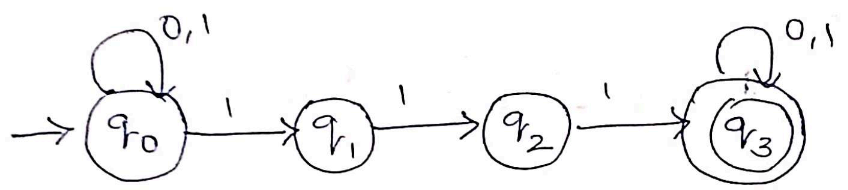
→ Design NFA to accept set of all strings over {0,1} containing 3 consecutive ones.

Sol: step1: The language is

$L = \{111, 0111, 1110 \ldots\}$

step2: The transition diagram for the above language is



step3: The transition table for the given problem is

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	ϕ	q_2
q_2	ϕ	q_3
$*q_3$	q_3	q_3

step 4: check some string from the above language

$w = 0111$

$\delta(q_0, 0111) \vdash \delta(q_0, 111) \vdash \delta(\{q_0, q_1\}, 11)$

$\vdash \delta(\{q_0, q_1, q_2\}, 1) \vdash \delta(\{q_0, q_1, q_2, q_3\}, \epsilon)$

$\vdash \{q_0, q_1, q_2, q_3\} \cap F \neq \phi$

∴ The string is accepted by the automata.

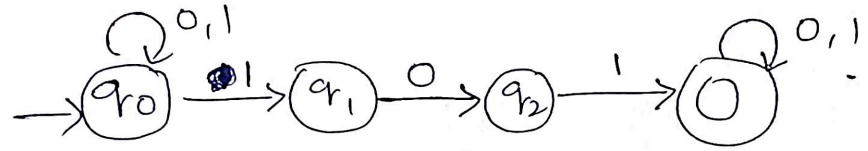
Step 5: represent the above with 5-tuple notation

$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, \{q_0\}, \{q_3\})$

→ Design NFA to accepts set of all strings containing 101 as substring.

Sol: step1: $L = \{101, 0101, 1010 \ldots\}$

step2: The transition diagram is

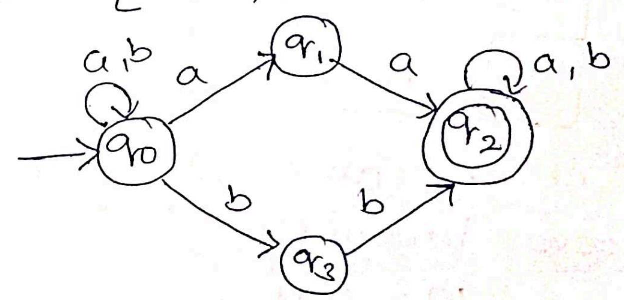


explain the remaining steps as explained in the before solution.

→ Design NFA accepting strings with a's & b's such that string containing two consecutive a's or two consecutive b's.

Sol: step1: $L = \{aa, bb, aaa, baa, bbb \ldots\}$

step2:

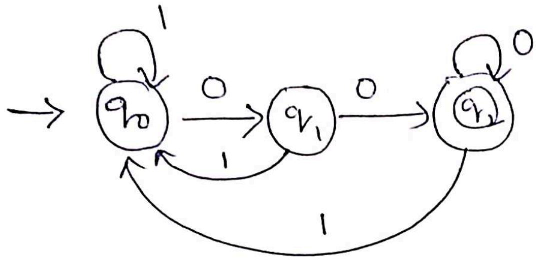


explain the remaining steps as mentioned in the solution of 1st problem.

→ Design DFA for the language all strings over {0,1} which are ending with 00

Sol:

$L = \{00, 100, 000, 1100 \dots\}$



δ	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
$*q_2$	q_2	q_0

check the acceptance of string

$w = 100$

$\delta(q_0, 100) \vdash \delta(q_0, 00)$

$\vdash \delta(q_1, 0)$

$\vdash \delta(q_2, \epsilon)$

$\vdash q_2 \cap F \neq \phi$

∴ The string is accepted.

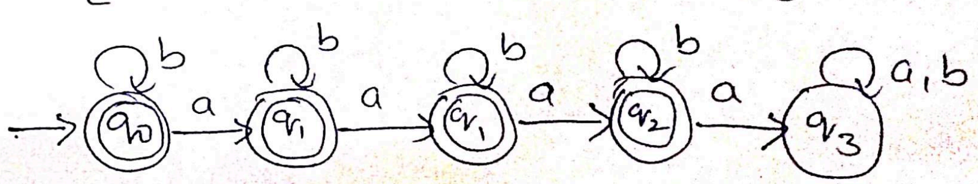
$M = (Q, \Sigma, \delta, q_0, F)$

$= (\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\})$

→ Design DFA for the following over {a,b}

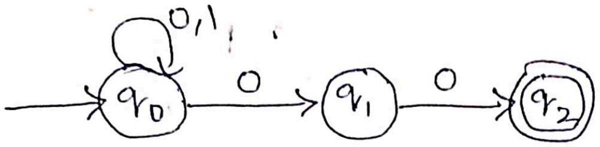
i> All strings containing not more than 3 a's.

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



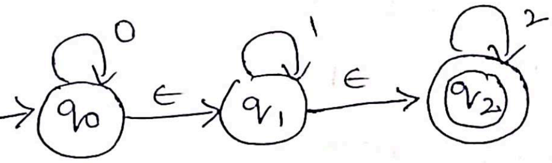
→ Design NFA for the language all strings over {0,1} which are ending with 00.

Sol: $L = \{00, 100, 000, \ldots\}$



→ Construct a NFA with epsilon, accepting a language consists a set of strings with any number of 0's followed by any number of 1's followed by any number of 2's over {0,1,2}

Sol: $L = \{\epsilon, 0, 1, 2, 01, 12, 02 \ldots\}$

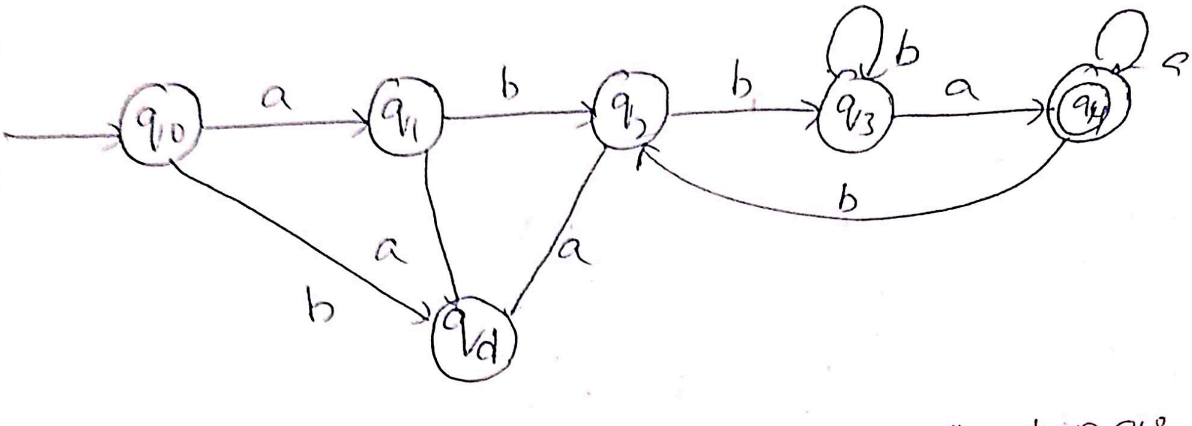


δ	0	1	2	ϵ
$\rightarrow q_0$	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
$*q_2$	ϕ	ϕ	q_2	ϕ

$M = (Q, \Sigma, \delta, q_0, F) = (\{q_0, q_1, q_2\}, \{0,1,2\}, \delta, \{q_0\}, \{q_2\})$

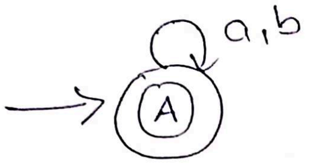
→ ii) All strings that has atleast two occurences of b between ~~any~~ Every two occurences of a.

L = {abba, abbabba - - - - -



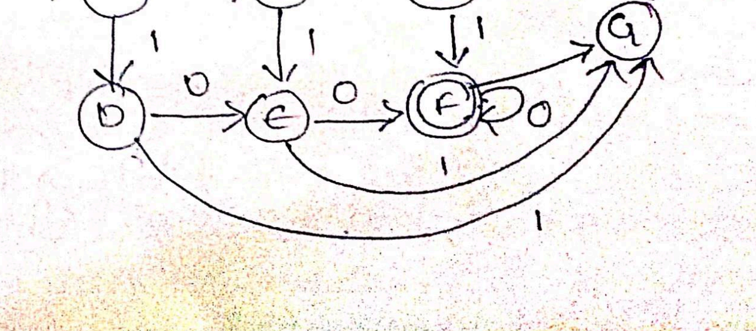
→ Construct DFA for the language all strings over {a,b} consisting of any number of a's & b's

L = {ε, a, b, ab, ba - - -}



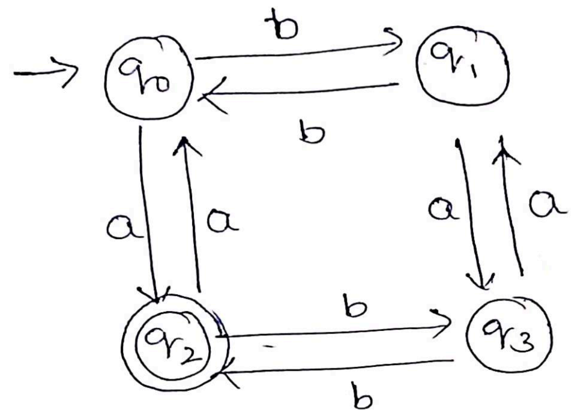
→ Construct DFA recognizing L = {w ∈ (0,1)* | w contains atleast two 0's & at most one 1}?

L = {00, 001, 010 - - -}



→ Design a DFA to accept odd number of a's and even number of b's where $\Sigma = \{a, b\}$. Show the acceptance of a string with an example? (3)

$L = \{a, abb, bab \ldots\}$



for w = abbaa check the acceptance

$\delta(q_0, abbaa) \vdash \delta(q_2, bbaa) \vdash \delta(q_3, baa)$

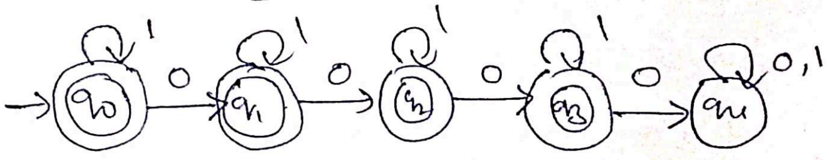
$\vdash \delta(q_2, aa) \vdash \delta(q_0, a) \vdash q_2 \in F$

∴ The string is accepted.

→ Design DFA for the following over {0, 1}

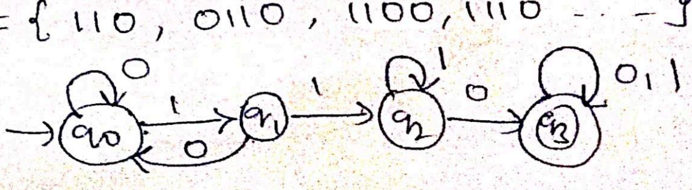
i> All strings containing not more than 3 0's

$L = \{\epsilon, 0, 1, 01, 10 \ldots\}$



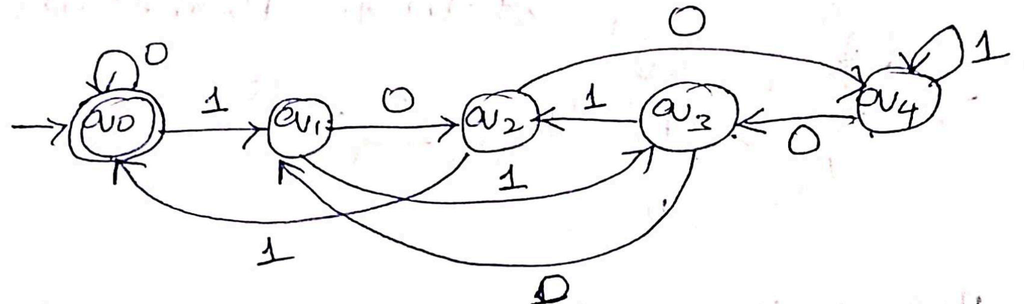
ii> All strings containing two consecutive one's followed by 0.

$L = \{110, 0110, 1100, 1110 \ldots\}$



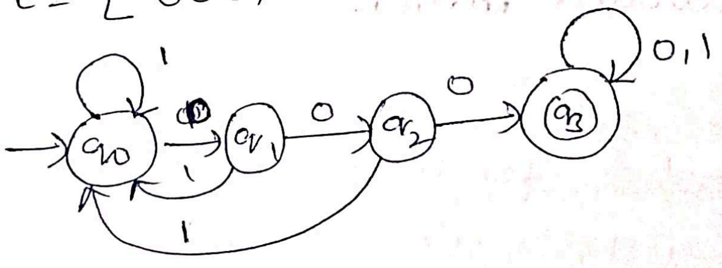
→ Design a DFA that accepts set of all integers (24) numbers represented as strings in binary that start with 1 & are multiples of 5.

$L = \{0, 01, 10, 11, 100, 101, \ldots\}$



→ Design FA to accept set of all strings over {0,1} containing 3 consecutive ones

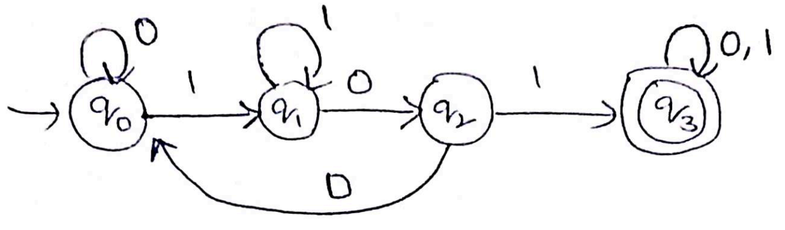
$L = \{000, 1000, 0001 \ldots\}$



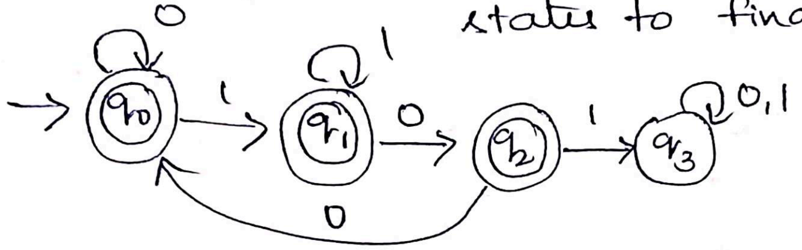
→ Design FA which accepts set of all strings not containing 101 as substring.

Construct the language for set of all strings containing 101 as substring.

$L = \{101, 0101, 1010 \ldots\}$



find the complement of above automata by changing the final states to non final & non final states to final states.

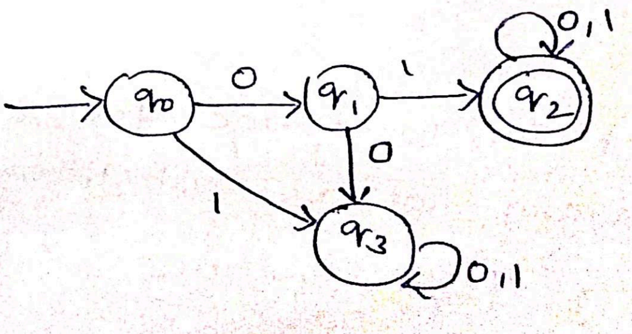


This is the automata which accepts all the strings that does not contain 101 as its substring.

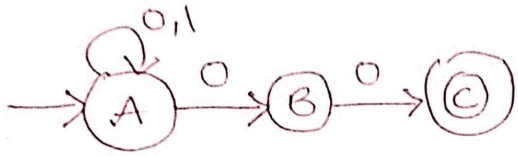
explain the remaining steps as mentioned earlier.

→ Design FA for the language all strings over {0, 1} which are starting with 01

L = {01, 010, 011 ----}



→ Construct equivalent DFA for the given NFA.



Sol: The transition table for the given NFA is

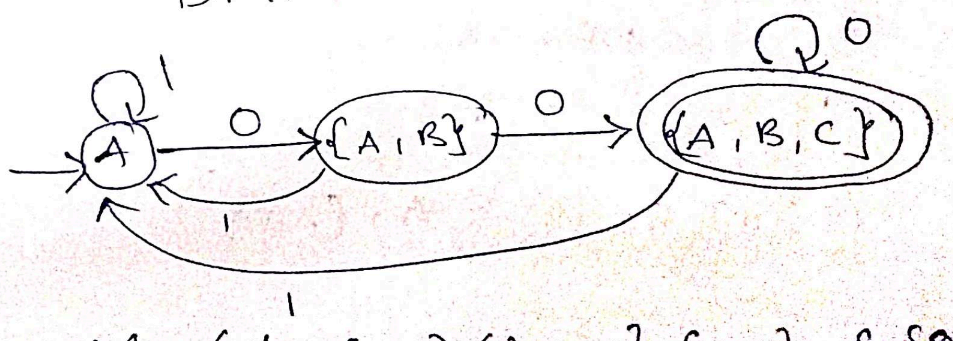
	0	1
→ A	{A, B}	A
B	C	ϕ
(C)	ϕ	ϕ

Construct the transition table for the equivalent DFA

	0	1.
→ A	{A, B}	A
{A, B}	{A, B, C}	A
* {A, B, C}	{A, B, C}	A

here, {A, B, C} is the final state as this is only the state which is having final state (C) of given NFA

The transition diagram of equivalent DFA M



M = ({A, {A, B}, {A, B, C}}, {0, 1}, δ, {A}, {A, B, C})

⇒ Convert the following NFA into DFA

Sol: The transition table for the given NFA is

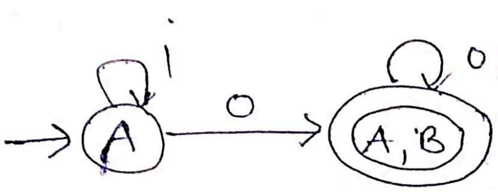
δ	0	1
$\rightarrow A$	$\{A,B\}$	A
B	B	B

Now construct transition table for the equivalent DFA

δ	0	1
$\rightarrow A$	$\{A,B\}$	A
$\{A,B\}$	$\{A,B\}$	$\{A,B\}$

here $\{A,B\}$ is the final state

The transition diagram for the equivalent DFA is



The 5-tuple Notation of equivalent DFA is

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

$$= (\{A,B\}, \{0,1\}, \delta, A, \{A,B\})$$

→ Convert the following NFA with ϵ - moves to DFA shown in figure 2. (6)

Sol: Step1: Find the ϵ-closures of A & B

ϵ-closure(A) = {A, B}

ϵ-closure(B) = {B}

Step2: Now the δ' transition on each input symbol is obtained as

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

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

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

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

$= \{A,B\}$

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

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

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

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

$= \{B\}$

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

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

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

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

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

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

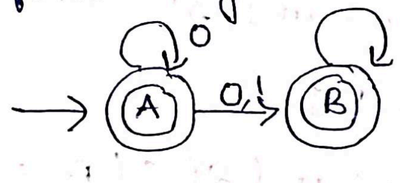
$= \{B\}$

step3: construct the transition table for the NFA by eliminating the ϵ-transitions.

δ	0	1
→A*	{A,B}	B
B*	ϕ	B

here A & B are the final states as the ϵ-closures of A & B contains the final state B

step 4: The NFA can be shown by the following transition diagram.



step5: The 5-tuple notation of equivalent NFA is M = ({A,B}, {0,1}, δ, {A}, {A,B})

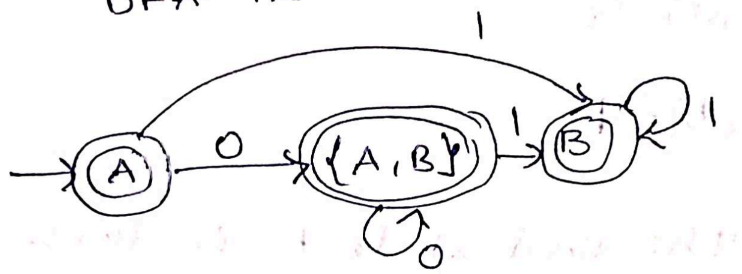
step6: Convert the above NFA into DFA using the NFA to DFA conversion algorithm

δ	0	1
→A*	{A,B}	B
{A,B}*	{A,B}	B
B*	ϕ	B

construct the transition table for the equivalent DFA

→ find the final states
A, {A,B}, B are the final states in the equivalent DFA. as they are the final states in the NFA.

→ The transition diagram for the equivalent DFA is



→ Represent the equivalent DFA with the 5-tuple notation.

$$M = (\{A, \{A,B\}, B\}, \{0,1\}, \delta, \{A\}, \{\{A\}, \{A,B\}, \{B\}\})$$

→ Convert the following NFA with ε-moves to equivalent DFA.

	a	b	ε
→P	∅	P	Q
Q	Q	∅	R
R	Q	P	∅

Sol: Step1: find the ε-closures of P, Q, R

ε-closure (P) = {P, Q, R}

" (Q) = {Q, R}

" (R) = {R}

step2 : $\delta'(P,a) = Q$

$\delta'(P,b) = P$

$\delta'(Q,a) = Q$

$\delta'(Q,b) = P$

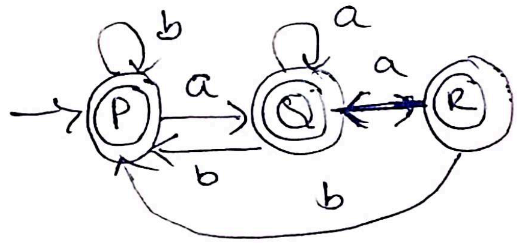
$\delta'(R,a) = Q$

$\delta'(R,b) = P$

find all the transitions over Σ by eliminating ϵ - transitions.

step3 : The final state R is there in ϵ-closures of P, Q, R so, all these are final states in the equivalent NFA.

step4 : construct transition ~~table~~ diagram for the equivalent NFA



step5 : construct transition table for the equivalent NFA

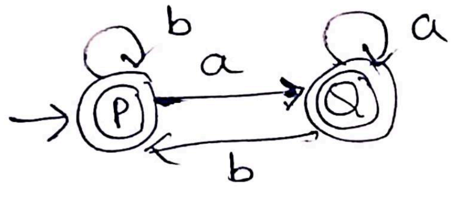
δ	a	b
→P	Q	P
Q	Q	P
R	Q	P

step 6: construct the equivalent DFA for the above NFA by using the NFA to DFA conversion algorithm

δ	a	b
→P	Q	P
Q	Q	P

here P, & Q are the final states in the equivalent DFA as they are final states in the NFA.

step 7: construct the transition diagram for the equivalent DFA for the given E-NFA



step 8: The 5-tuple representation of the equivalent DFA is

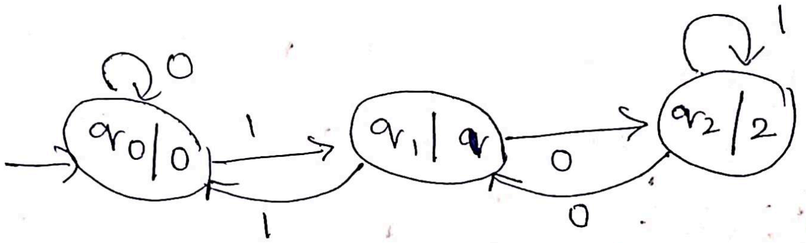
$M = (Q, \Sigma, \delta, q_0, F)$

$= (\{P, Q\}, \{a, b\}, \delta, \{P\}, \{P, Q\})$

→ Convert the moore m/c to determine residue mod 3 into Mealy m/c.

Sol: construct moore m/c to determine residue mod 3 where we consider $\Sigma = \{0, 1\}$

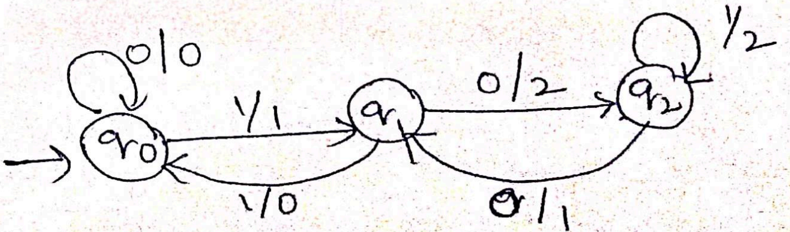
Q	0	1	O/P
→ q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2



Now construct equivalent Mealy m/c for the above moore m/c

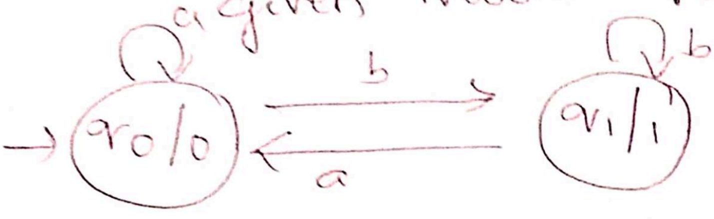
Q\Σ	0: state	0: O/P	1: state	1: O/P
→ q_0	q_0	0	q_1	1
q_1	q_2	2	q_0	0
q_2	q_1	1	q_2	2

The transition diagram is



→ construct equivalent Mealy M/c for the given moore M/c.

Sol:



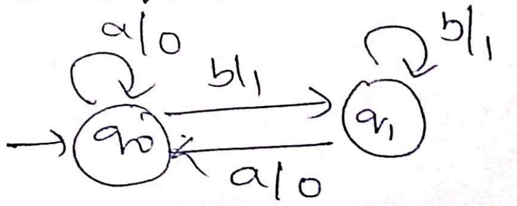
construct transition table

Q \ Σ	a	b	O/P
→ q0	q0	q1	0
q1	q0	q1	1

construct equivalent mealy M/c for the given moore M/C

Q \ Σ	a		b	
	Next state	O/P	Next state	O/P
q0	q0	0	q1	1
q1	q0	0	q1	1

The equivalent mealy M/c will be



→ Convert the following Mealy M/C into equivalent Moore M/C

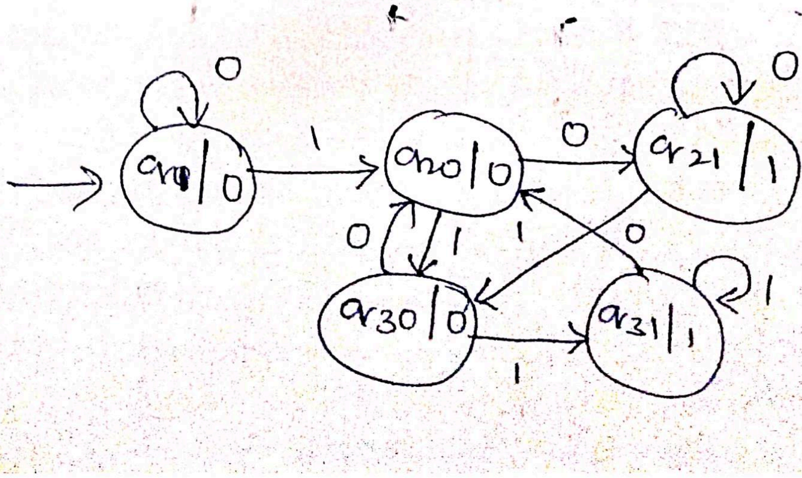
Sol:

	0		1	
	Next state	o/p	Next state	o/p
→ q_1	q_1	0	q_2	0
q_2	q_2	1	q_3	0
q_3	q_2	0	q_3	1

Sol: The state transition ~~diagram~~ table for the equivalent Moore M/C is

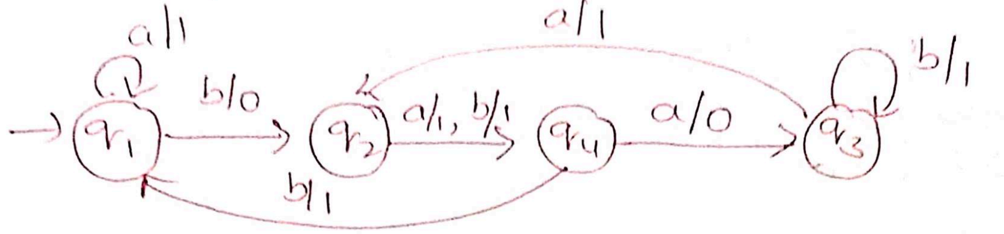
	0	1	o/p
q_1	q_1	q_{20}	0
q_{20}	q_{21}	q_{30}	0
q_{21}	q_{21}	q_{30}	1
q_{30}	q_{20}	q_{31}	0
q_{31}	q_{20}	q_{31}	1

The transition diagram for the equivalent Moore M/C is



→ Convert the following Mealy machine into equivalent Moore Machine.

i>



Sol: Construct transition table for the given Mealy M/C

	a		b	
	Next state	output	Next state	output
→ q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

here states q_1, q_4 gives only one output i.e; 1 so we retain states q_1 & q_4. But states q_2, q_3 are producing different outputs (1 & 0). So we divide q_2 into q_{20}, q_{21} & q_3 into q_{30} & q_{31}

→ The state table for the equivalent Moore M/C is

	Next state		output
	a	b	
q_1	q_1	q_{20}	1
q_{20}	q_4	q_4	0
q_{21}	q_4	q_4	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_{30}	q_1	1