$	$\cancel{\phi}$



transition diagram:

FABER-CASTELL Date Page No.

U = union.

* NFA with ε - transition. $M = (Q, \Sigma, \delta, q_0, F)$

ε ↓ null symbol

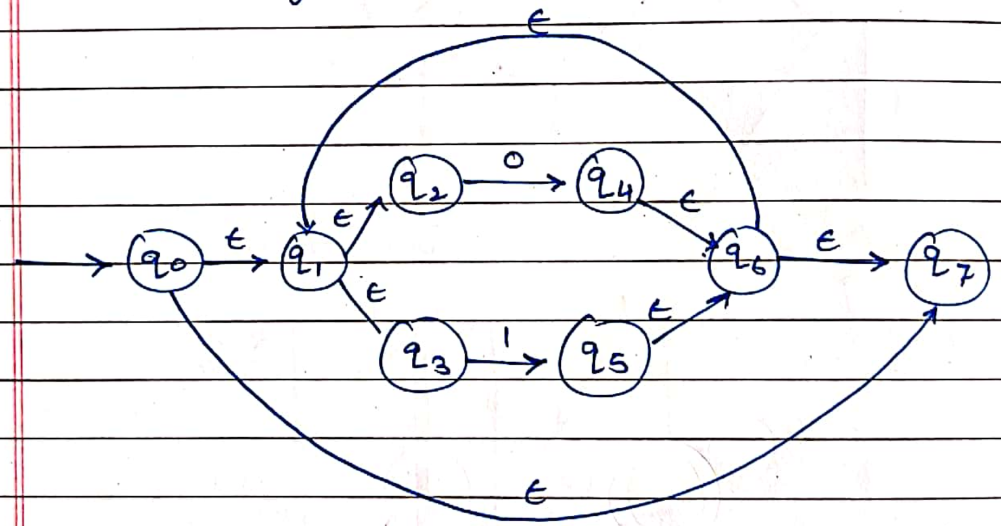
$\delta : Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$

A machine can make transition without input.

→ NFA - ε → NFA → DFA

- ε-closure of q_i

ε closure of q_i includes set of states reachable from q_i on ε moves



ϵ - closure $(q_0) = \{q_0, q_1, q_2, q_3, q_7\}$

$\epsilon^*(q_1) = \{q_1, q_2, q_3\}$
$\epsilon^*(q_2) = \{q_2\}$
$\epsilon^*(q_3) = \{q_3\}$
$\epsilon^*(q_4) = \{q_4, q_6, q_7, q_1, q_2, q_3\}$
$\epsilon^*(q_5) = \{q_5, q_6, q_7, q_1, q_2, q_3\}$
$\epsilon^*(q_6) = \{q_6, q_7, q_1, q_2, q_3\}$
$\epsilon^*(q_7) = \{q_7\}$

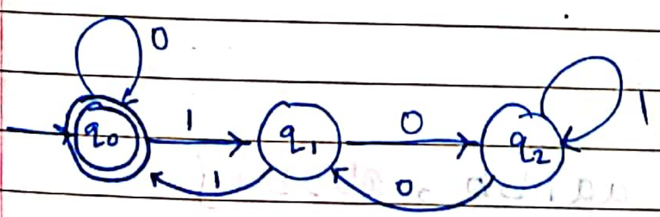
* Construct minimal DFA $\Sigma = \{0,1\}$ which is completly divisible by 3.

Ans:-

Decimal	Binary	Remainder (÷3)	State
0	000	000	q_0
1	001	001	q_1
2	010	010	q_2
3	011	000	q_0
4	100	001	q_1
5	101	010	q_2
6	110	000	q_0
7	111	001	q_1

Q \ Σ	0	1
(00) q_0	q_0	q_1
(01) q_1	q_2	q_0
(10) q_2	q_1	q_2

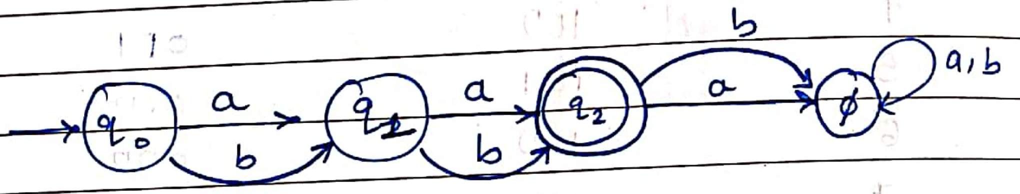
Transition table



* Construct DFA , $\Sigma = \{a, b\}$ such that
(i) length = 2
(ii) length is atleast 2
(iii) length is atmost 2

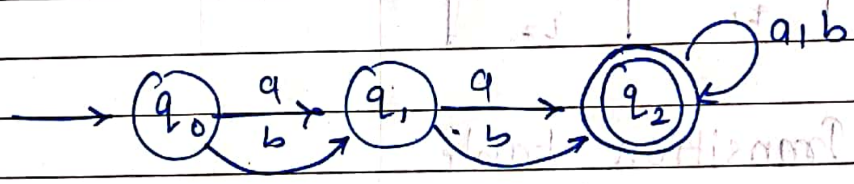
Ans:-

(I) $L = \{aa, ab, bb, ba\}$



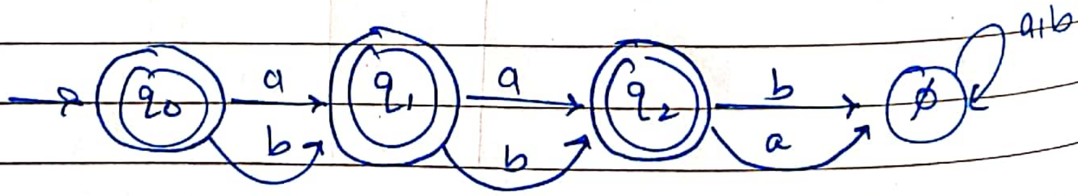
(II) atleast 2.

$L = \{aa, aab, aba \ldots\}$



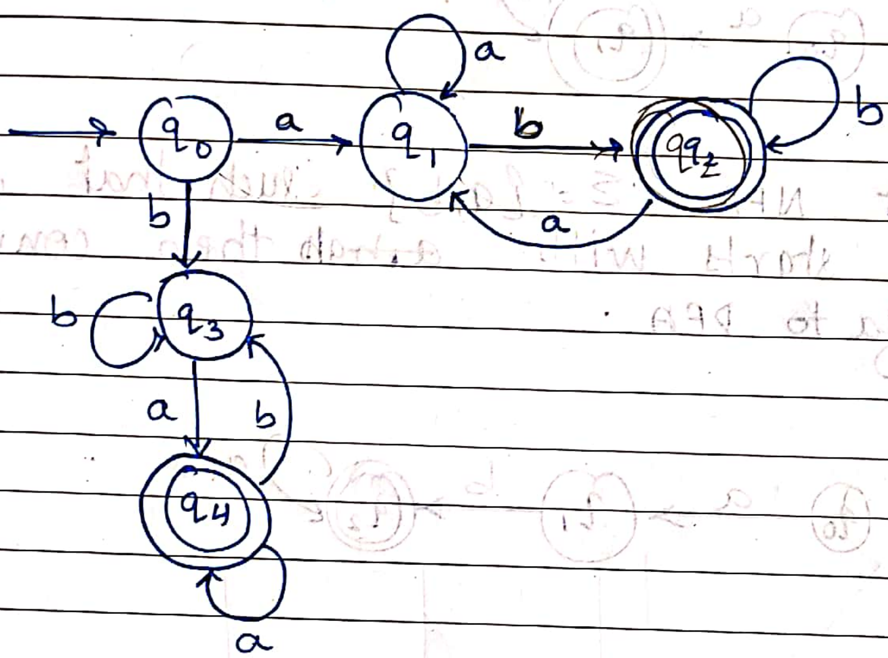
(III) atmost 2.

$L = \{\epsilon, a, b, aa, bb, ab, ba\}$.



* Construct a minimal DFA $\Sigma = \{a, b\}$ which Starts & ends with different symbol.

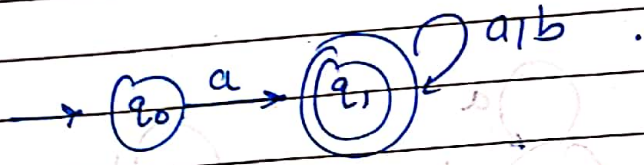
→ $L = \{ab, ba, aab, baa, abab \ldots\}$



NFA

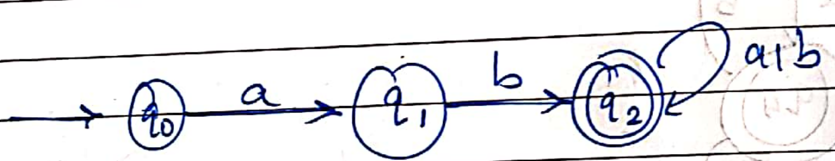
* ~~L=~~ $\Sigma = \{a,b\}$ starts with a.

$L = \{a, aa, ab, aab, aba, \ldots\}$



* Construct NFA $\Sigma = \{a,b\}$ such that all strings starts with ~~a+b~~ ab then convert this nfa to DFA.

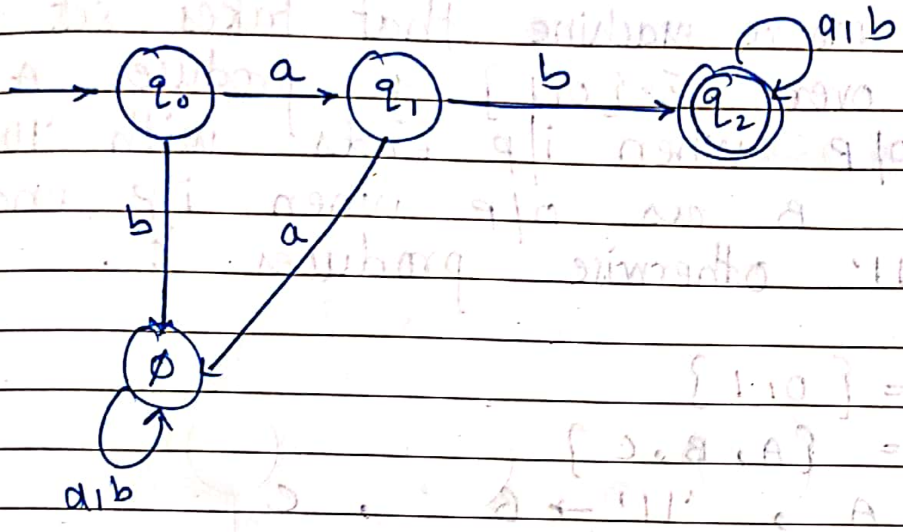
Ans:-



$Q \backslash \Sigma$	a	b
q_0	q_1	ϕ
q_1	ϕ	q_2
q_2	q_2	q_2

NFA Table

$Q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_1	ϕ
q_1	ϕ	q_2
$*q_2$	q_2	q_2
ϕ	ϕ	ϕ


q0
a
q1
b
q2
a,b
b
a
∅
a,b

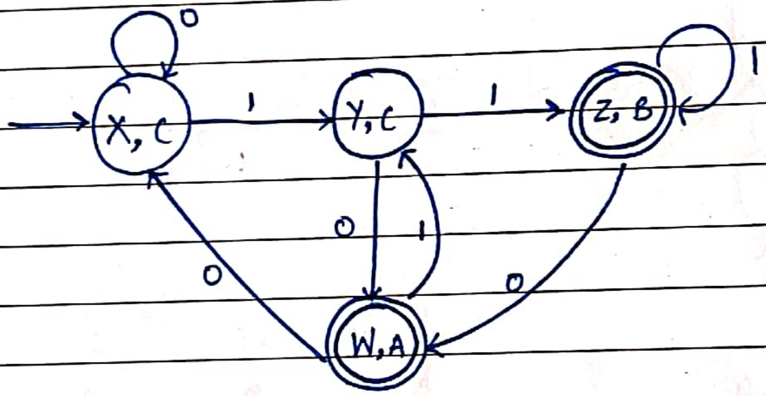
* Construct moore machine that takes set of all strings over $\Sigma = \{0,1\}$ & produces A as o/p when i/p ends with '10' or produces B as o/p when i/p ends with '11' otherwise produces C.

Ans:- $\Sigma = \{0,1\}$

$\Delta = \{A, B, C\}$

'10' → A , '11' → B , C

L = { 10, 11, 010, 111, 011 ... }



$Q \setminus \Sigma$	0	1	Δ
→ X	X	Y	C
Y	W	Z	C
*Z	W	Z	B
*W	X	Y	A

* Construct a moore machine over $\Sigma = \{a, b\}$ such that all strings ending with 'baa' & printing 1 as output otherwise output is 0

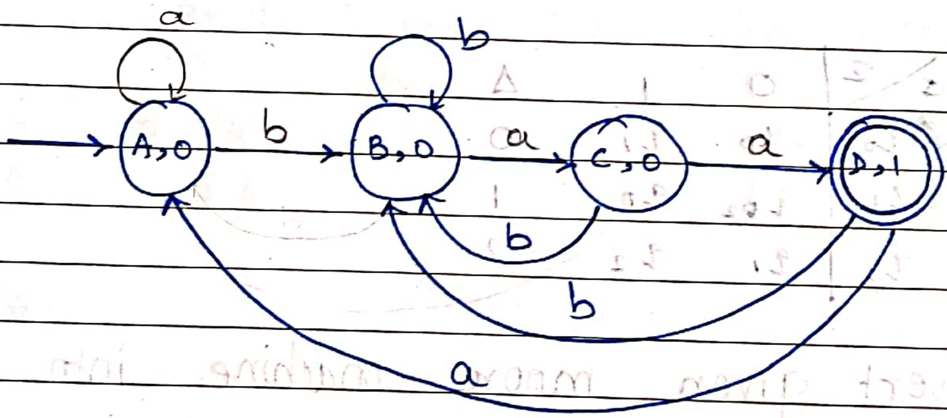
Ans:-

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

$\Delta = \{0, 1\}$

'baa' → 1 , 0 .

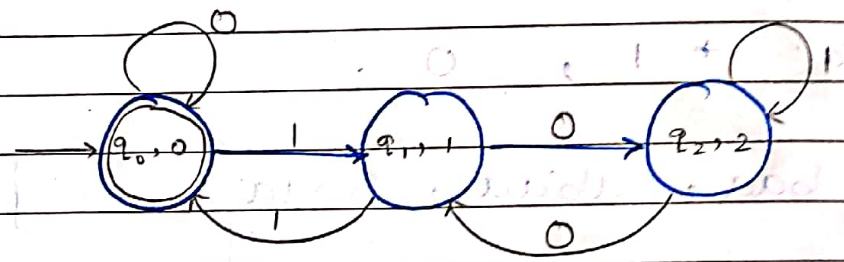
L = { baa, abaa, bbaa ..... }



Q \ Σ	a	b	Δ
→A	A	B	0
B	C	B	0
C	D	B	0
D*	A	B	1

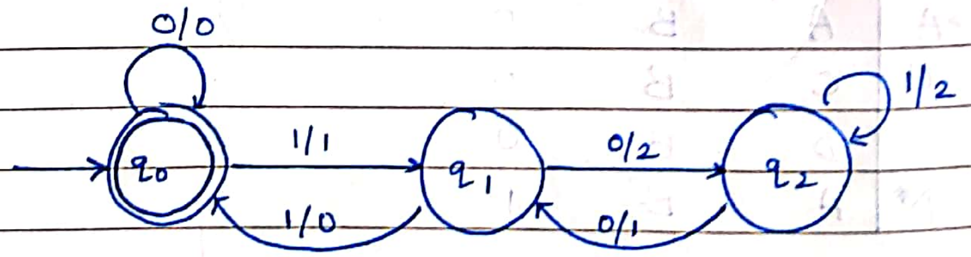
* Construct a moore machine that takes binary no. as i/p & that binary no. is completely divisible by 3.

Ans:- $\Sigma = \{0, 1\}$
$\Delta = \{0, 1, 2\}$



Q \ Σ	0	1	Δ
→* q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

→ convert given moore machine into mealy machine



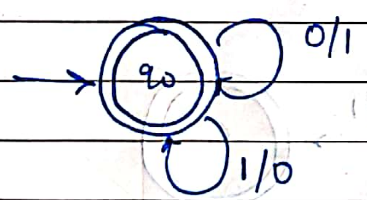
Q \ Σ	0	1
→ q_0	$(q_0, 0)$	$(q_1, 1)$
q_1	$(q_2, 2)$	$(q_0, 0)$
q_2	$(q_1, 1)$	$(q_2, 2)$

* Construct melay machine that takes binary no. as i/p & produces 2's complement of that binary no. as o/p.
Assume the string is read from LSB to MSB & end carry will be discarded.

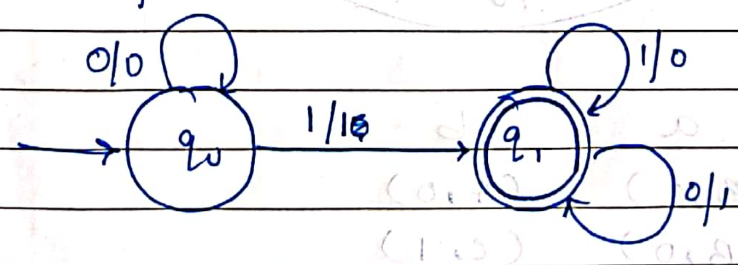
$\Sigma = \{0, 1\}$
$\Delta = \{0, 1\}$

11101100
00010011
1
00010100

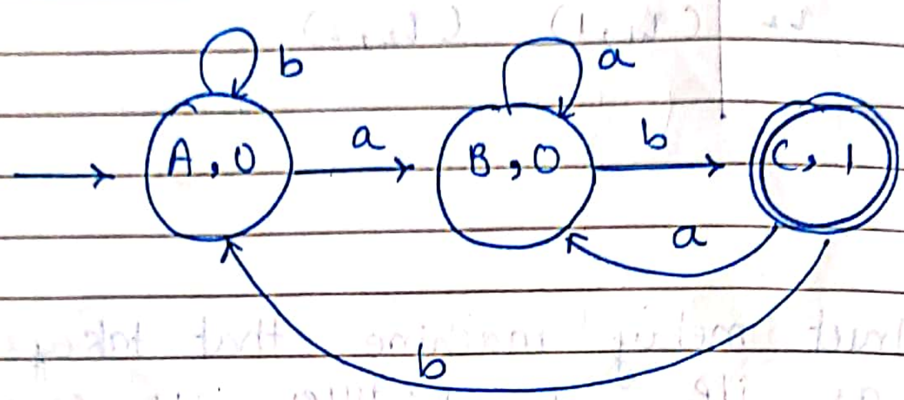
1's complement



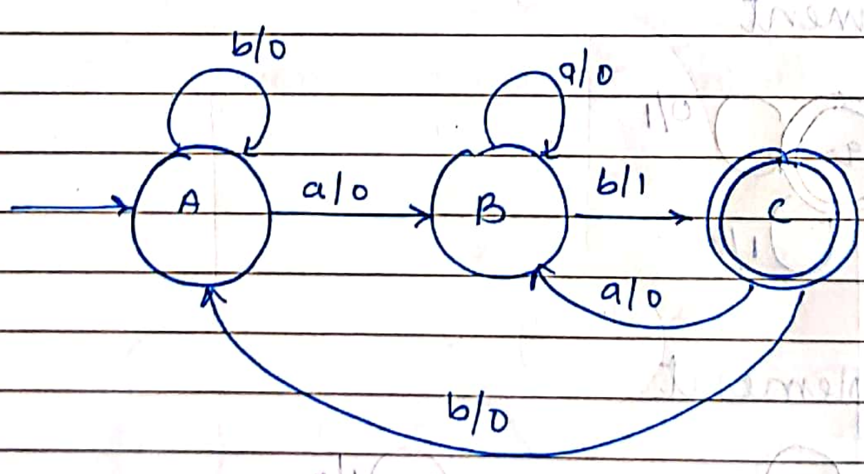
2's complement



* Convert following moore machine into mealy & construct the table as well

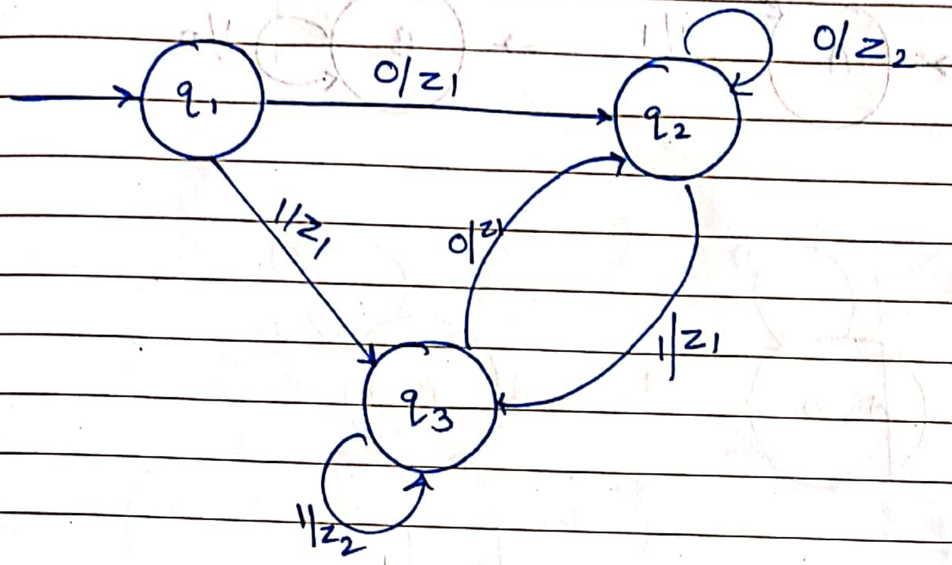


q \ Σ	a	b	Δ
→ A	B	A	0
B	B	C	0
* C	B	A	1

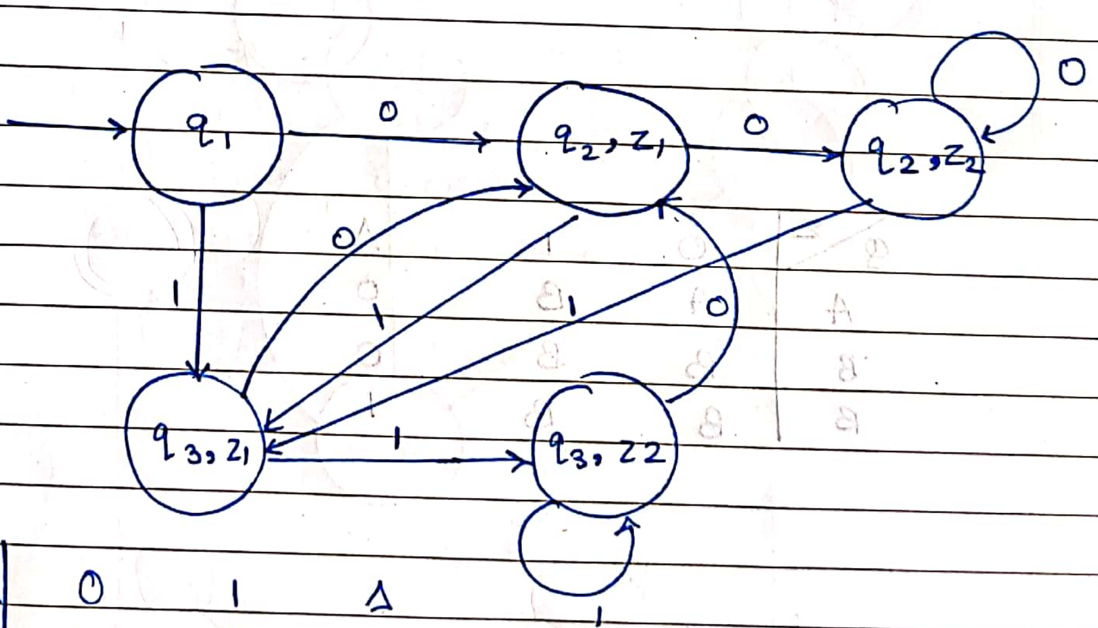


q \ Σ	a	b
→ A	(B, 0)	(A, 0)
B	(B, 0)	(C, 1)
* C	(B, 0)	(A, 0)

* Convert given melady machine to moore machine & also construct graph or transition table for both machines.

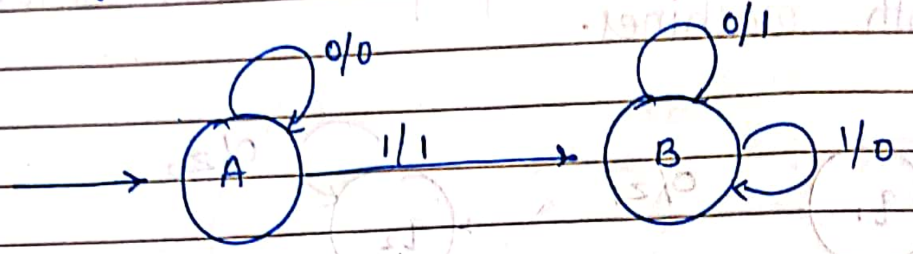


Ans:-

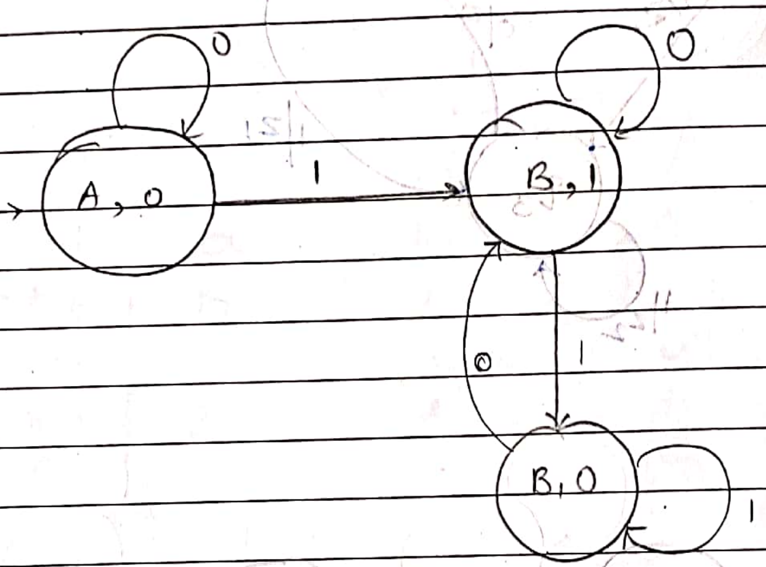


Q \ Σ	0	1	Δ
q_1	q_2	q_3	-
q_2	q_2	q_3	z_1
q_2	q_2	q_3	z_2
q_3	q_2	q_3	z_1
q_3	q_2	q_3	z_2

* Mealy to moore machine.



Ans:-



Q \ Σ	0	1	Δ
A	A	B	0
B	B	B	0
B	B	B	1

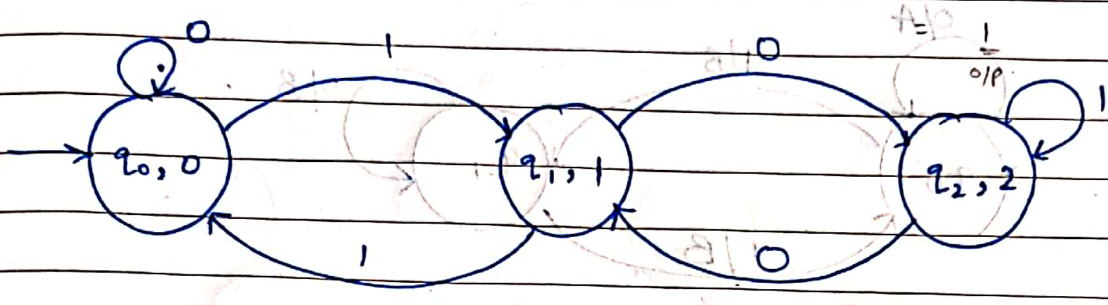
* Moore to mealy.

imp to use formula in exam.

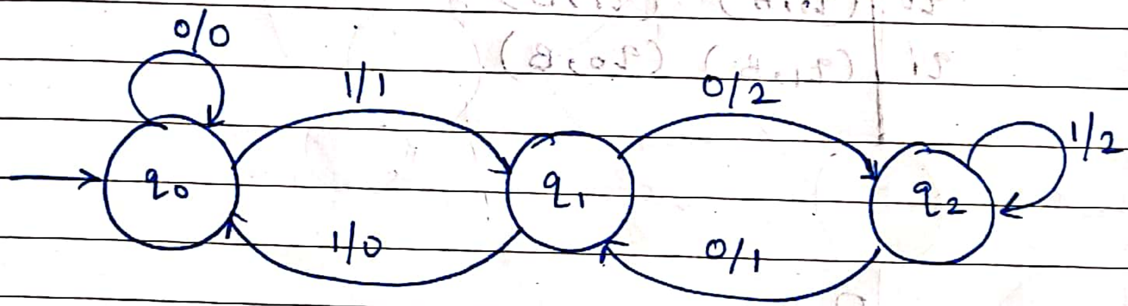
$$\left[\lambda'(q,a) = \lambda(\delta(q,a))\right]$$

$\rightarrow$ eg $\lambda(q_0,1) = \lambda(\delta(q_0,1))$

$= \lambda(q_1)$

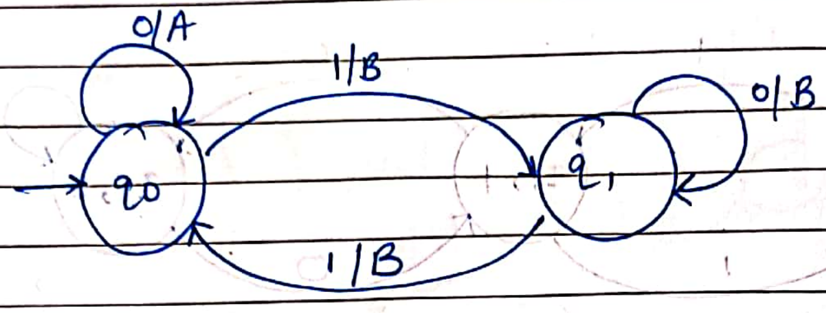


Ans:-

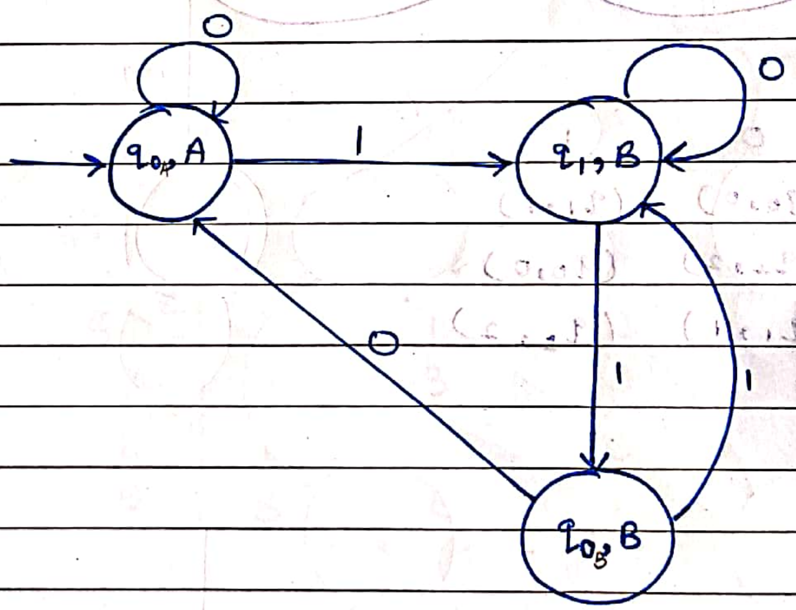


Q \ Σ	0	1
q_0	$(q_0, 0)$	$(q_1, 1)$
q_1	$(q_2, 2)$	$(q_0, 0)$
q_2	$(q_1, 1)$	$(q_2, 2)$

* Mealy to moore.

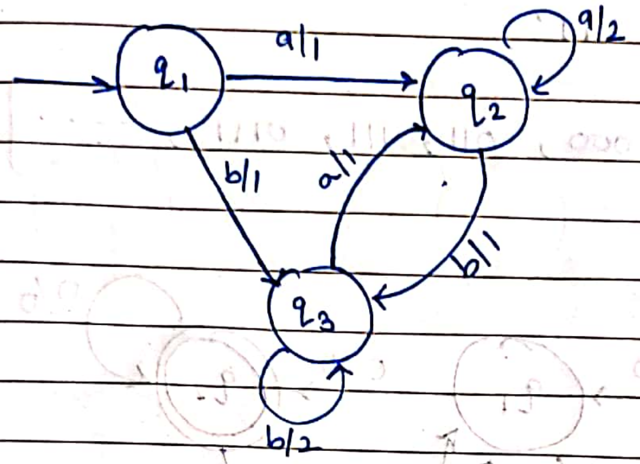


q \ Σ	0	1
q_0	(q_0, A)	(q_1, B)
q_1	(q_1, B)	(q_0, B)

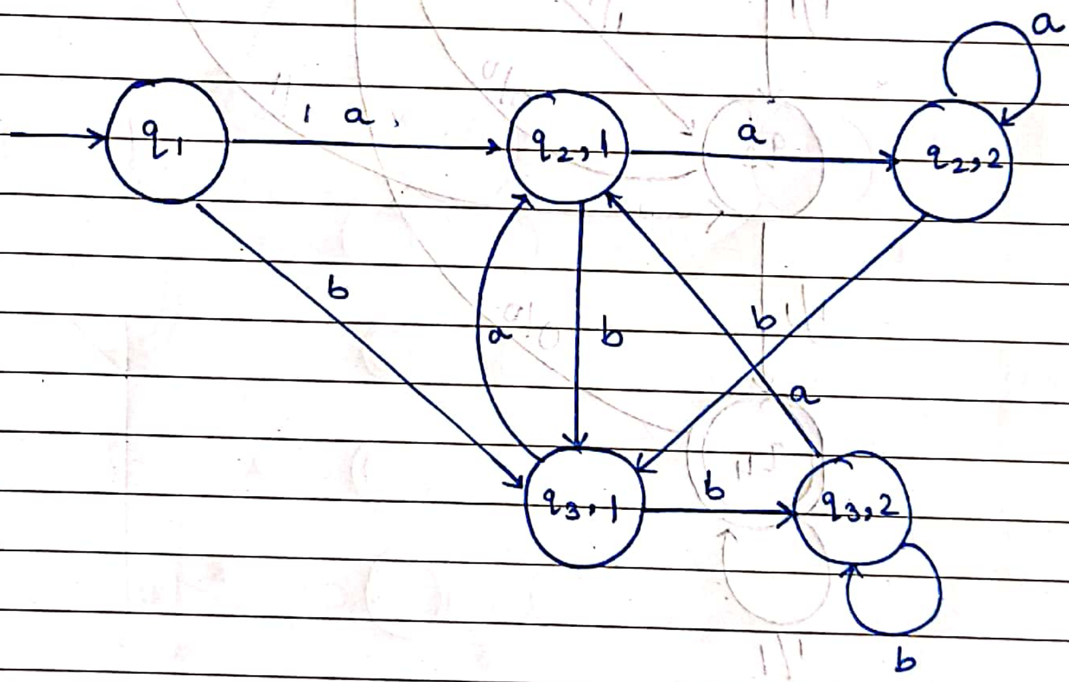


Q \ Σ	0	1	Δ
q_{0A}	q_{0A}	q_1	A
q_{0B}	q_{0A}	q_1	B
q_1	q_1	q_{0B}	B

* convert

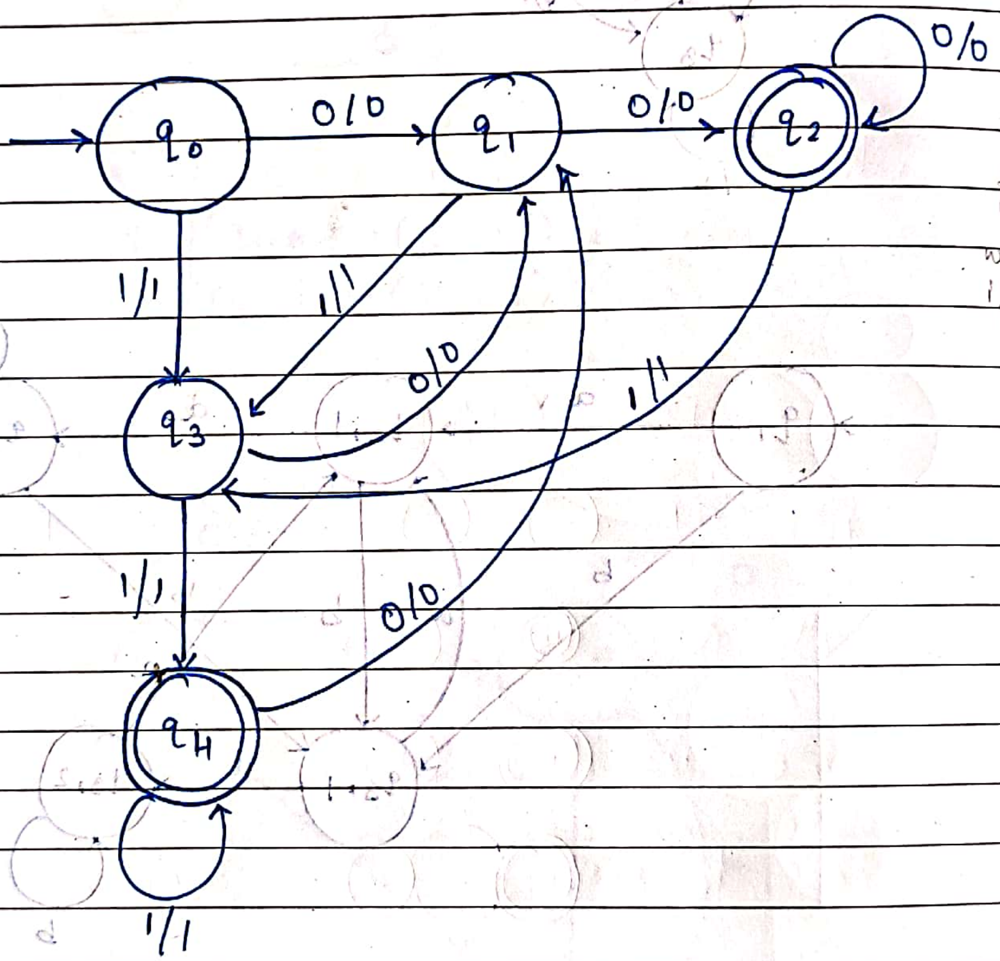


Ans



* Design mealy machine that accepts string ending in '00' or '11'

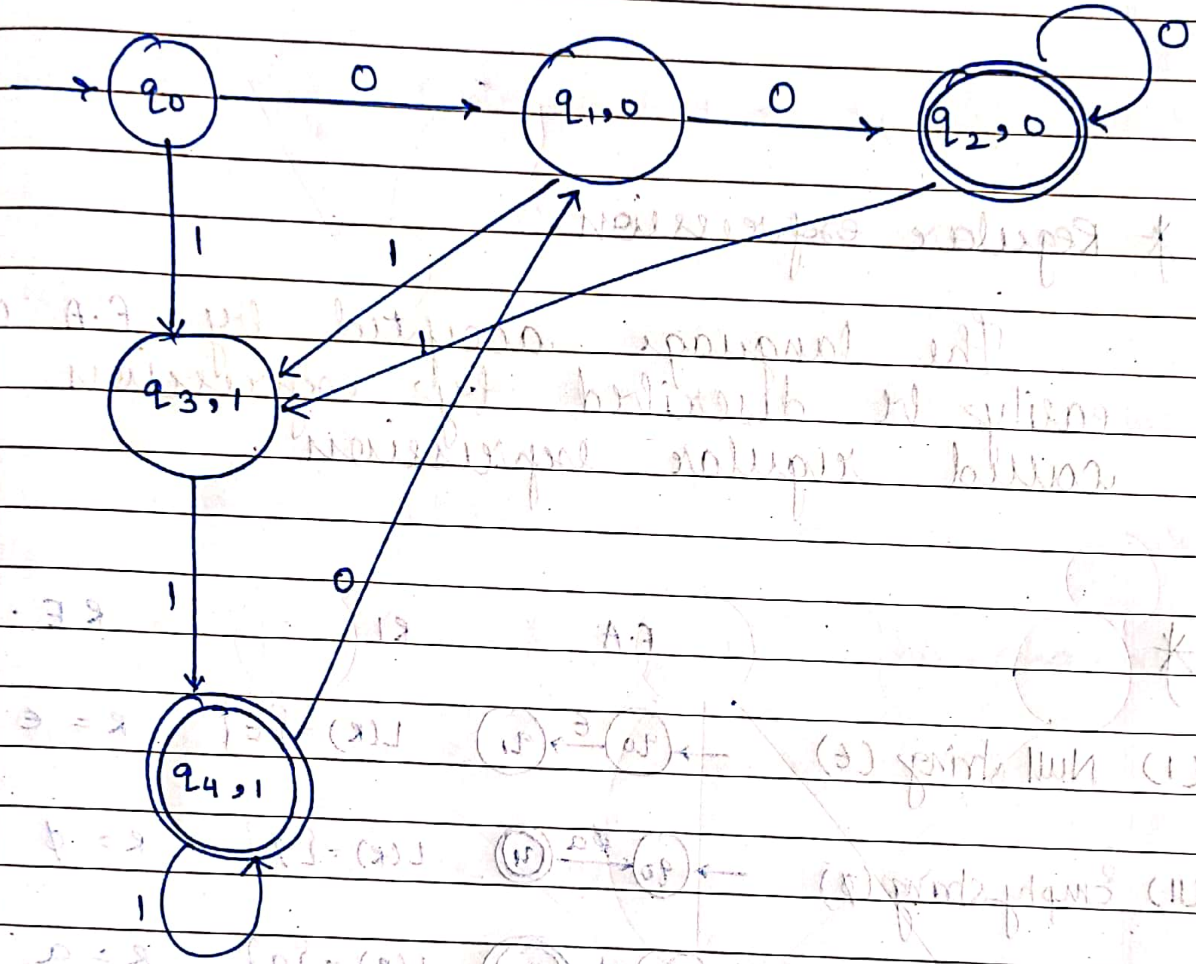
Ans:- $L = \{00, 11, 000, 011, 111, 0111, \ldots\}$



for string ending with 00 or 11
if i/p = 1, o/p = 1
i/p = 0, o/p = 0

Q \ Σ	0	1
→ q0	(q1, 0)	(q3, 1)
q1	(q2, 0)	(q3, 1)
* q2	(q2, 0)	(q3, 1)
q3	(q1, 0)	(q4, 1)
* q4	(q1, 0)	(q4, 1)

→ moore machine.



Q \ Σ	0	1	Δ
→ q_0	q_1	q_3	–
q_1	q_2	q_3	0
* q_2	q_2	q_3	0
q_3	q_1	q_4	1
* q_4	q_1	q_4	1

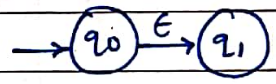
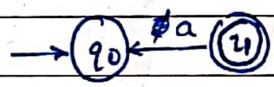
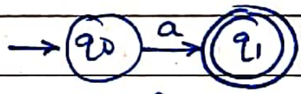
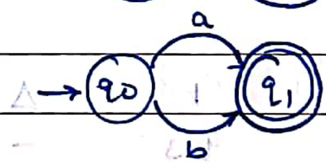
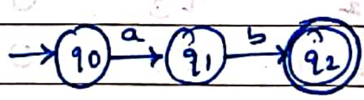
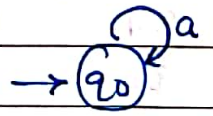
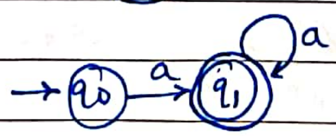
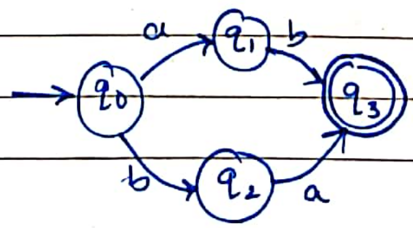
UNIT II

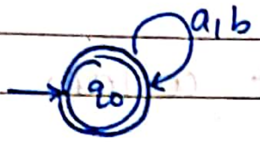
Regural Expression (RE)

Language → Finite Automata (FA) $\xrightarrow{\text{expression}}$ RE

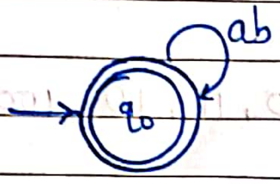
* Regulare expression

The language accepted by F.A can easily be described by expression is called regular expression.

*		F.A	RL	R E.
(I)	Null string (ϵ)		$L(R) = \{\epsilon\}$	$R = \epsilon$
(II)	Empty string (ϕ)		$L(R) = \{\phi\}$	$R = \phi$
(III)			$L(R) = \{a\}$	$R = a$
(IV)	union/OR (+)		$L(R) = \{a, b\}$	$R = a+b$
(V)	concatenation (·)		$L(R) = \{ab\}$	$R = ab$
(VI)	closure (*)		$L(R) = \{\epsilon, a, aa, \dots\}$	$R = a^*$
(VII)	+ve closure		$L(R) = \{a, aa, aaa \dots\}$	$R = aa^*$ or a^+
(VIII)			$L(R) = \{ab, ba\}$	$R = ab+ba$

IX 

$L(R) = \{\epsilon, a, b, ab, aa, bb, aab \dots\}$ $R = (a+b)^*$

X 

$L(R) = \{\epsilon, ab, abab \dots\}$ $R = (ab)^*$

Q* Write R.E. for the Language accepting all Strings that can take any no. 'a' & 'b'.

Ans- R.E = $(a+b)^*$

$L(R) = \{\epsilon, a, b, ab, ba, aa, bb, aba, abb, \dots\}$

* Starting with ~~one~~ '1' & ending with 0.

Ans:- $L(R) = \{10, 110, 100, 1010010 \dots\}$

$RE = 1(1+0)^*0$

* Starting with 'a' but not having consecutive 'b'

Ans:- $L(R) = \{a, ab, aba, aab, aaba, aaa, \dots\}$

$RE = (a+ab)^*$

* All the strings do not contain the substring '01'

Ans:- $L(R) = \{0, 1, 00, 11, 10, 100, 111000 \dots\}$

$R.E = 1^* 0^*$

* At least 2 consecutive '0'

Ans:- $L(R) = \{aa, aab, baa, aaab, babaaba \dots\}$

$R.E = (a+b)^* aa (a+b)^*$

inkem2012
Q3a- * The set of strings over alphabet $\Sigma = \{0,1\}$ that have atleas one '1'.

$L(R) = \{1, 10, 01, 110, 11, 011 \dots\}$

$R.E = (1+0)^* 1 (1+0)^*$

* Atmost one '1' (Add ϵ)

$L = \{\epsilon, 0, 1, 00, 01, 10, 0001, 1000\}$

$RE = 0^* (1+\epsilon) 0^*$

* ending with '00' & begining with '1'

$$R.E = 1(0+1)^*00$$