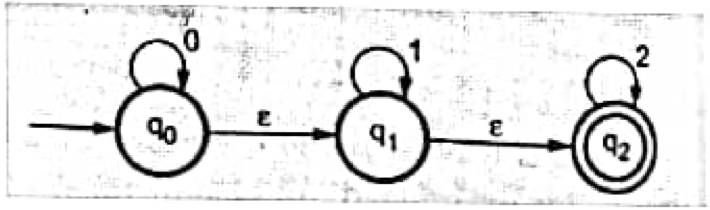
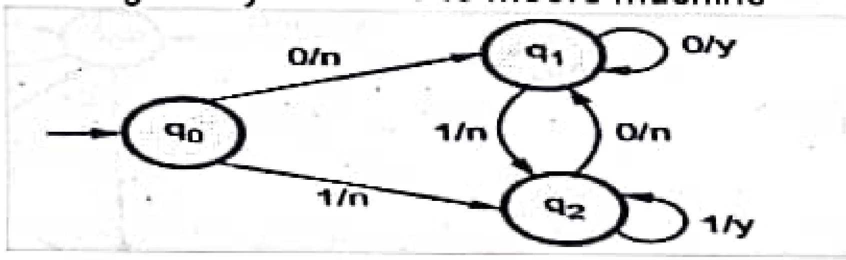
10 marks::

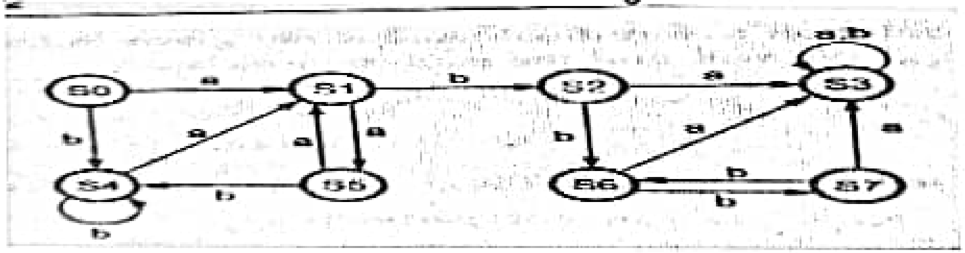
1.Convert from ε-NFA to DFA



2.Explain about Melay and Moore Machines. Write any three differences.
Convert the following Melay Machine to Moore Machine

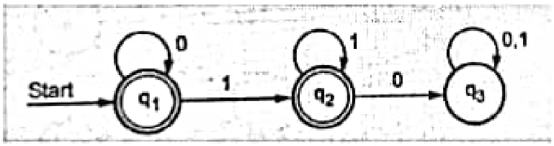


3..Illustrate Optimization of DFA for the following

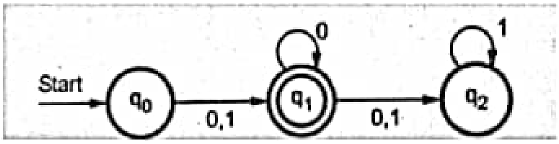


s2,s7 are final states,so is initial state.

4.What is Arden's Theorem. Illustrate Arden's Theorem with the following examples.

a) 

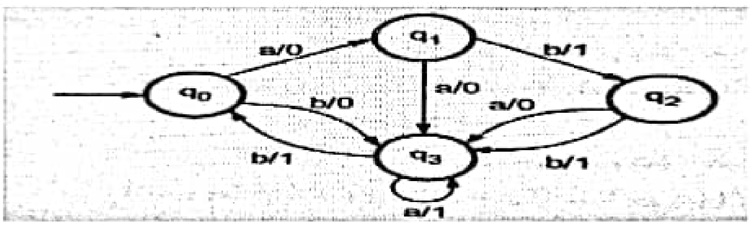
b)



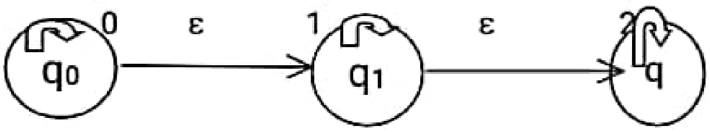
5.Construct RG from the given RE

(ab+a)*(aa+b)

6.Explain about Melay and Moore Machines. Write any three differences. Convert the following Melay Machine to Moore Machine



7.Convert the given epsilon -NFA to NFA.



q_2-- final state

(OR)

8..A)Construct a Moore machine to determine the residue modulo 3 of the input treated as a binary number.

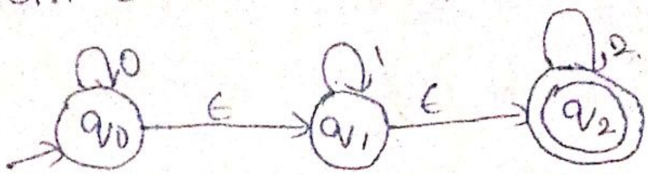
b)Construct a mealy machine to prints 'a' whenever sub string 01 occurred.

9. What is the Method of Conversion from Regular Expression to ε-NFA. Convert the following Regular Expression into Finite Automata r=(0+11)*(00+1)*

10.

1st question

Convert from ε-NFA to DFA



Let M be the given ε-NFA

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

$\Sigma = \{0, 1, 2, \epsilon\}$

δ = is a transition function

states	I/P symbol 0	1	2	ε
$\rightarrow q_0$	q_0	–	–	q_1
q_1	–	q_1	–	q_2
q_2	–	–	q_2	–

$q_0 = q_0$

$F = \{q_2\}$

Now find out ε-closure for all states in the given ε-NFA

ε-closure $q_0 = \{q_0, q_1, q_2\}$

ε-closure $q_1 = \{q_1, q_2\}$

ε-closure $q_2 = \{q_2\}$

Now compute δ' transition ~~state~~

$\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 0))$

$= \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 0))$

$= \epsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-closure}(q_0 \cup \phi \cup \phi)$

$\delta'(q_0, 0) = \epsilon\text{-closure}(q_0)$

$= \{q_0, q_1, q_2\}$

$\delta'(q_0, 1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 1))$

$= \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 1))$

$= \epsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \epsilon\text{-closure}(\phi \cup q_1 \cup \phi)$

$= \epsilon\text{-closure}(q_1)$

$\delta'(q_0, 1) = \{q_1, q_2\}$

$\delta'(q_0, 2) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 2))$

$= \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 2))$

$= \epsilon\text{-closure}(\delta(q_0, 2) \cup \delta(q_1, 2) \cup \delta(q_2, 2))$

$= \epsilon\text{-closure}(\phi \cup \phi \cup q_2)$

$\delta'(q_0, 2) = \{q_2\}$

$\delta'(q_1, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1), 0))$

$= \epsilon\text{-closure}(\delta(\{q_1, q_2\}, 0))$

$= \epsilon\text{-closure}(\delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-closure}(\phi \cup \phi)$

$\delta'(q_1, 0) = \phi$

$\delta'(q_1, 1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1), 1))$

$= \epsilon\text{-closure}(\delta(q_1, 1) \cup \delta(q_2, 1))$

$= \epsilon\text{-closure}(q_1 \cup \phi)$

$= \epsilon\text{-closure}(q_1)$

$\delta'(q_1, 1) = \{q_1, q_2\}$

$\delta'(q_1,2) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1),2))$

$= \epsilon\text{-closure}(\delta(\{q_1,q_2\},2))$

$= \epsilon\text{-closure}(\delta(q_1,2) \cup \delta(q_2,2))$

$= \epsilon\text{-closure}(\phi \cup q_2)$

$= \epsilon\text{-closure}(q_2)$

$\delta'(q_1,2) = \{q_2\}$

$\delta'(q_2,0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_2),0))$

$= \epsilon\text{-closure}(\delta((q_2),0))$

$= \epsilon\text{-closure}(\phi)$

$\delta'(q_2,0) = \phi$

$\delta'(q_2,1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_2),1))$

$= \epsilon\text{-closure}(\delta(q_2,1))$

$= \epsilon\text{-closure}(\phi)$

$\delta'(q_2,1) = \phi$

$\delta'(q_2,2) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_2),2))$

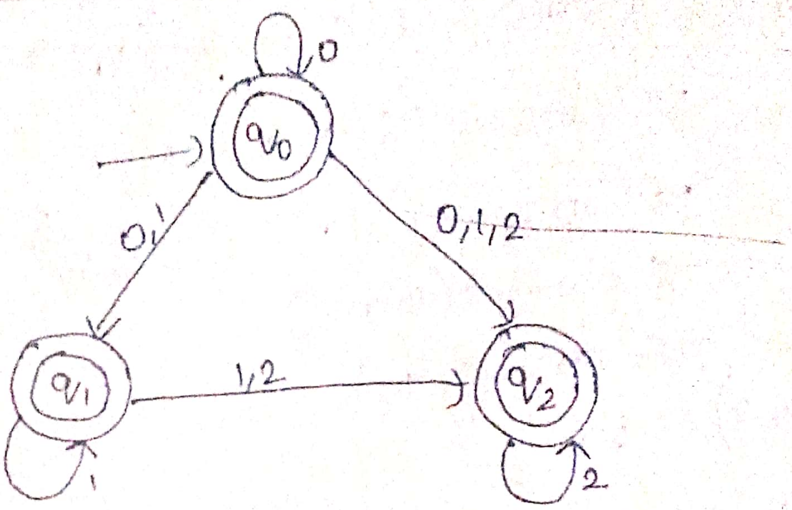
$= \epsilon\text{-closure}(\delta(q_2,2))$

$= \epsilon\text{-closure}(q_2)$

$\delta'(q_2,2) = q_2.$

	I/P symbol		
State	0	1	2
q_0	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$	q_2
q_1	ϕ	$\{q_1,q_2\}$	q_2
q_2	ϕ	ϕ	q_2

Scanned with OKEN Scanner



Then the DFA is

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

Apply subset construction Algorithm

~~$\delta(\{q_0,$~~

	I/p Symbol		
state	0	1	2
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	q_2
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	q_2
$\{q_1, q_2\}$	ϕ	$\{q_1, q_2\}$	q_2
q_2	ϕ	ϕ	q_2

$$\delta(\{q_0, q_1, q_2\}, 0) = \delta((q_0, 0)) \cup \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \{q_0, q_1, q_2\} \cup \phi \cup \phi$$
$$= \{q_0, q_1, q_2\}$$

$$\delta(\{q_0, q_1, q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1)$$
$$= \{q_1, q_2\} \cup \{q_1, q_2\} \cup \phi$$
$$= \{q_1, q_2\}$$

$\delta(\{q_0,q_1,q_2\},2) = \delta(q_0,2) \cup \delta(q_1,2) \cup \delta(q_2,2)$

$= q_2 \cup q_2 \cup q_2$

$\delta(\{q_0,q_1,q_2\},2) = q_2$

$\delta(\{q_1,q_2\},0) = \delta(q_1,0) \cup \delta(q_2,0)$

$= \phi \cup \phi$

$= \phi$

$\delta(\{q_1,q_2\},1) = \delta(q_1,1) \cup \delta(q_2,1)$

$= \{q_1,q_2\} \cup \phi$

$= \{q_1,q_2\}$

$\delta(\{q_1,q_2\},2) = \delta(q_1,2) \cup \delta(q_2,2)$

$= q_2 \cup q_2$

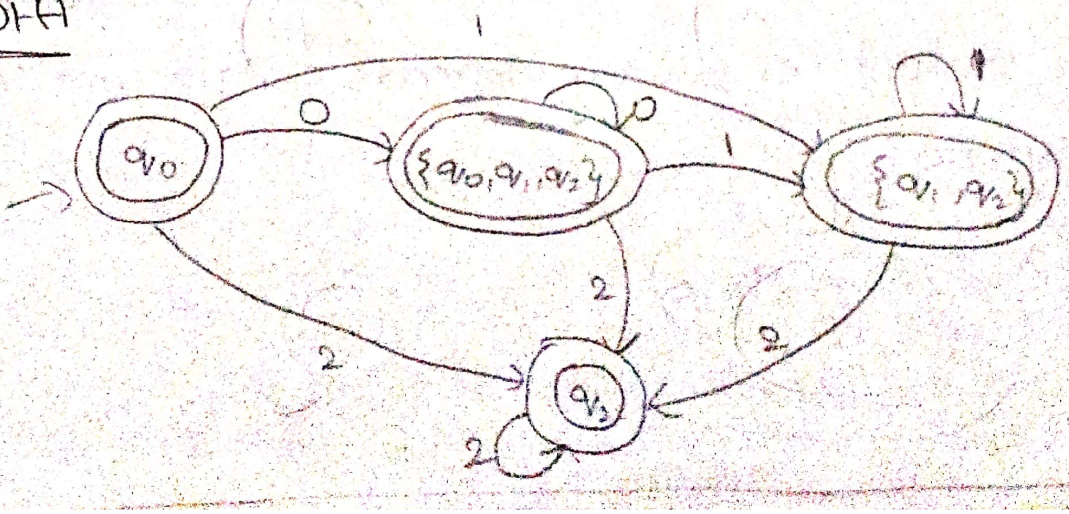
$= q_2$

$\delta(q_2,0) = \phi$

$\delta(q_2,1) = \phi$

$\delta(q_2,2) = q_2$

DFA



2) Explain about Melay and Moore machines. Write any differences. convert the following melay machine to Moore machine.

Moore Machine:

Moore machine is a finite automata that contains set of states in which the output is always depends on present state only

Mathematically moore machine is 6 tuples like

$$M = (Q, \Sigma, \delta, \lambda, q_0, \Delta)$$

Q = finite and nonempty set of state

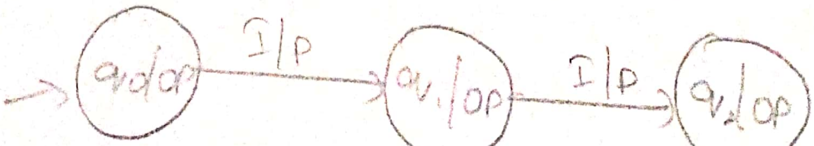
Σ = finite and non empty set of input symbols

δ = Transition function

Δ = finite & non empty set of output symbols or output alphabets

λ = Output function

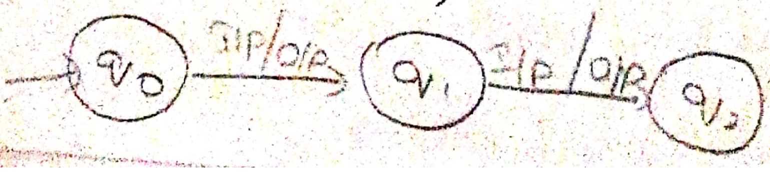
q_0 = initial state.



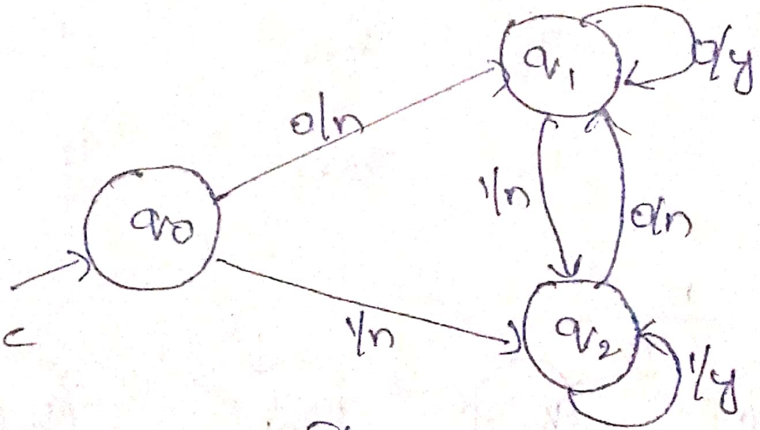
Mealy machine.

It is a finite automata contains set of states in which the output is alway depends on present state and input symbol

$$M = (Q, \Sigma, \delta, \lambda, q_0, \Delta)$$



Moore Machine	Mealy machine
Output depends only on present State	Output depends on present State as well as Present input
More states are required	Less no of states are required
Easy to design	It is difficult to design



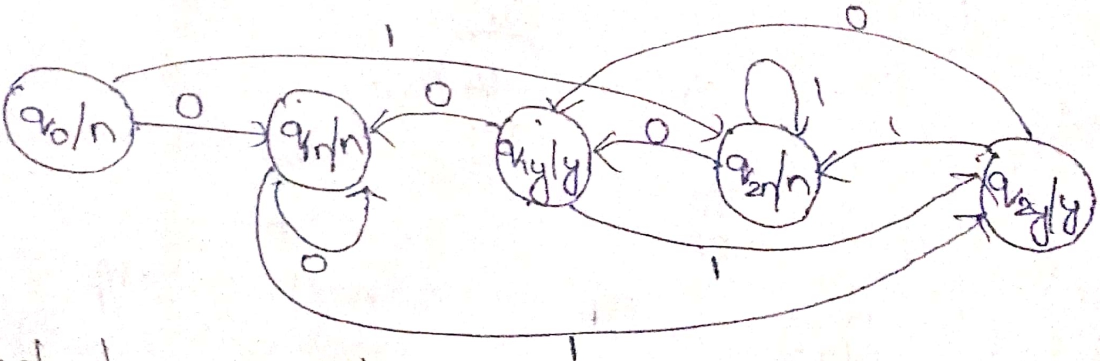
Mealy table

Present State	I/P 0 next state	o/p	I/P 1 next State	o/p
$\rightarrow q_0$	q_1	n	q_2	n
q_1	q_1	n	q_2	y
q_2	q_1	y	q_2	n

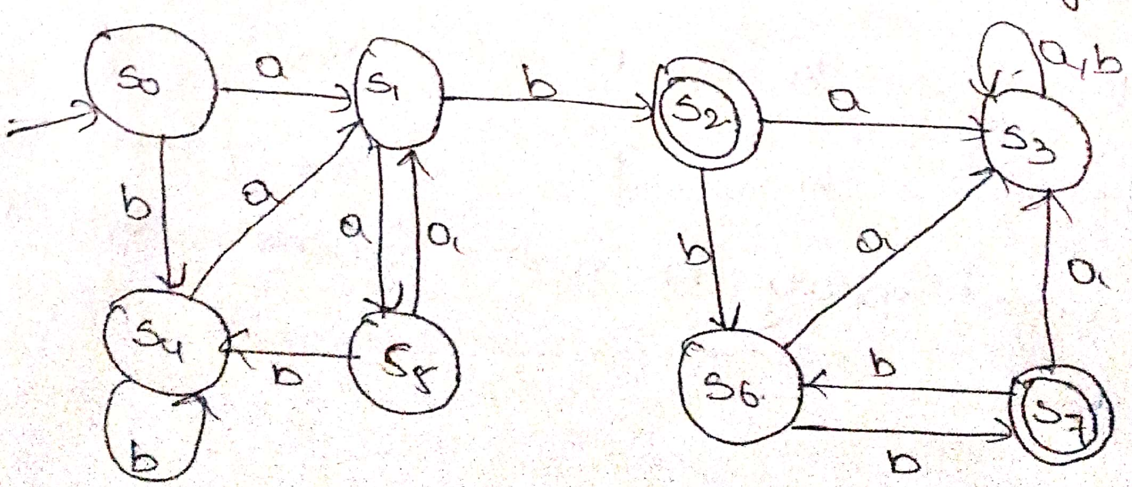
Now convert mealy to moore machine the transitio table is

Moore table.

Present State	Next state 0	Next state 1	O/P
q_0	q_{1n}	q_{2n}	n
q_{1n}	q_{1n}	q_{2y}	n
q_{1y}	q_{1n}	q_{2y}	y
q_{2n}	q_{1y}	q_{2n}	n
q_{2y}	q_{1y}	q_{2n}	y



2) Illustrate optimization of DFA for the following.



the finite automata $M = (Q, \delta, \Sigma, q_0, F)$

$Q = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$

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

Transition table.

States	I/p symbol a	b
$\rightarrow s_0$	s_1	s_4
s_1	s_5	s_2
s_2	s_3	s_6
s_3	s_3	s_3
s_4	s_1	s_4
s_5	s_1	s_4
s_6	s_3	s_7
s_7	s_3	s_6

Now we have to calculate zero equivalence.

Zero equivalence

It contains mainly two sets

i, Non final state

ii, final state

$\Pi_0 = \{s_0, s_1, s_3, s_4, s_5, s_6\} \{s_2, s_7\}$

One equivalence

It will be calculated based on 0-equivalence

now $\Pi_1 = (s_0, s_1) = \delta(s_0, a) = s_1 \qquad \delta(s_0, b) = s_4$

$\delta(s_1, a) = s_5 \qquad \delta(s_1, b) = s_2$

Here s_1 & s_5 are in same set

and s_4 & s_2 are not in same set

$\therefore$ $s_0 \neq s_1$

$\Pi_1 = (s_0, s_3)$: $\delta(s_0, a) = s_1$ $\quad \delta(s_0, b) = s_4$

$\delta(s_3, a) = s_3$ $\quad \delta(s_3, b) = s_3$

here s_1 & s_3 are in same set also s_4 & s_3 belongs to same set

$\therefore$ $s_0 = s_3$

$\Pi_1 = (s_0, s_4)$ = $\delta(s_0, a) = s_1$ $\quad \delta(s_0, b) = s_4$

$\delta(s_4, a) = s_1$ $\quad \delta(s_4, b) = s_4$

here s_1 & s_1 are in same set also s_4 & s_4 belongs to same set

$\therefore$ $s_0 = s_4$

$\Pi_1 = (s_0, s_5)$ = $\delta(s_0, a) = s_1$ $\quad \delta(s_0, b) = s_4$

$\delta(s_5, a) = s_1$ $\quad \delta(s_5, b) = s_4$

here s_1 & s_4 & s_4 & s_4 belongs to same set.

$\therefore$ $s_0 = s_5$

$\Pi_1 = (s_0, s_6)$: $\delta(s_0, a) = s_1$ $\quad \delta(s_0, b) = s_4$

$\delta(s_6, a) = s_3$ $\quad \delta(s_6, b) = s_7$

here s_1 & s_3 belongs to same set but s_4 & s_7 does not belongs to same set

$\therefore$ $s_0 \neq s_6$

~~$\therefore \Pi_1 = \{s_0, s_3, s_4, s_5\} \{s_1\} \{s_6\} \{s_6$~~

$\Pi_1 = (s_2, s_7)$: $\delta(s_2, a) = s_3$ $\quad \delta(s_2, b) = s_6$

$\delta(s_7, a) = s_3$ $\quad \delta(s_7, b) = s_6$

Scanned with OKEN Scanner

Here s_3 & s_6 belongs to same set

$\therefore$ $s_2 = s_7$

$\therefore \pi_1 = \{s_0, s_3, s_4, s_5\} \{s_1\} \{s_6\} \{s_2, s_7\}$

2-equivalence

$\pi_2 = (s_0, s_3)$: $\delta(s_0, a) = s_1$ $\quad$ $\delta(s_0, b) = s_4$

$\delta(s_3, a) = s_3$ $\quad$ $\delta(s_3, b) = s_2$

Here s_1 & s_3 does not belongs to same set

$\therefore$ $s_0 \neq s_3$.

$\pi_2 = (s_0, s_4)$: $\delta(s_0, a) = s_1$ $\quad$ $\delta(s_0, b) = s_4$

$\delta(s_4, a) = s_1$ $\quad$ $\delta(s_4, b) = s_4$

$\therefore$ $s_0 = s_4$

$\pi_2 = (s_0, s_5) = \delta(s_0, a) = s_1$ $\quad$ $\delta(s_0, b) = s_4$

$\delta(s_5, a) = s_1$ $\quad$ $\delta(s_5, b) = s_4$

$\therefore$ $s_0 = s_5$

$\therefore$ $\pi_2 = \{s_0, s_4, s_5\} \{s_1\} \{s_6\} \{s_3\} \{s_2, s_7\}$

3-equivalence

$\pi_3 = (s_0, s_4)$: $\delta(s_0, a) = s_1$ $\quad$ $\delta(s_0, b) = s_4$

$\delta(s_4, a) = s_1$ $\quad$ $\delta(s_4, b) = s_4$

s_1 & s_1 belongs to same set & s_4, s_4 belongs to same set

$\therefore$ $s_0 = s_4$

$\pi_3 = (s_0, s_5) = \delta(s_0, a) = s_1 \quad \delta(s_0, b) = s_4$

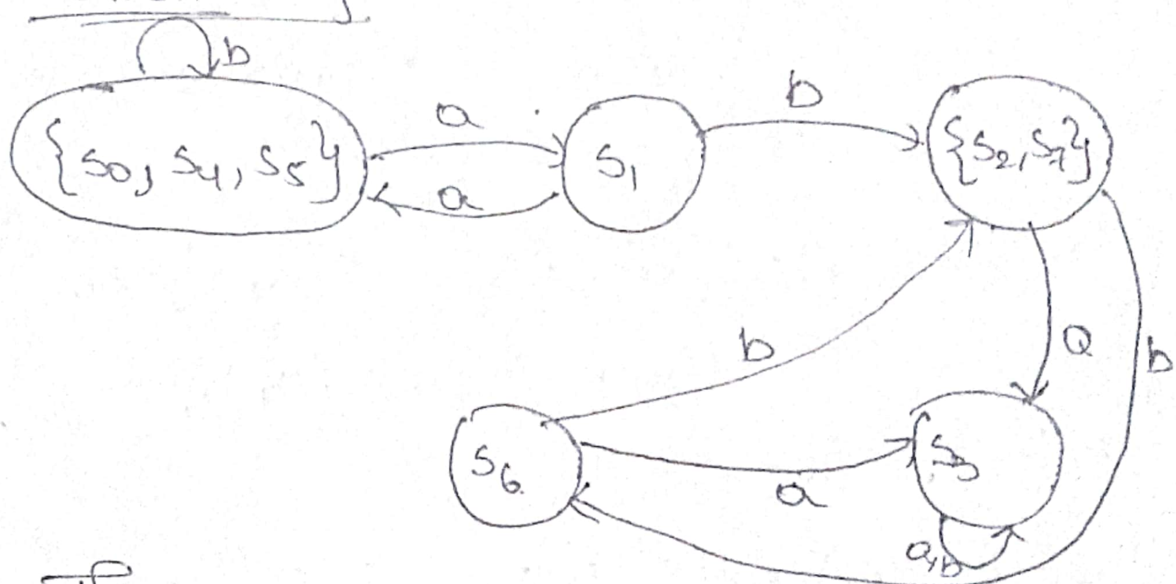
$\delta(s_5, a) = s_1 \quad \delta(s_5, b) = s_4$

$s_0 = s_5$

$\therefore \pi_3 = \{s_0, s_4, s_5\} \{s_1\} \{s_6\} \{s_3\} \{s_2, s_7\}$

$\therefore \pi_2 = \pi_3.$

Transition diagram



4) Arden's Theorem.

This therom is used to checking the equivalence of two regular expression

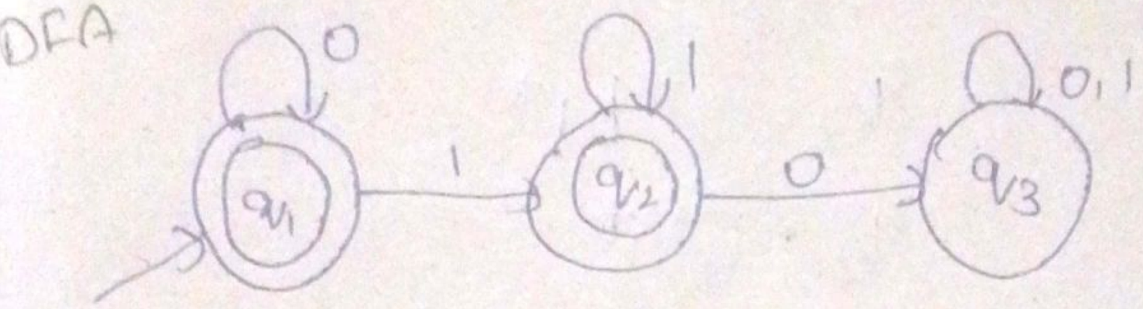
Theorem:

Let P & Q be two regular expression over the input set Σ if P does not contain ε then the following equation

$R = Q + RP$ has unique solution as $R = QP^*$

4a question

construct regular expression for the following DFA



let us write down the eqn.

$$q_1 = q_1 0 + \epsilon$$

$$q_2 = q_2 1 + q_1 1$$

$$q_3 = q_3 0 + q_3 1 + q_2 0$$

$$q_3 = q_3 (0+1) + q_2 0$$

Now simplify q_1

$$q_1 = \epsilon + q_1 0$$

where it is similar $R = Q + RP$

$R = q_1 \quad Q = \epsilon \quad P = 0$

reduced to $R = QP^*$

$$R = \epsilon 0^*$$

$$q_1 = 0^*$$

Substitute q_1 in q_2.

$q_2 = q_2 1 + q_1 1$

$q_2 = q_2 1 + 0^* 1$

$q_2 = 0^* 1 + q_2 1$

This is in the form $R = Q + RP$

$R = q_2 \quad Q = 0^* 1 \quad P = 1$

Reduced to $R = QP^*$

$q_2 = 0^* 1 1^*$

$q_2 = 0^* 1^+$

substitute q_2 in q_3

$q_3 = q_3 (0+1) + q_2 0$

$= q_3 (0+1) + 0^* 1^+ 0$

$q_3 = 0^* 1^+ + q_3 (0+1)$

This is in the form $R = Q + RP$

$q_3 = 0^* 1^+ + q_3 (0+1)$

$R = q_3 \quad P = (0+1) \quad Q = 0^* 1^+$

Reduced to $R = QP^*$

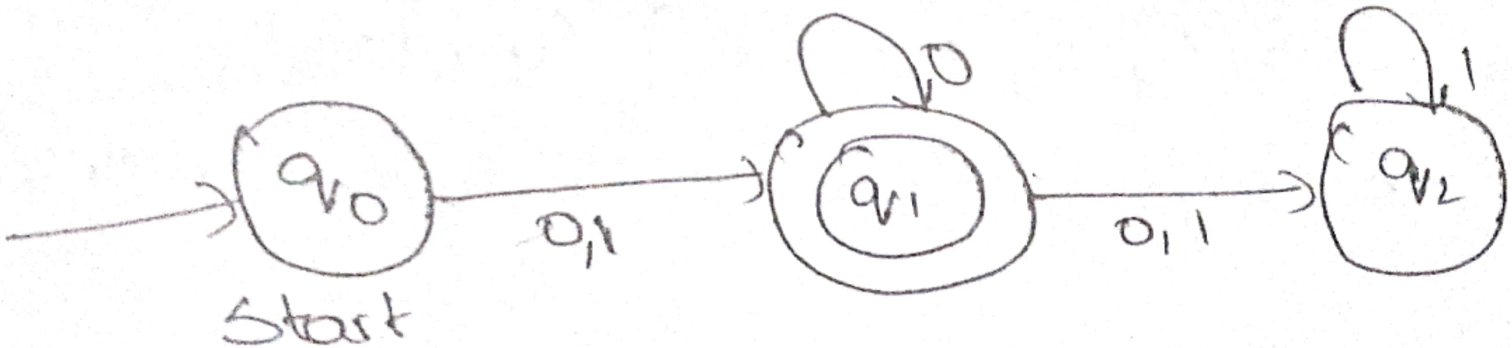
$q_3 = 0^* 1^+ (0+1)^*$ $\quad 0^* + 0^* 1^+$

As the RE are final states

$R.E = q_1 + q_2 = 0^* + 0^* 1^+$

R = Q+RP has unique solution as R = QP*

4b)



$q_0 = \phi + \epsilon \Rightarrow \epsilon$

$q_1 = q_0 1 + q_0 0 + q_1 0 \Rightarrow q_1 0 + q_0(1+0)$

$q_2 = q_1 0 + q_1 1 + q_2 1$

$q_2 = q_1(0+1) + q_2 1$

~~Initial state~~

$\cancel{q_0 = \epsilon + \phi}$

~~It is similar to R~~

$q_1 = q_0(1+0) + q_1 0$

It is similar to $R = Q + RP$

Where $R = q_1$ $P = 0$ $Q = q_0(1+0)$

then reduce to $R = QP^*$

$R = q_0(1+0)(0)^*$

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

$q_1 = (1+0)(0)^*$

$\therefore$ The regular expression $= (1+0)(0)^*$

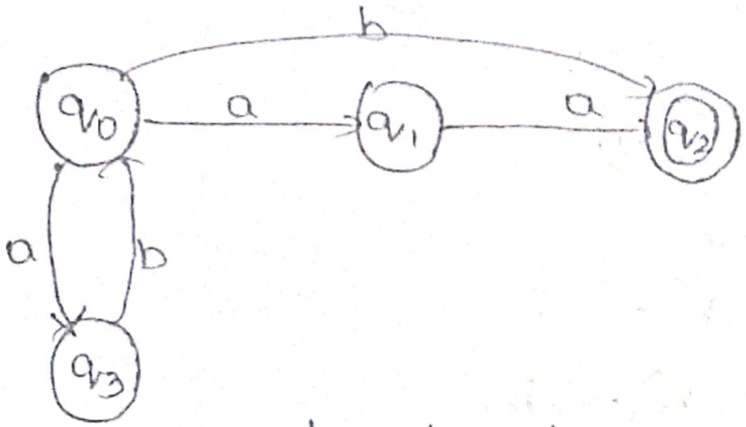
5th question

Construct regular grammar from the given RE $(ab+a)^*(aa+b)$

The given RE is $(ab+a)^*(aa+b)$

$$\left(a[b+\epsilon]\right)^*(aa+b)$$

$$[ab]^*(aa+b)$$



The Given Finite Autometa is NFA convert it into DFA

states	I/p symols a	b
$\to q_0$	$\{q_1, q_3\}$	q_2
q_1	q_2	-
q_2	-	-
q_3	-	q_0

Now for DFA

$$\delta(\{q_1, q_3\}, a) = \delta(q_1, a) \cup \delta(q_3, a)$$
$$= q_2 \cup \phi$$
$$= q_2$$

$\delta(\{q_1,q_3\},b) = \delta(q_1,b) \cup \delta(q_3,b)$

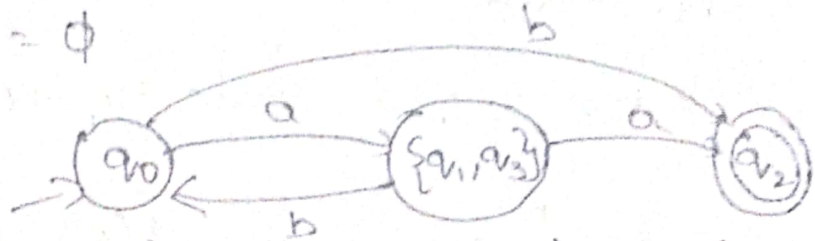
$= \phi \cup q_0$

$= q_0$

states	I/p symbols a	b
$\to q_0$	$\{q_1,q_3\}$	q_2
$\{q_1,q_3\}$	q_2	q_0
q_2	ϕ	ϕ

$\delta(q_2,a) = \phi$

$\delta(q_2,b) = \phi$



The transition form $\delta(q_0,a) \to \{q_1,q_3\}$, $\delta(q_0,b) \to q_2$

$\delta(\{q_1,q_3\},a) \Rightarrow q_2$, $\delta(\{q_1,q_3\},b) \Rightarrow q_0$,

$\delta(q_0,a) \to \{q_1,q_3\}$ can be written as $q_0 \to a\{q_1,q_3\}$

$\delta(q_0,b) \to \{q_2\}$ can be written as $q_0 \to b\,q_2$

$\delta(\{q_1,q_3\},a) \to q_2$ can be written as $\{q_1,q_3\} \to a$

$\delta(\{q_1,q_3\},b) \to q_0$ can be written as $\{q_1,q_3\} \to b q_0$

$\therefore$ The productions are $q_0 \to a\{q_1,q_3\}$

$q_0 \to bq_2 | b$

$\{q_1,q_3\} \to aq_2 | a$

$\{q_1,q_3\} \to bq_0$

RG: $\{V,T,P,S\}$

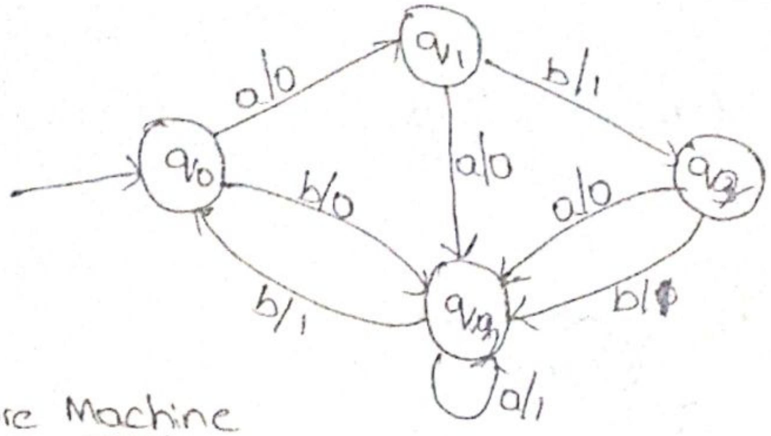
$V = \{q_0, \{q_1,q_3\}, q_2\}$, $T = \{a,b\}$, $S = \{q_0\}$

$P = \{q_0 \to a\{q_1,q_3\}, q_0 \to bq_2 | b, \{q_1,q_3\} \to aq_2 | a, \{q_1,q_3\} \to b$

construction of Finite Automata from RG and conversion from RG to FA

6th question

6) Explain about melay and moore machines. Write any 3 differences. convert the following melay machine to moore machine.



Moore Machine

Moore machine is a finite autometa that contains set of states in which the output is always depends on present state only.

Mathematically moore machine is 6 tuples

$$M = (Q, \Sigma, \delta, \lambda, q_0, \Delta)$$

Q = finite and non empty set of state.

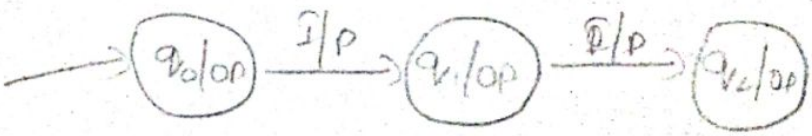
Σ = finite and non-empty set of input symbols

δ = Transition function

Δ = finite & non empty set of output symbols

λ = output function

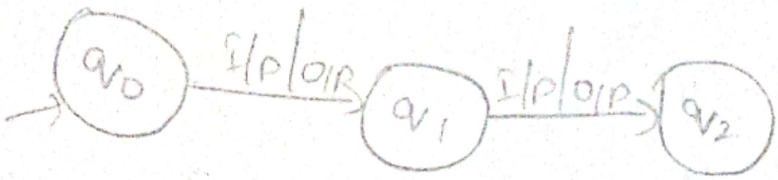
q_0 = initial state.



Mealy machine:

It is a finite autometa contains set of states in which the o/p is always depends on present state nd output symbol

$M = (Q, \Sigma, \delta, \lambda, q_0, \Delta)$

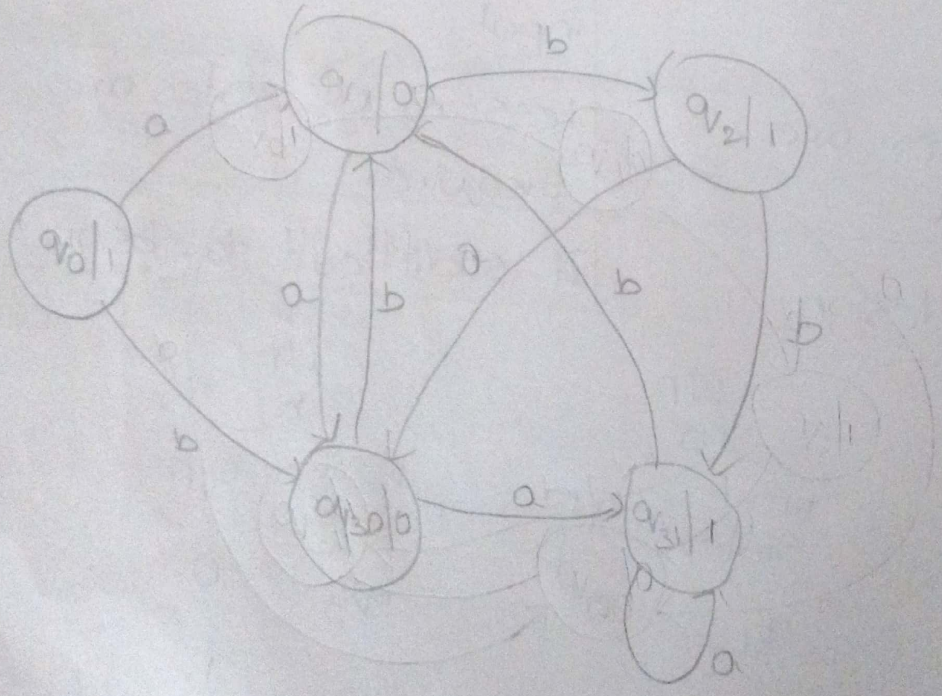


Moore Machine	Mealy Machine
Output depends only on Present state	output depends on present State as well as present input
More states are required.	Less no of states are required
Easy to design	It is difficult to design

Present state.	I/p a Next State	I/p a O/p	I/p b Next State	I/p b O/p
q_0	q_1	0	q_3	0
q_1	q_3	0	q_2	1
q_2	q_3	0	q_3	1
q_3	q_3	1	q_0	1

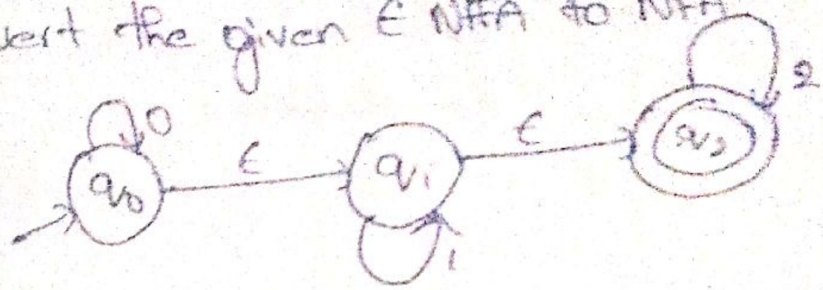
Now convert mealy to moore machine the transition table is.

Present state	Next state a	Next state b	O/P
q_0	q_1	q_{30}	1
q_1	q_{30}	q_2	0
q_2	q_{30}	q_{31}	1
q_{30}	q_{31}	q_1	0
q_{31}	q_{31}	q_1	1



7th question

Convert the given ϵ NFA to NFA



Let M be a given ϵ NFA

$M = (Q, \Sigma, \delta, q_0, F)$

$\Sigma = \{0, 1, 2, \epsilon\}$

δ = is a transition function.

States	I/p symbol 0	1	2	ϵ
$\rightarrow q_0$	q_0	–	–	q_1
q_1	–	q_1	–	q_2
q_2	–	–	q_2	–

$q_0 = q_0$

$F = \{q_2\}$

Now find out ϵ-closure for all states in the given ϵ-NFA

ϵ-closure $q_0 = \{q_0, q_1, q_2\}$

ϵ-closure $q_1 = \{q_1, q_2\}$

ϵ-closure $q_2 = \{q_2\}$

Now compute δ' transition.

$\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 0))$

Scanned with OKEN Scanner

$= \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 0))$

$= \epsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-closure}(q_0 \cup \phi \cup \phi)$

$= \epsilon\text{-closure}(q_0)$

$\delta'(q_0, 0) = \{q_0, q_1, q_2\}$

$\delta'(q_0, 1) = \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 1))$

$= \epsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \epsilon\text{-closure}(\phi \cup q_1 \cup \phi)$

$= \epsilon\text{-closure}(q_1)$

$\delta'(q_0, 1) = \{q_1, q_2\}$

$\delta'(q_0, 2) = \epsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, 2))$

$= \epsilon\text{-closure}(\delta(q_0, 2) \cup \delta(q_1, 2) \cup \delta(q_2, 2))$

$= \epsilon\text{-closure}(\phi \cup \phi \cup q_2)$

$= \epsilon\text{-closure}(q_2)$

$\delta'(q_0, 2) = \{q_2\}$

$\delta'(q_1, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1), 0))$

$= \epsilon\text{-closure}(\delta(q_1, q_2), 0))$

$= \epsilon\text{-closure}(\delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-closure}(\phi \cup \phi)$

$\delta'(q_1, 0) = \phi$

$$\delta'(q_1,1) = \epsilon\text{-closure}(\delta(q_1,q_2),1)$$

$$= \epsilon\text{-closure}(\delta(q_1,1) \cup \delta(q_2,1))$$

$$= \epsilon\text{-closure}(q_1 \cup \phi)$$

$$= \epsilon\text{-closure}(q_1)$$

$$\delta'(q_1,1) = \{q_1, q_2\}$$

$$\delta'(q_1,2) = \epsilon\text{-closure}(\delta(q_1,q_2),2)$$

$$= \epsilon\text{-closure}(\delta(q_1,2) \cup \delta(q_2,2))$$

$$= \epsilon\text{-closure}(\phi \cup q_2)$$

$$\delta'(q_1,2) = \{q_2\}$$

$$\delta'(q_2,0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_2)),0)$$

$$= \epsilon\text{-closure}(\delta(q_2,0))$$

$$= \epsilon\text{-closure}(\phi)$$

$$\delta'(q_2,0) = \phi$$

$$\delta'(q_2,1) = \epsilon\text{-closure}(\delta(q_2,1))$$

$$= \epsilon\text{-closure}(\phi)$$

$$\delta'(q_2,1) = \phi$$

$$\delta'(q_2,2) = \epsilon\text{-closure}(\delta(q_2,2))$$

$$= \epsilon\text{-closure}(q_2)$$

$$\delta'(q_2,2) = \{q_2\}$$

Now the NFA without ϵ moves M' is like

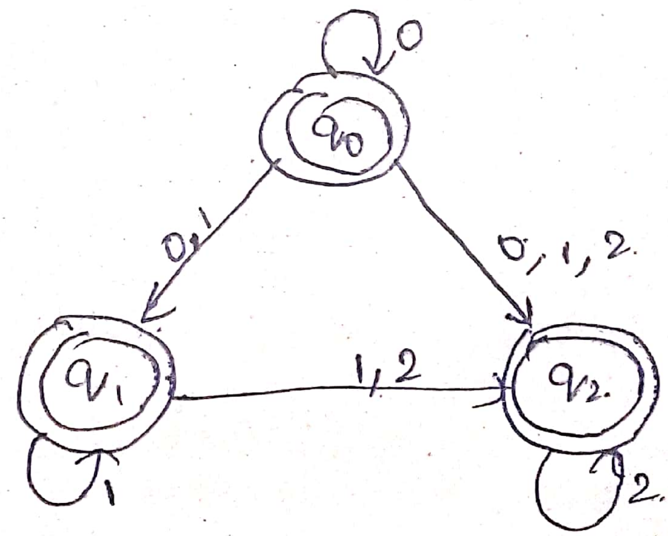
$$M' = (Q, \Sigma, \delta', q_0, F')$$

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

$\Sigma = \{0, 1, 2\}$

$\delta' =$

State	I/P symbol 0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	q_2
q_1	ϕ	$\{q_1, q_2\}$	q_2
q_2	ϕ	ϕ	q_2.



8a question

Design a moore machine for recidue [3] for the input stream is created as a binary number.

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

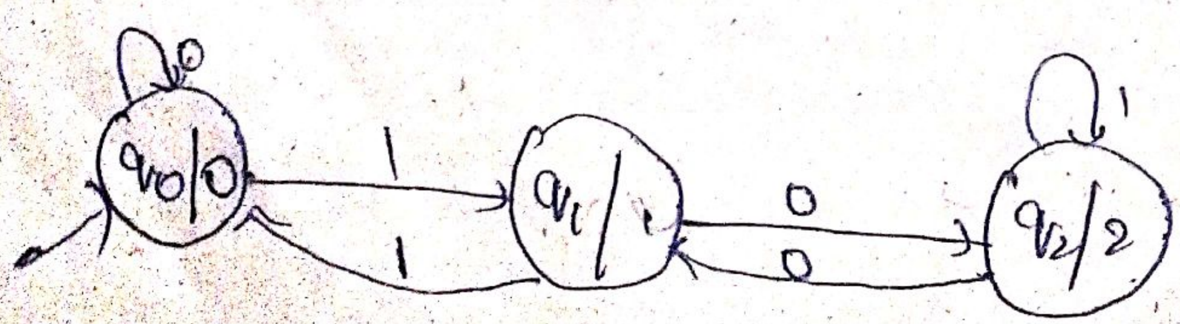
recidue [3] means if any no. is divisible by 3 then we get the possible remainders are 0, 1, 2

$\therefore \Delta = \{0,1,2\}$

Possible states are q_0, q_1, q_2

$\therefore \delta$ = transition function.

Present state	I/P symbol 0	I/P symbol 1	O/P (Δ)
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2



Accept a string $\rightarrow$ 15 - 1 1 1 1

q_0 - q_1 q_0 q_1 q_0 $\Rightarrow$ output = 0

10 - 1 0 1 0

q_0 - q_1 q_2 q_2 q_1 $\Rightarrow$ output = 1

$\rightarrow$ 1

eg:

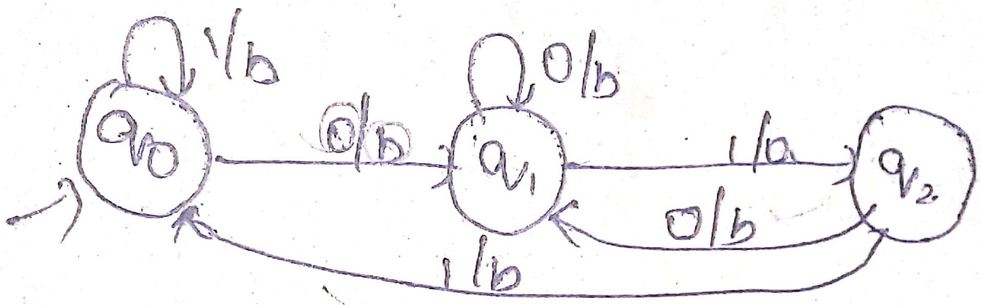
8b question

Print a when ever sub string 0,1 occur

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

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

Possible states are q_0, q_1, q_2.



Present state	I/p symbol a: Next state	I/p symbol a: O/p	I/p symbol b: Next state	I/p symbol b: O/p
$\rightarrow q_0$	q_1	b.	q_0	b
q_1	q_1	b	q_2	a
q_2	q_1	b	q_0	b

9th question

Conversion of RE to FA

To convert the RE to FA, we are going to use a method called the subset method. This method is used to obtain FA from the given regular expression. This method is given below:

Step 1: Design a transition diagram for given regular expression, using NFA with ε moves.

Step 2: Convert this NFA with ε to NFA without ε.

Step 3: Convert the obtained NFA to equivalent DFA.

Construct NFA for ∈ moves for RE $(11+0)^*(00+1)^*$

Given Regular Expression $r = (11+0)^* (00+1)^*$

Let r can be broken into r_1 & r_2 where

$$r_1 = (11+0)^*$$

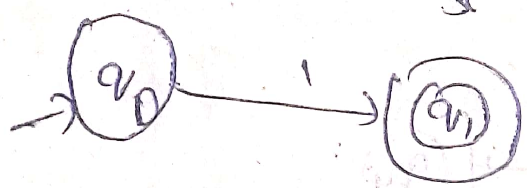
$$r_2 = (00+1)^*$$

Let r_1 can be broken into r_2 & r_3.

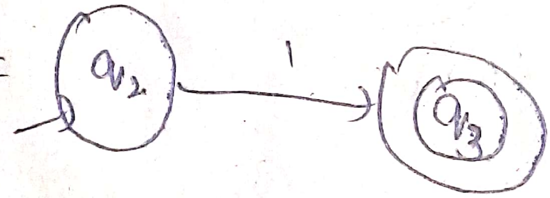
$r_3 = 11$ & $r_4 = 0$

$r_5 = 1$ $\quad$ $r_6 = 1$

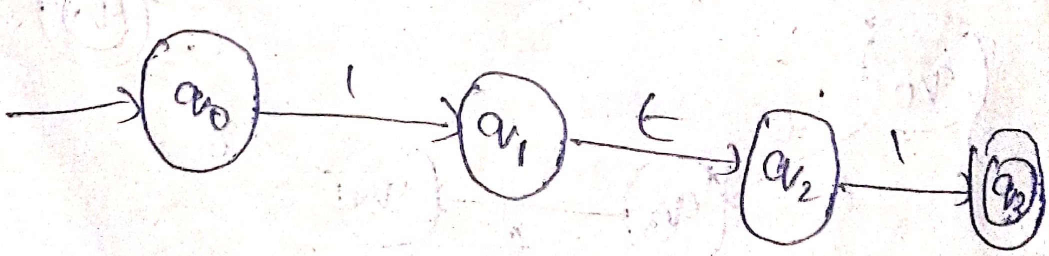
$r_5 =$



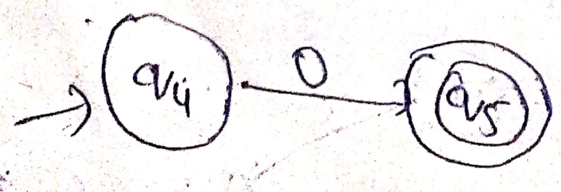
$r_6 =$



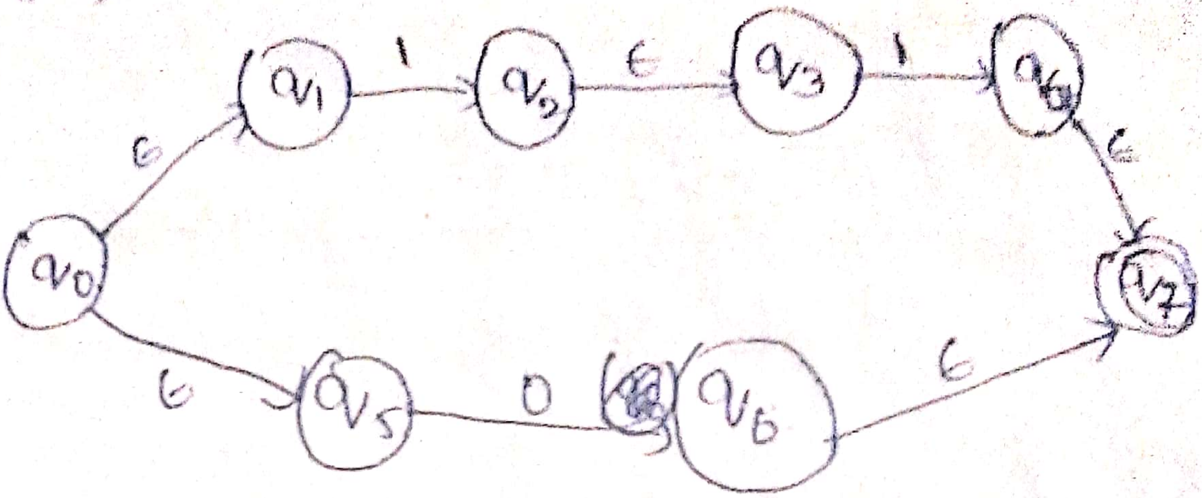
$r_3 = r_5 . r_6 \Rightarrow$



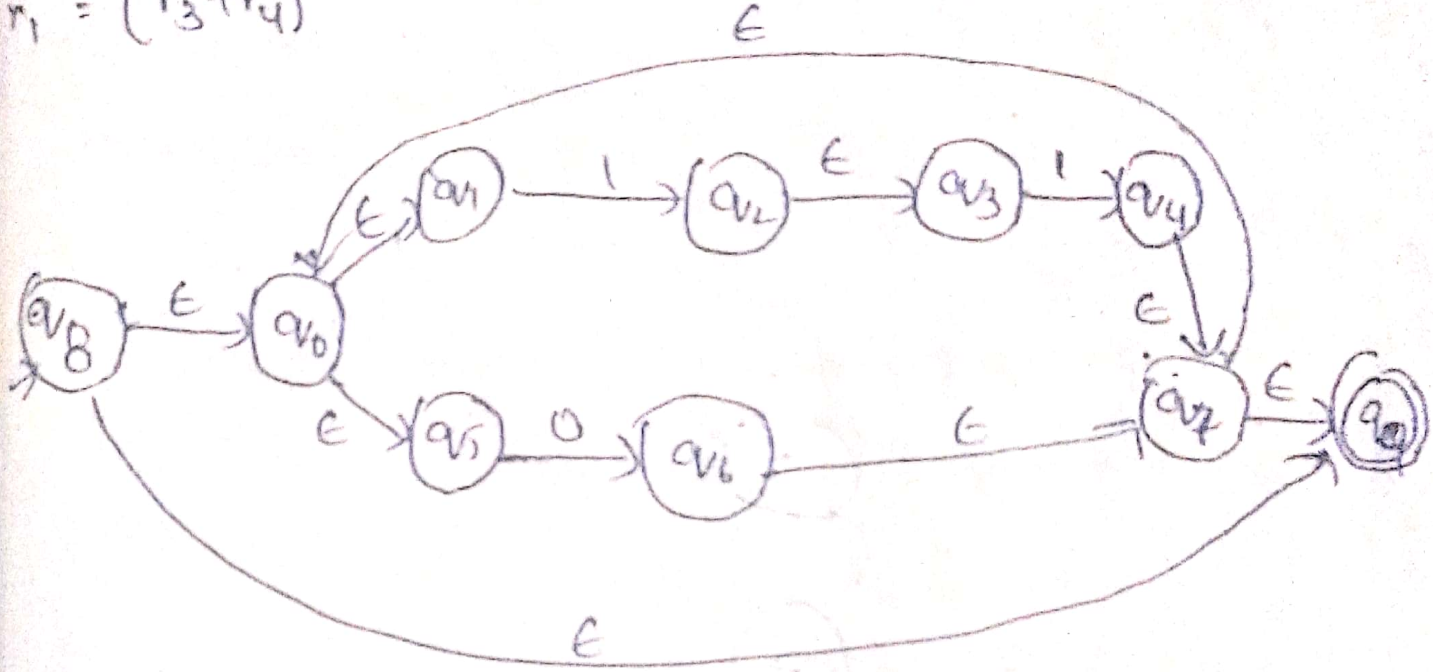
$r_4 = 0$



$r_3 + r_4 \Rightarrow$

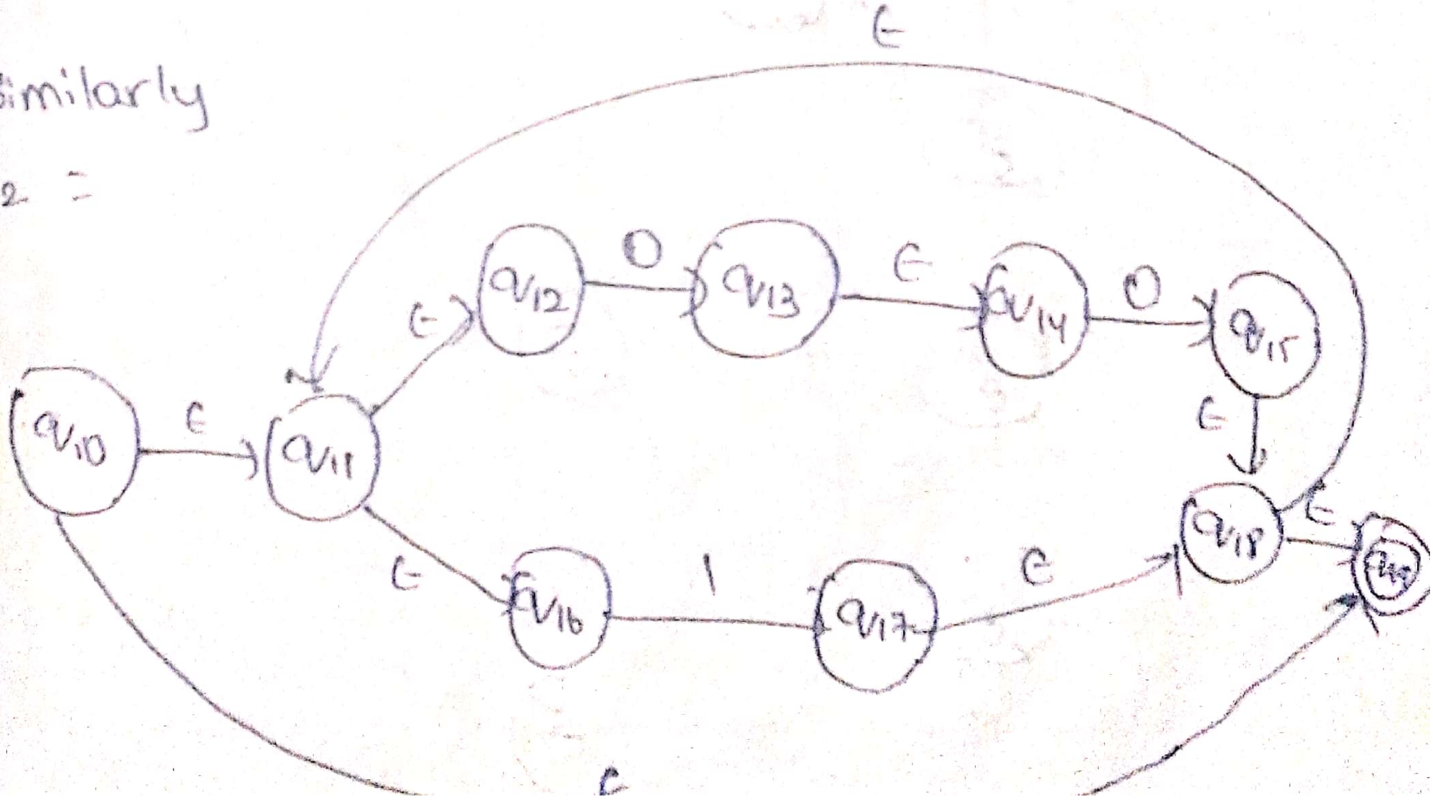


$r_1 = (r_3 + r_4)^*$



similarly

$r_2 =$



$r = r_1 \cdot r_2$