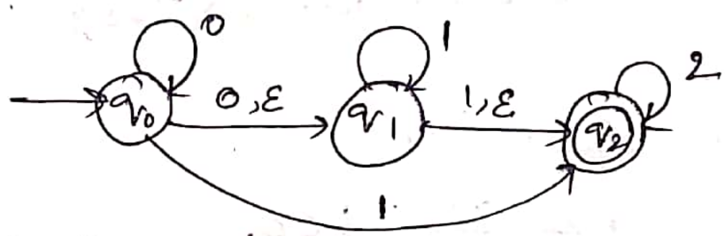
Finite AutoMata

NFA with ε Transitions : (NFA with ε-Transitions) :-

The ε-transitions in NFA are given in order to move from one state to another without having any symbol from input set Σ.

Example:

Consider an Example of NFA with ε as:



In this NFA with ε,

* q_0 is start with input '0' one can be either in q_0 or in q_1
* If we get at the start a symbol '1' then the ε-move we can change from q_0 to q_1
* On the other hand from start state q_0, with input '1' we can reach to 'q_2'
* So, this is not definite with '1' input, we will be in state q_1 or q_2. So it is called Nondeterministic Finite Automata.
* Since there are some 'ε' moves by which we can simply change the states from one to another state. Hence it is called "NFA with ε".

Acceptance of Language:

The language 'L' accepted by NFA with ε, denoted by $M = (Q, \Sigma, \delta, q_0, F)$ can be defined as

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a NFA with ε

Where Q = Finite set of States

Σ = Input set

δ = Transition or Mapping function

Transition from $Q \times \{\Sigma \cup \varepsilon\}$ to 2^Q.

q_0 = Initial State

F = Final State $F \in Q$.

$$\boxed{\therefore \delta : Q \times \{\Sigma \cup \varepsilon\} \rightarrow 2^Q}$$

Problems on NFA with ε:

① Construct NFA with ε which accepts a language consisting the strings of any no. of a's followed by any no. of b's, followed by any no. of c's.

Sol:- * Any No. of a's or b's or c's means zero & more in number.

* There can be zero or more a's followed by zero & more b's followed by zero or more c's.

NFA - ε :-

Transition Table:

State \ Inputs	a	b	c	ε
q_0	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

String acceptance: aabbcc

$\delta(q_0, aabbcc)$

$\delta(q_0, bbcc) \Rightarrow \delta(q_0, \varepsilon bbcc)$

$\delta(q_1, bbcc)$

$\delta(q_1, bcc)$

$\delta(q_1, cc)$

$\delta(q_1, \varepsilon cc)$

$\delta(q_2, cc)$

$\delta(q_2, c)$

$\delta(q_2, \varepsilon)$

We reach accept/Final state.

Definition of ε- closure:

The ε-closure (P) is a set of all states which are reachable from state P on ε-transitions such that

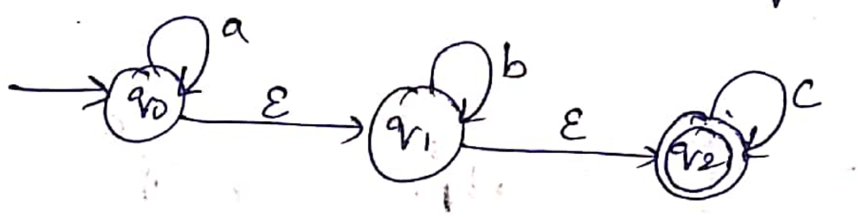
i) ε-closure (P) = p where p ∈ Q

ii) If there exists ε-closure (p) = {q} and δ(q, ε) = r then

ε- closure (p) = {q, r}

① Example :- $\boxed{\text{* Every state on } \varepsilon \text{ goes to itself.}}$

Find ε-closure for following NFA with ε



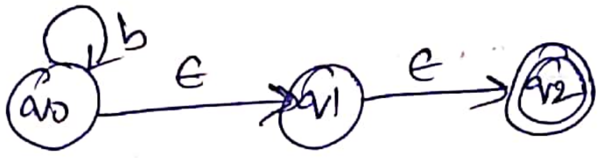
Solution:

ε-closure (q0) = {q0, q1, q2} ⇒ Self state + ε-reachable states.

ε-closure (q1) = {q1, q2}

Means 'q1' is itself self state and q2 is a state obtained from q1 with ε input..

ε-closure (q2) = {q2}

(2)  → ϵ-closure of States = ?

Sol:

Definition:

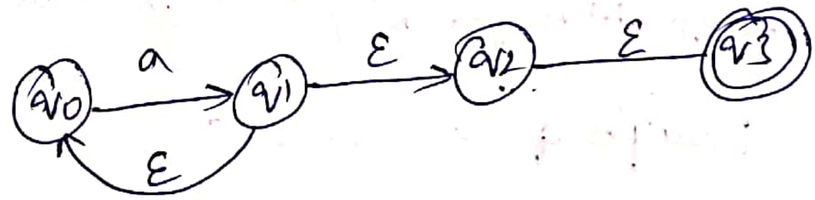
The ϵ-closure of a state 'q_0' is defined as the set of all state 'P'. Such that there is a path from $q_0 \to P$ with lable 'ϵ'

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

ϵ-closure of $q_1 = \{q_1, q_2\}$

ϵ-closure of $q_2 = \{q_2\}$

(3) Write ϵ-closures states for the following Machine?

Sol: 

Sol:

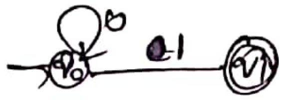
ϵ-closure of $(q_0) = \{q_0\}$

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

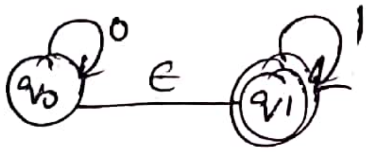
ϵ-closure$(q_3) = \{q_3\}$

NFA with ε transition Example:

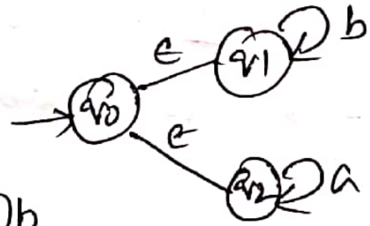
① 0^*1 ⇒



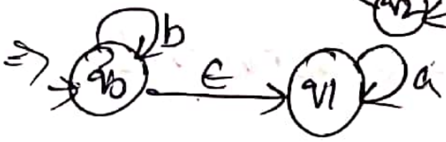
② 0^*1^* ⇒



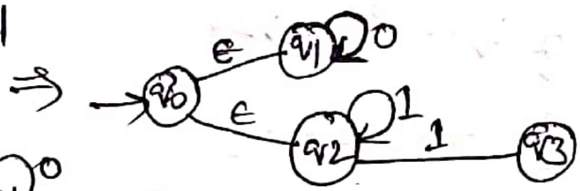
③ b^*/a^* ⇒



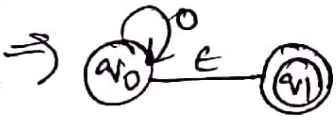
④ b^*a^* ⇒



⑤ $0^*/1^*1$ ⇒



⑥ 0^*. ⇒

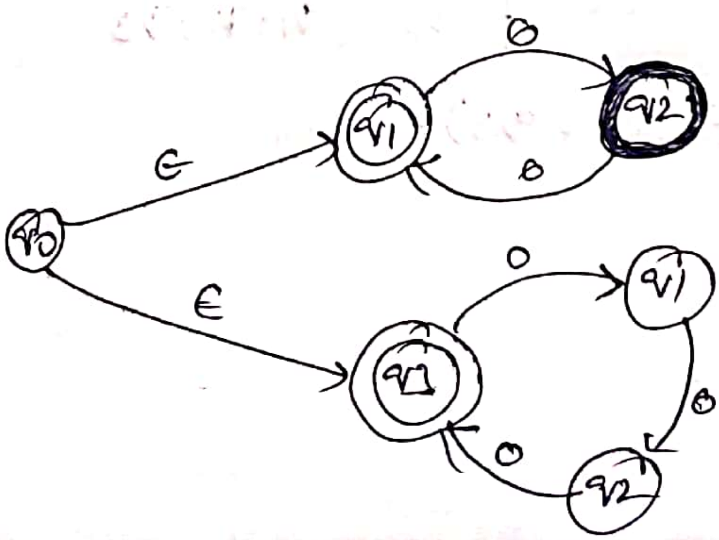


Q: Design NFA for $L = \{0^k / k$ is Multiples of $2/3\}$

Soln Multiples of 2 ⇒ $0^2, 0^4, 0^6, \ldots$

Multiply of 3 ⇒ $0^3, 0^6, 0^9, \ldots$

Multiples of 2 / Multiples of 3



} 2 Multiply.

} 3 Multiply

Conversions & Equivalence :-

Conversion is of 3 types.

① NFA with ε to NFA without 'ε'

2) NFA to DFA

3) NFA with ε to DFA.

"NFA with ε can be converted to NFA without ε and then NFA without ε can be converted to DFA.

① NFA with ε to NFA without ε:

In this method remove all the ε transitions from given NFA. The method will be

① Find out all the ε-transitions from each state Q. That will be called as $\{q_i\}$ where $q_i \in Q$.

② Then δ' transitions can be obtained. The δ' transitions means an ε-closure on δ moves.

③ Step-2 is repeated for each input symbol and for each state of given NFA.

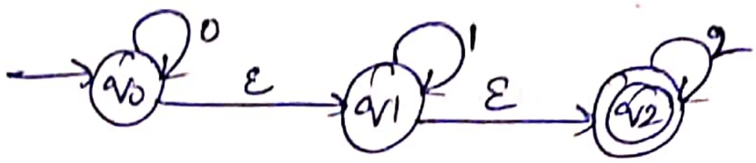
④ Using the resultant states the transition table for equivalent NFA without ε can be built.

Rule for Conversion : $\delta'(q, a) = \delta(\hat{\delta}(q, \varepsilon), a)$

where $\hat{\delta}(q, \varepsilon) = \varepsilon\text{-closure}(q)$.

Problems

① Convert the given NFA with ε to NFA without ε



Sol: Let NFA with ε transition is denoted by M

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

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

Let NFA without ε is denoted by M'

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

$$\delta' = Q \times \Sigma \rightarrow 2^Q.$$

ε-closures of each state :-

a) $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$
b) $\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$
c) $\epsilon\text{-closure}(q_2) = \{q_2\}$.

Here ε-closure (q_0) means with null point (no i/p symbol), we can reach to q_0, q_1, and q_2. Similarly Remaining.

Now we will obtain δ' transitions for each state on each input symbol. Those are...

$\delta'(q_0, 0)$	$\delta'(q_1, 0)$	$\delta'(q_2, 0)$
$\delta'(q_0, 1)$	$\delta'(q_1, 1)$	$\delta'(q_2, 1)$
$\delta(q_0, 2)$	$\delta'(q_1, 2)$	$\delta(q_2, 2)$.

Now we will obtain δ' transitions for each state on each input symbol.

$\rightarrow \delta'(q_0,0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0,\varepsilon),0))$

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

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

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

$= \varepsilon\text{-closure}(q_0 \cup \emptyset \cup \emptyset)$

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

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

$\rightarrow \delta'(q_0,1)$:

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

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

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

$= \varepsilon\text{-closure}(\emptyset \cup q_1 \cup \emptyset)$

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

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

$\rightarrow \delta'(q_1,0)$:-

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

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

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

$= \varepsilon\text{-closure}(\emptyset \cup \emptyset) = \varepsilon\text{-closure}(\emptyset) = \{\emptyset\}$ //.

→ $\delta'(q_1, 1)$

$$\delta'(q_1,1) = \varepsilon\text{-Closure}(\delta(\hat{\delta}(q_1,\varepsilon),1))$$
$$= \varepsilon\text{-closure}(\delta(q_1,q_2),1)$$
$$= \varepsilon\text{-closure}(\delta(q_1,1) \cup \delta(q_2,1))$$
$$= \varepsilon\text{-Closure}(q_1 \cup \emptyset)$$
$$= \varepsilon\text{-closure}(q_1)$$
$$= \{q_1, q_2\}$$

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

→ $\delta'(q_1,2)$:

$$\delta'(q_1,2) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1,\varepsilon),2))$$
$$= \varepsilon\text{-closure}(\delta(q_1,q_2),2)$$
$$= \varepsilon\text{-closure}(\delta(q_1,2) \cup \delta(q_2,2))$$
$$= \varepsilon\text{-Closure}(\emptyset \cup q_2)$$
$$= \varepsilon\text{-Closure}(q_2)$$

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

$\delta'(q_0,2)$

$$\delta'(q_0,2) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0,\varepsilon),2))$$
$$= \varepsilon\text{-closure}(\delta(q_0,q_1,q_2),2)$$
$$= \varepsilon\text{-closure}(\delta(q_0,2) \cup \delta(q_1,2) \cup \delta(q_2,2))$$
$$= \varepsilon\text{-closure}(\emptyset \cup \emptyset \cup q_2)$$
$$= \varepsilon\text{-closure}(q_2)$$

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

→ $\delta'(q_2,0)$:

$$\delta'(q_2,0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_2,\varepsilon),0))$$
$$= \varepsilon\text{-closure}(\delta(q_2),0)$$
$$= \varepsilon\text{-closure}(\cancel{\delta(q_1,0)\cup}\,\delta(q_2,0))$$
$$= \varepsilon\text{-closure}(\delta(q_2,0))$$
$$= \varepsilon\text{-closure}(\emptyset) = \emptyset$$ //.

$\delta'(q_2, 1)$

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

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

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

$= \varepsilon\text{-closure}(\emptyset)$

$= \emptyset$ //

$\delta'(q_2, 2)$:-

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

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

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

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

$= \{q_2\}$

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

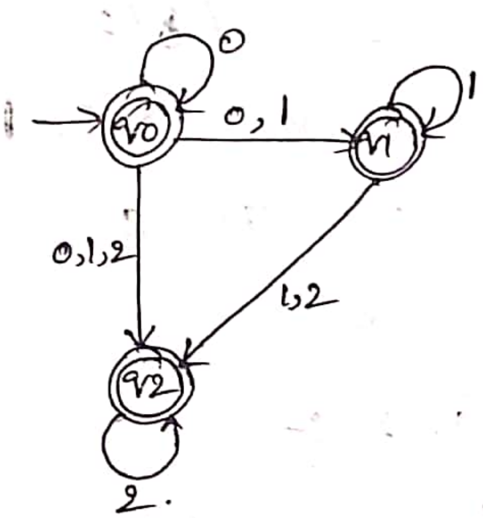
Summarize the all Computed δ' transitions:

1) $\delta'(q_0,0) = \{q_0,q_1,q_2\}$	$\delta'(q_0,1) = \{q_1,q_2\}$	$\delta'(q_0,2) = \{q_2\}$
2) $\delta'(q_1,0) = \emptyset$	$\delta'(q_1,1) = \{q_1,q_2\}$	$\delta'(q_1,2) = \{q_2\}$
3) $\delta'(q_2,0) = \emptyset$	$\delta'(q_2,1) = \{\emptyset\}$.	$\delta(q_2,2) = \{q_2\}$.

Transition Table :-

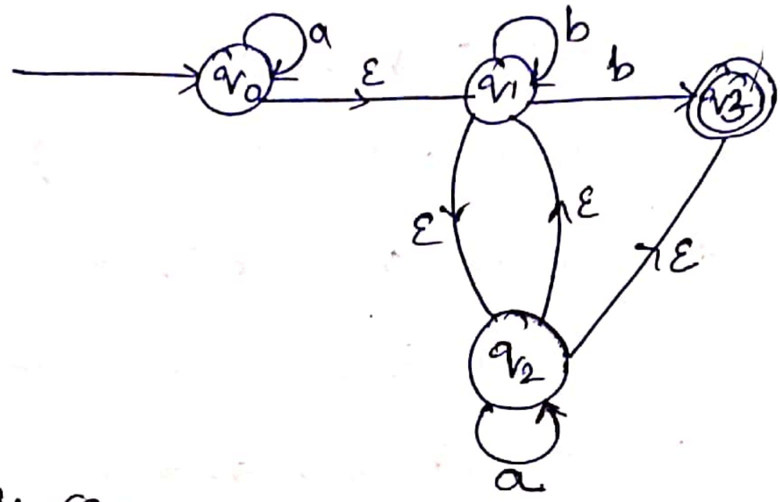
Inputs / States	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$

NFA without ε



Here q_0, q_1 & q_2 is a Final State, because ε-closure(q_0); ε-Closure(q_1), ε-Closure(q_2) Contains Final State.

② Convert NFA-ε into without ε.



Soln Step①:

ε-closure of $q_0 = \{q_0, q_1, q_2, q_3\}$

ε-closure$(q_1) = \{q_1, q_2, q_3\}$

ε-closure$(q_2) = \{q_2, q_1, q_3\}$

ε-closure$(q_3) = \{q_3\}$.

Transition Table

	a	b	ϕ	ε
q_0	q_0	∅	∅	q_1
q_1	~~∅~~	$\{q_1, q_3\}$.	∅	q_2
q_2	q_2	∅	∅	$\{q_1, q_3\}$
q_3	∅	∅	∅	∅

According to Conversion Formula.

The transitions as below.

Step②:

Finding of below Values:

$\delta'(q_0, a) = ?$	$\delta'(q_2, a) = ?$
$\delta'(q_0, b) = ?$	$\delta'(q_2, b) = ?$
$\delta'(q_1, a) = ?$	$\delta'(q_3, a) = ?$
$\delta'(q_1, b) = ?$	$\delta'(q_3, b) = ?$

Finding each value one by one.

* $\delta'(q_0, a) = \varepsilon\text{-closure}\left(\delta(\hat{\delta}(q_0, \epsilon), a)\right)$

$= \varepsilon\text{-closure}\left(\delta(q_0, \underset{q_2,q_3}{q_1}), a\right)$

$= \varepsilon\text{-closure}\left(\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_3, a)\right)$

$= \varepsilon\text{-closure}(q_0 \cup q_2)$

$= \varepsilon\text{-closure}(q_0) \cup \varepsilon\text{-closure}(q_2)$

$= \{q_0, \underset{q_2,q_3}{q_1}\} \cup \{q_1, q_2, q_3\}$

$$\boxed{\delta'(q_0, a) = \{q_0, q_1, q_2, q_3\}}$$

* $\delta'(q_0, b)$:

$= \varepsilon\text{-closure}\left(\delta(\hat{\delta}(q_0, \epsilon), b)\right)$

$= \varepsilon\text{-closure}\left(\delta(q_0, q_1), b\right)$

$= \varepsilon\text{-closure}\left(\delta(q_0, b) \cup \delta(q_1, b)\right)$

$= \varepsilon\text{-closure}(\emptyset \cup q_3)$

$= \varepsilon\text{-closure}(q_3)$

$= \{q_3\}$

$$\boxed{\therefore \delta'(q_0, b) = \{q_3\}.}$$

* $\delta'(q_1, a)$.

$= \varepsilon\text{-closure}\left(\delta(\hat{\delta}(q_1, \epsilon), a)\right)$

$= \varepsilon\text{-closure}\left(\delta(q_1, q_2, q_3), a\right)$

$= \varepsilon\text{-closure}\left(\delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_3, a)\right)$

$= \varepsilon\text{-closure}(q_1 \cup q_2 \cup \emptyset)$

$= \varepsilon\text{-closure}(q_1) \cup \varepsilon\text{-closure}(q_2)$

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

$$\boxed{\delta'(q_1, a) = \{q_1, q_2, q_3\}}$$

* $\delta'(q_1, b)$:

$= \varepsilon\text{-closure}\left(\delta(\hat{\delta}(q_1, \epsilon), b)\right)$

$= \varepsilon\text{-closure}\left(\delta(q_1, q_2, q_3), b\right)$

$= \varepsilon\text{-closure}\left(\delta(q_1, b) \cup \delta(q_2, b) \cup \delta(q_3, b)\right)$

$= \varepsilon\text{-closure}(q_3 \cup \emptyset \cup \emptyset) \Rightarrow \varepsilon\text{-closure}(q_3) = \{q_3\}$

$$\boxed{\therefore \delta'(q_1, b) = \{q_3\}}$$

* $\delta'(q_2, a)$:

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_2, \varepsilon), a))$

$= \varepsilon\text{-closure}(\delta(q_1, q_2, q_3), a))$

$= \varepsilon\text{-closure}(\delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_3, a))$

$= \varepsilon\text{-closure}(q_1 \cup q_2 \cup \phi)$

$= \varepsilon\text{-closure}(q_1 \cup q_2)$

$= \varepsilon\text{-closure}(q_1) \cup \varepsilon\text{-closure}(q_2)$

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

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

$\boxed{\delta'(q_2, a) = \{q_1, q_2, q_3\}}$

* $\delta'(q_2, b)$: ?

$\Rightarrow \varepsilon\text{-closure}(\delta(\hat{\delta}(q_2, \varepsilon), b))$

$= \varepsilon\text{-closure}(\delta(q_1, q_2, q_3), b)$

$= \varepsilon\text{-closure}(\delta(q_1, b) \cup \delta(q_2, b) \cup \delta(q_3, b))$

$= \varepsilon\text{-closure}(q_3 \cup \phi \cup \phi)$

$= \varepsilon\text{-closure}(q_3)$

$= \boxed{\delta'(q_2, b) = \{q_3\}}$

* $\delta'(q_3, a)$: ?

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_3, \varepsilon), a))$

$= \varepsilon\text{-closure}(\delta(q_3, a))$

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

$= \phi$.

* $\delta'(q_3, b)$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_3, \varepsilon), b))$

$= \varepsilon\text{-closure}(\delta(q_3, b))$

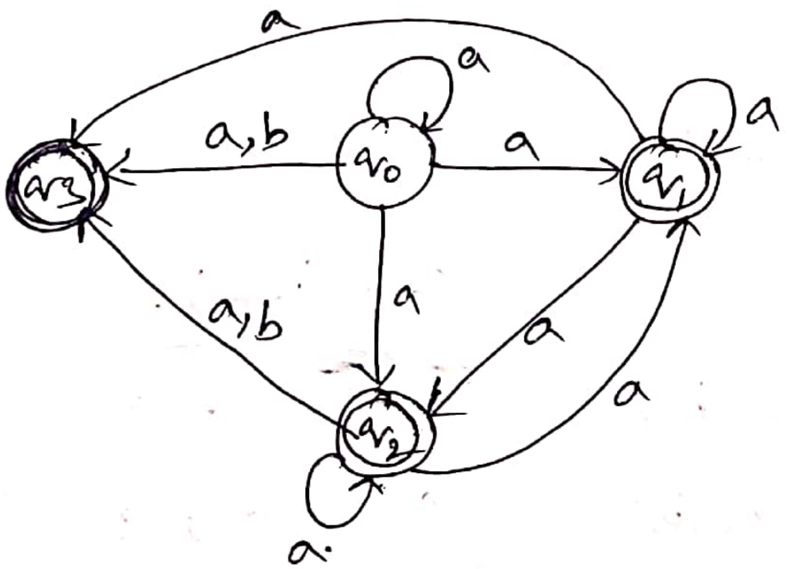
$= \varepsilon\text{-closure}(\phi) = \phi$.

Transition Table for NFA without ε

	a	b
q_0	$\{q_0, q_1, q_2, q_3\}$	$\{q_3\}$
q_1	$\{q_1, q_2, q_3\}$	$\{q_3\}$
q_2	$\{q_1, q_2, q_3\}$	$\{q_3\}$
q_3	ϕ	ϕ

	a	b
q_0	$\{q_0, q_1, q_2, q_3\}$	$\{q_3\}$
q_1	$\{q_1, q_2, q_3\}$	$\{q_3\}$
q_2	$\{q_1, q_2, q_3\}$	$\{q_3\}$
q_3	ϕ	ϕ

NFA-without ε Diagram:



q_1, q_2, q_3 — Final States.

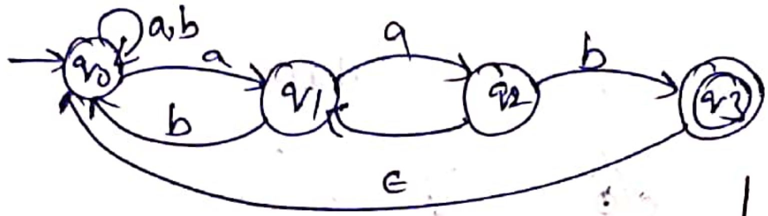
Final State Declaration:

In given problem q_3 is final state, This 'q_3' included in ε-closure of (q_1), ε-closure (q_2), ε-closure (q_3).

So the all q_1, q_2, q_3 will become Final States.

Scanned by CamScanner

③ $L = \{x \in \{a,b\} / x$ end with $aab\}$



ϵ-closure$(q_0) = q_0$

ϵ-closure$(q_1) = q_1$

ϵ-closure$(q_2) = q_2$

ϵ-closure$(q_3) = \{q_0, q_3\}$

	a	b	ϵ
$\to q_0$	$\{q_0,q_1\}$	$\{q_0\}$	ϕ
q_1	q_2	q_0	ϕ
q_2	q_1	q_3	ϕ
q_3	ϕ	ϕ	$\{q_0\}$.

String : abaab checking

$\to \hat{\delta}(q_0,\epsilon) = q_0$

$\to \hat{\delta}(q_0,a) = \epsilon\text{-closure}\ \delta(\hat{\delta}(q_0,\epsilon), a) = \delta(q_0,a) = \{q_0,q_2,q_3\}$.

$\to \hat{\delta}(q_0,ab) = \epsilon\text{-closure}\ \delta((q_0,\overset{q_3}{q_2}),b) = \delta(q_0,b) \cup \delta(q_2,b) \cup \delta(q_3,b)$

$= q_0 \cup q_3 \cup \phi$

$= \{q_0, q_3\}$

$\to \hat{\delta}(q_0,aba) = \epsilon\text{-closure}(\delta((q_0,q_3),a)) = \delta(q_0,a) \cup \delta(q_3,a)$

$= \{q_0,q_1\} \cup \phi = \{q_0,q_1\}$

$\to \hat{\delta}(q_0,abaa) = \epsilon\text{-closure}(\delta((q_0,q_1),a) = \delta(q_0,a) \cup \delta(q_1,a)$

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

$\to \hat{\delta}(q_0,abaab) = \epsilon\text{-closure}(\delta((q_0,q_1,q_2),b))$

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

$= \{q_0\} \cup q_0 \cup q_3$

$= \{q_0, q_3\}$.

$\epsilon\text{-closure}(\{q_0,q_3\}) = \epsilon\text{-closure}(q_0) \cup \epsilon\text{-closure}(q_3)$

$= \{q_0 \cup q_0, q_3\} = \{q_0, q_3\}$

Method: 2 $\hat{\delta}(q_0, abaab)$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, \varepsilon), abaab))$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, a), baab))$.

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, q_1), baab))$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, b) \cup \hat{\delta}(q_1, b)), aab)$

$= \varepsilon\text{-closure}(\delta(q_0 \cup q_0), aab)$

$= \varepsilon\text{-closure}(\delta(q_0, aab))$

$= \varepsilon\text{closure}(\delta(\hat{\delta}(q_0, a), ab))$

$= \varepsilon\text{-closure}(\delta(q_0, q_1), ab)$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, a) \cup \hat{\delta}(q_1, a)), b)$

$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), b)$

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

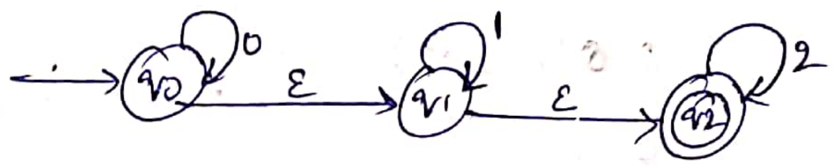
$= \varepsilon\text{-closure}(q_0 \cup q_0 \cup q_3)$

$= \varepsilon\text{-closure}(q_0) \cup \varepsilon\text{-closure}(q_0) \cup \varepsilon\text{-closure}(q_3)$

$= \{q_0, q_3\}$ //

Conversion of NFA with ε-transitions into DFA :-

Ex: 1



Sol:

$\hat{\delta}(q_0,\varepsilon) = \varepsilon\text{-closure}(q_0) = \{q_0,q_1,q_2\} = A$

$\hat{\delta}(q_1,\varepsilon) = \varepsilon\text{-closure}(q_1) = \{q_1,q_2\} = B$

$\hat{\delta}(q_2,\varepsilon) = \varepsilon\text{-closure}(q_2) = \{q_2\} = C.$

$\delta'(A,0)$

$* \ \delta'(q_0,0) = \varepsilon\text{-closure}(\delta(\overbrace{\hat{\delta}(q_0,\varepsilon)}^{A}, 0))$

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

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

$= \varepsilon\text{-closure}(q_0 \cup \emptyset \cup \emptyset)$

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

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

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

$\delta'(A,1)$:-

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

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0,\varepsilon),1))$

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

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

$= \varepsilon\text{-closure}(\emptyset \cup q_1 \cup \emptyset)$

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

$= \{q_1,q_2\} \longrightarrow B$

$\delta'(A,2)$:-

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(A,\varepsilon),2))$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0,\varepsilon),2))$

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

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

$= \varepsilon\text{-closure}(\emptyset \cup \emptyset \cup q_2)$

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

$= \{q_2\} \longrightarrow C$

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

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1,\varepsilon),0))$

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

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

$= \varepsilon\text{-closure}(\emptyset \cup \emptyset)$

$= \varepsilon\text{-closure}(\emptyset)$

$\therefore \delta'(B,0) = \emptyset \Rightarrow D$

* $\delta'(B,1)$:

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

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1,\varepsilon),1))$

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

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

$= \varepsilon\text{-closure}(q_1 \cup \emptyset)$

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

$= \{q_1, q_2\}$

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

* $\delta'(B,2)$:

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(B,\varepsilon),2))$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1,\varepsilon),2))$

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

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

$= \varepsilon\text{-closure}(\emptyset \cup q_2)$

$= \varepsilon\text{-closure}(q_2) = \{q_2\} \Rightarrow C$

* $\delta'(C,0)$

$= \varepsilon\text{closure}(\delta(\hat{\delta}(C,\varepsilon),0))$

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

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

$= \varepsilon\text{-closure}(\emptyset) = \emptyset \rightarrow D$

* $\delta'(C,1)$

$= \varepsilon\text{-closure}(\delta(\hat{\delta}(C,\varepsilon),1))$

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

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

$= \varepsilon\text{-closure}(\emptyset) = \emptyset \Rightarrow D.$

* $\delta'(C,2)$

$\Rightarrow \varepsilon\text{-closure}(\delta(\hat{\delta}(C,\varepsilon),2))$

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

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

$= \varepsilon\text{-closure}(q_2) = \{q_2\} \Rightarrow C.$

$\therefore$

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

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

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

$\delta'(B,0) = \emptyset = D$

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

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

$\delta'(C,0) = \emptyset = D$

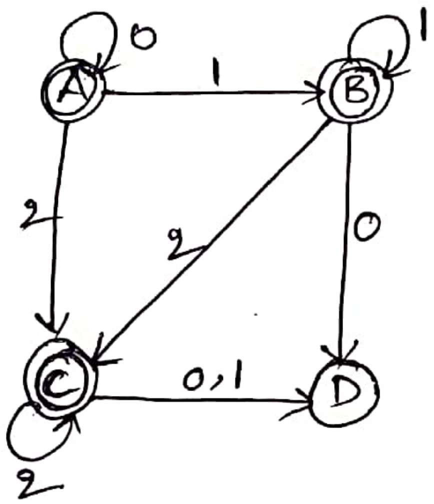
$\delta'(C,1) = \emptyset = D$

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

Transition Table:

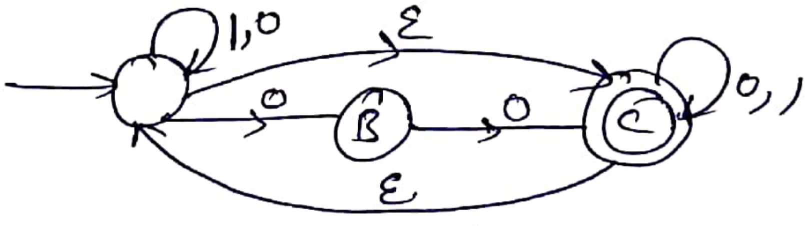
	0	1	2
A	A	B	C
B	D	B	C
C	D	D	C

Transition Diagram

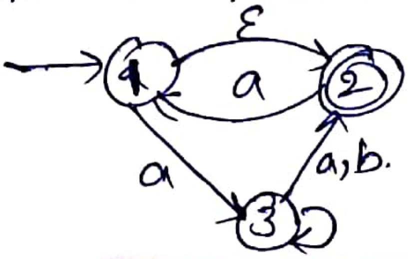


Example Solving Problems: **Assessment**: Construct DFA for the following.

1)



2) Draw an equivalent Deterministic Finite Automaton for the following Automaton.

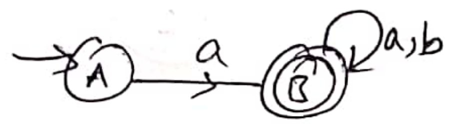


Conversion From NFA to DFA :-

Subset Construction:

① $\Sigma = \{a, b\}$; $L = \{$Start with 'a'$\}$.

NFA :



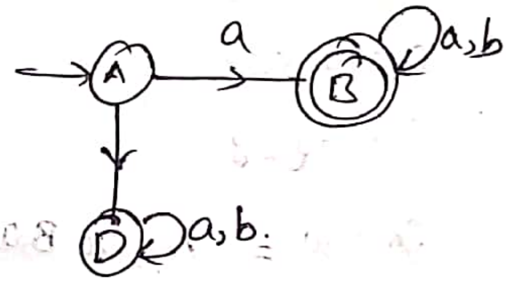
	a	b
→A	B	∅
*B	B	B

DFA :-

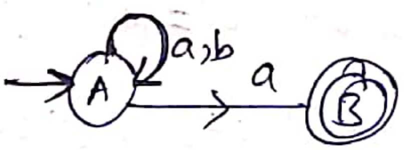
	a	b
→A	B	D
B	B	B
D	D	D

$\delta(A, a) = B$
$\delta(A, b) = \emptyset$
$\delta(B, a) = B$
$\delta(B, b) = B.$

∴



② $L = \{$ends with an 'a'$\}$.



→ $\delta(A, a) = \{A, B\}$
→ $\delta(A, b) = \{A\}$
→ $\delta(B, a) = \emptyset$
→ $\delta(B, b) = \emptyset.$

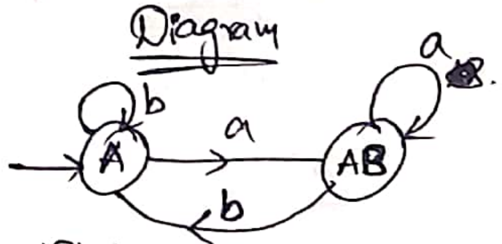
→ $\delta([A,B], a) = \delta(A, a) \cup \delta(B, a)$
$= \{A, B\} \cup \{\emptyset\} = \{A, B\}$

→ $\delta([A,B], b) = \delta(A, b) \cup \delta(B, b)$
$= \{A\} \cup \delta(\emptyset)$
$= \{A\}.$

Transition ~~Diagram~~ Table

	a	b
[A]	{A,B}	{A}
[B]	∅	∅
[AB]	{A,B}	{A}.

Diagram

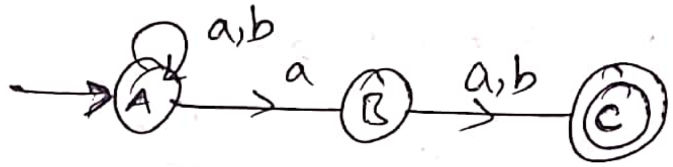


'B' is never reachable, No Need of thinking about 'B'-State.

(3) Conversion of NFA to DFA for the "All strings in which second symbol from RHS is 'a'".

Sol: $\Sigma = \{a, b\}$.

NFA:



$\rightarrow \delta(A, a) = \{A, B\}$

$\delta(A, b) = \{A\}$

$\delta(B, a) = \{C\}$

$\delta(B, b) = \{C\}$

$\delta(C, a) = \{\emptyset\}$

$\delta(C, b) = \{\emptyset\}$.

$\delta(\{A, B\}, a) = \delta(A, a) \cup \delta(B, a) = \{A, B\} \cup \{C\} = \{A, B, C\}$

$\delta(\{A, B\}, b) = \delta(A, b) \cup \delta(B, b) = \{A\} \cup \{C\} = \{A, C\}$.

$\delta(\{A, B, C\}, a) = \delta(A, a) \cup \delta(B, a) \cup \delta(C, a) = \{A, B\} \cup \{C\} \cup \{\emptyset\} = \{A, B, C\}$.

$\delta(\{A, B, C\}, b) = \delta(A, b) \cup \delta(B, b) \cup \delta(C, b) = \{A\} \cup \{C\} \cup \{\emptyset\} = \{A, C\}$

$\delta((A, C), a) = \delta(A, a) \cup \delta(C, a) = \{A, B\} \cup \{\emptyset\} = \{A, B\}$.

$\delta((A, C), b) = \delta(A, b) \cup \delta(C, b) = \{A\} \cup \{\emptyset\} = \{A\}$.

NFA

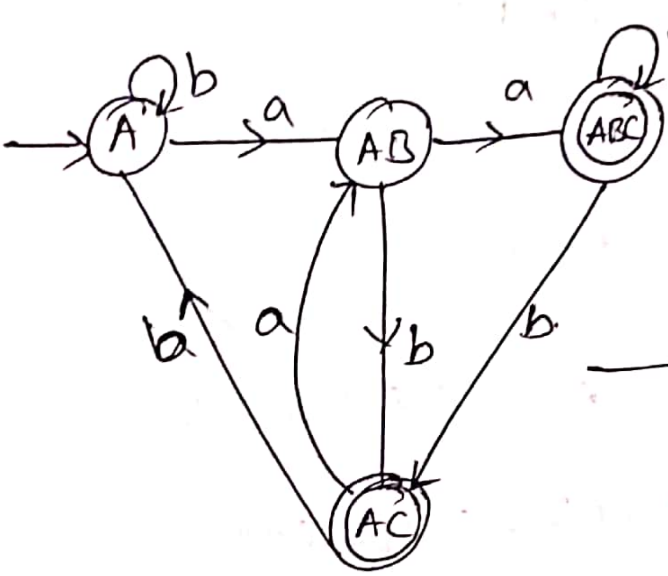
	a	b
→ A	{A B}	{A}
B	{C}	{C}
(C)	{∅}	{∅}

⟹

DFA

	a	b
→ A	{A, B}	{A}
{A B}	{A, B, C}	{A C}
({A, B, C})	{A, B, C}	{A, C}
({A, C})	{A, B}	{A}

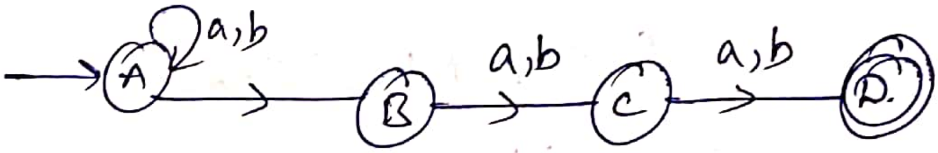
The final state 'C' is included in which state (C) included consider those as final state



— Final DFA Diagram.

(4) NFA to DFA

"All strings in which 3rd symbol from RHS is 'a'."



NFA

	a	b
→A	{A,B}	{A}
B	{C}	{C}
C	{D}	{D}
D	{∅}	{∅}

⟹

DFA

$\delta(\{A,B\}, a) = \delta(A,a) \cup \delta(B,a) = \{A,B,C\}$

$\delta((A,B), b) = \delta(A,b) \cup \delta(B,b) = \{A\} \cup \{C\}$
$= \{A,C\}$.

$\delta(\{ABC\}, a) = \delta(A,a) \cup \delta(B,a) \cup \delta(C,a)$
$= \{A,B\} \cup \{C\} \cup \{D\}$
$= \{A,B,C,D\}$.

⟹ $\delta(\{A,B,C\}, b) = \delta(A,b) \cup \delta(B,b) \cup \delta(C,b)$
$= \delta(A,B) \cup \{C\} \cup \{D\}$
$= \{A,B,C,D\}$

$\delta(\{A,B,C,D\}, a) = \delta(A,a) \cup \delta(B,a) \cup \delta(C,a) \cup \delta(D,a)$
$= \{A,B\} \cup \{C\} \cup \{D\} \cup \phi$
$= \{A,B,C,D\}$

$\rightarrow \delta(\{A,B,C,D\},b) = \delta(A,b) \cup \delta(B,b) \cup \delta(C,b) \cup \delta(D,b)$

$= \{A\} \cup \{C\} \cup \{D\} \cup \emptyset$

$= \{A,C,D\}$

$\rightarrow \delta(\{A,C,D\},a) = \delta(A,a) \cup \delta(C,a) \cup \delta(D,a)$

$= \{A,B\} \cup \{D\} \cup \{\emptyset\}$

$= \{A,B,D\}.$

$\rightarrow \delta(\{A,C,D\},b) = \delta(A,b) \cup \delta(C,b) \cup \delta(D,b)$

$= \{A\} \cup \{D\} \cup \delta\{\emptyset\}$

$= \{A,D\}.$

$\rightarrow \delta(\{A,D\},a) = \delta(A,a) \cup \delta(D,a) = \cancel{\delta}\ \{A,B\} \cup \{\emptyset\} = \{A,B\}$

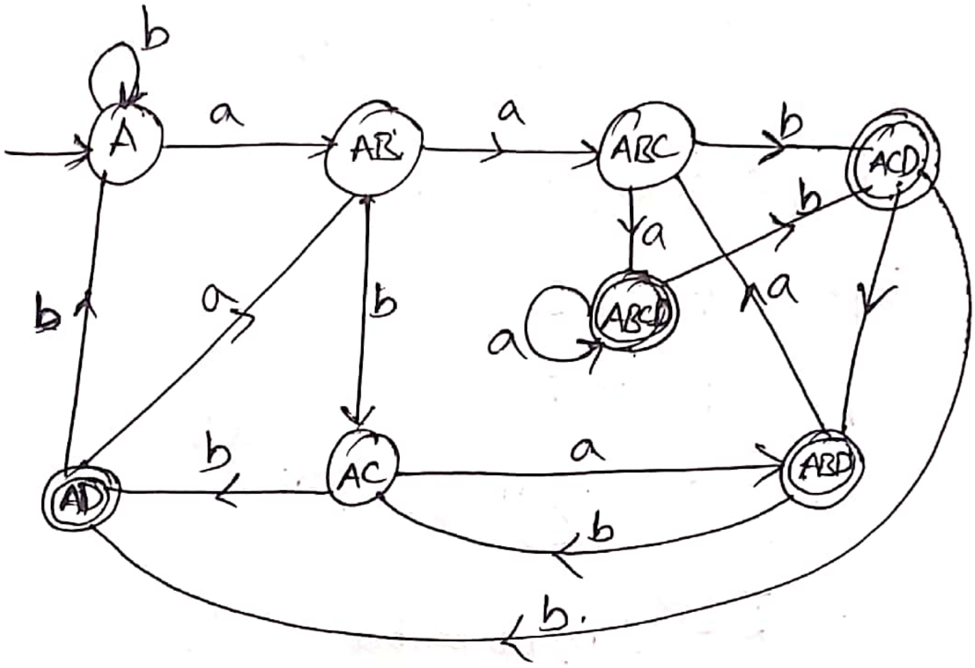
$\rightarrow \delta(\{A,D\},b) = \delta(A,b) \cup \delta(D,b) = \cancel{\delta}\ \{A\} \cup \{\emptyset\} = \{A\}.$

DFA ~~Diagram~~ Table:

	a	b.
→{A}	{AB}	{A}
{AB}	{ABC}	{AC}
{AC}	{ABD}	{AD}
{ABC}	{ABCD}	{ACD}
* {A,D}	{AB}	{A}
*{A,B,D}	{ABC}	{AC}
* {ABCD}	{ABCD}	{ACD}
*{A CD}	{ABD}	{AD}

* – Final States

DFA Diagram:

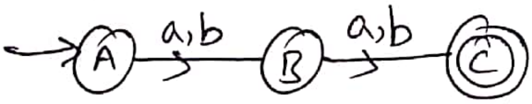


Required DFA //.

(5) NFA for strings of length a) exactly 2 (b) atmost 2
c) atleast 2

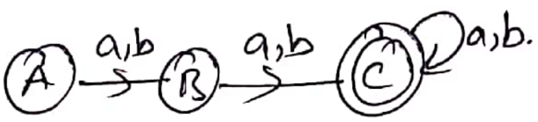
Soln $L = \{ab, ba, aa, bb\}$.

a) Exactly (2)



(b) atmost 2

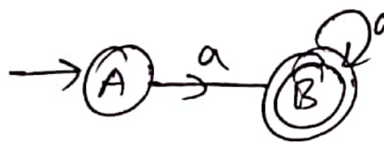
c)



(6) Complementation of NFA :-

If 'L' is a Language of NFA then the '$\bar{L}$' is called as Complementation of NFA.

Ex: $L = \{$Start with 'a'$\}$.

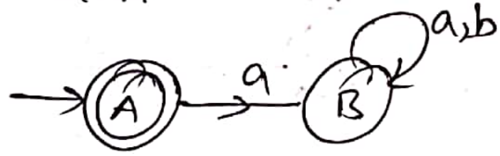
$M =$ 

$\Rightarrow$ NFA $L = \{$Start with 'a'$\}$

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

Complement of $L = \bar{L}$

Complement of $M = \bar{M}$

$\bar{M} =$ 

$\bar{L} = \{\varepsilon, b, bb, ba, bbb, \ldots\}$.

$L = \{\varepsilon\}$.

Minimization of FSM's

* "Minimization of FSM means "Reducing the no. of States" from given Finite Automata.

* While minimizing FSM, we first find out which two States are equivalent, we can represent those two states by one representative state.

* The two states q_1 & q_2 are equivalent if both $\delta(q_1, x)$ & $\delta(q_2, x)$ are final states (or) both of them are non-final states, for all $x \in \Sigma^*$.

Procedure: ① Divide the set of States into '2' groups

a) Final States group ⓑ Non-Final States group.

② Divide the groups based on i/p's, up to forming of Individual States (or) there is no chance to divide the group.

③ The given machine & reduced machine should Consists of same no. of States, then two machines are called as "equivalence" otherwise "Not equivalence"

If (P, q) equivalent then $\delta(P, \omega) \in F \Rightarrow \delta(q, \omega) \in F$

$\delta(P, \omega) \notin F \Rightarrow \delta(q, \omega) \notin F$

$|\omega| = 0 \Rightarrow$ 0-equivalent

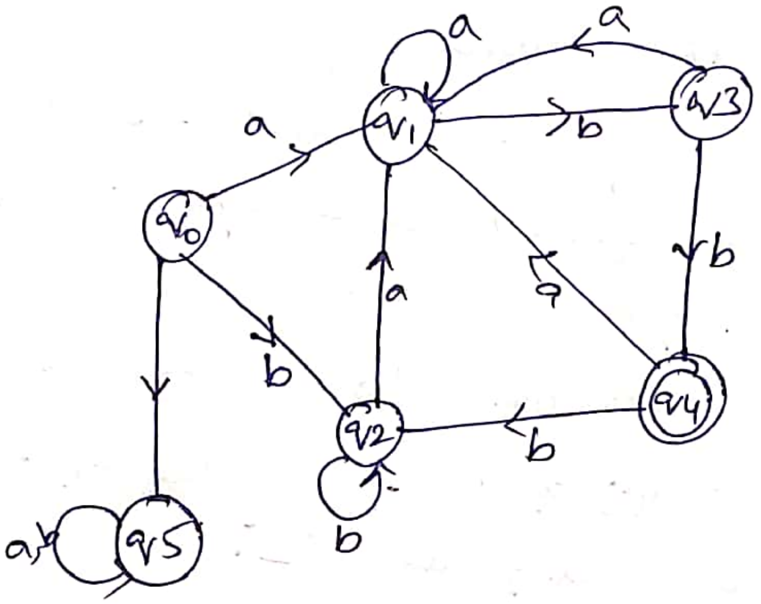
$|\omega| = 1 \Rightarrow$ 1-equivalent

$\vdots$

$|\omega| = n \Rightarrow$ n-equivalent

Problem

1) Construct the minimum State Automaton for the following transition Diagram?



* Delete the States which is not reachable to Final State.

q5 is not reaching So, remove it.

Step①:

Transition Table:

	a	b
→q_0	q1	q2
q_1	q1	q3
q_2	q1	q2
q_3	q1	q4
* q_4	q1	q2

Step②: Start Constructing the equivalence classes.

Step③: divide the No. of States "Q" as Final States (Q1) & Non-Final States set (Q2).

Q — Q_1 – Final States
Q_2 – Non Final States.

Given Diagram has

Q_1 = No. of Final States $\{q_4\}$

Q_2 = No. of Non-Final States = $\{q_0, q_1, q_2, q_3, q_5\}$

q_5 → dead state

→ Divide the terms as 0 - equivalence
1 - equivalence
⋮
n - equivalence..

	a	b
→ q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
* q_4	q_1	q_2

Step (4) :-

0 - equivalence :-

$[q_0\ q_1\ q_2\ q_3]$ $[q_4]$

* Compare q_0, q_1
q_0, q_2
q_0, q_3
} If these are not in the same set divide the set.

	a	b
q_0	q_1	q_2
q_1	q_1	q_3

Same Set

	a	b
q_0	q_1	q_2
q_2	q_1	q_2

Same Set

	a	b
q_0	q_1	q_2
q_3	q_1	q_4

different in set. So divide it.

1 - equivalence :-

$[q_0\ q_1\ q_2]$ $[q_3]$ $[q_4]$

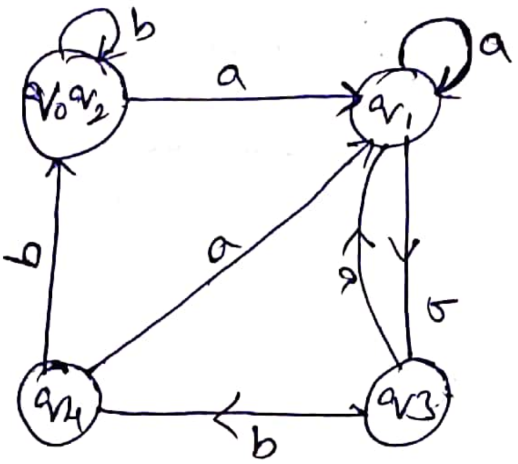
2 - Equivalence :

$[q_0\ q_2]$ $[q_1]$ $[q_3]$ $[q_4]$

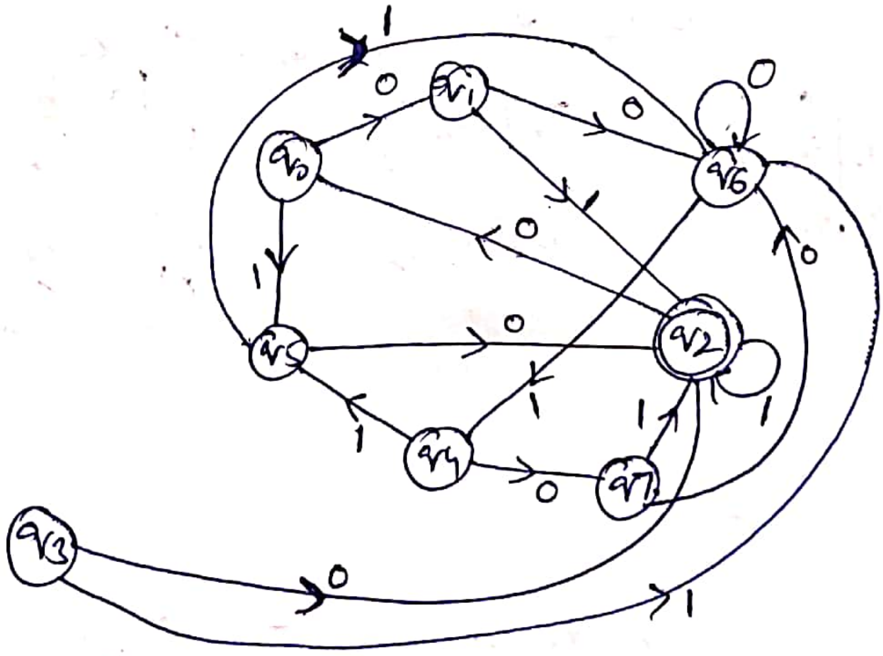
3- equivalence :-

$[q_0\ q_2]\ [q_1]\ [q_3]\ [q_4]$.

Step⑤ : Final Diagram



Problem:② Construct the minimum State Automaton for the following Transition Diagram?



Sol: Final State $= \{q_2\}$

Non Final State $= \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$ → Don't have any transition. It will become dead state.

$= \{q_0, q_1, q_4, q_5, q_6, q_7\}$.

* Transition Table :

I/P / States	0	1
→ q_0	q_1	q_5
q_1	q_6	$*q_2$
* q_2	q_0	$*q_2$
q_3	$*q_2$	q_6
q_4	q_7	q_5
q_5	$*q_2$	q_6
q_6	q_6	q_4
q_7	q_6	$*q_2$

→ Dead State, So remove it from set.

→ **0-Equivalence**

$\{q_0, q_1, q_4, q_5, q_6, q_7\}$ $\{q_2\}$

→ ~~1-Equivalence~~

→ **1-Equivalence :-**

$\{q_0, q_4, q_6\}$ $\{q_1, q_7\}$ $\{q_5\}$ $\{q_2\}$

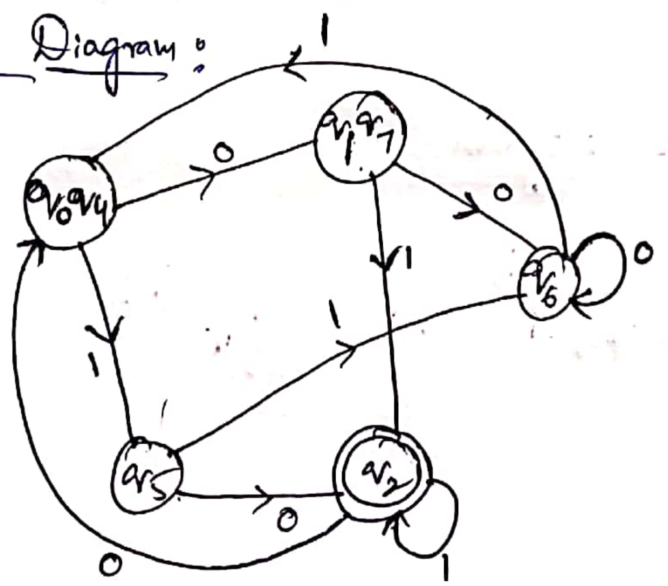
→ **2-Equivalence :**

$\{q_0, q_4\}$ $\{q_6\}$ $\{q_1, q_7\}$ $\{q_5\}$ $\{q_2\}$

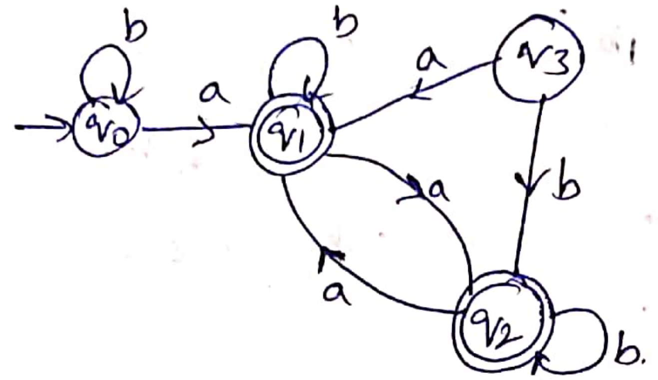
→ **3-Equivalence :**

$\{q_0, q_4\}$ $\{q_6\}$ $\{q_1, q_7\}$ $\{q_5\}$ $\{q_2\}$

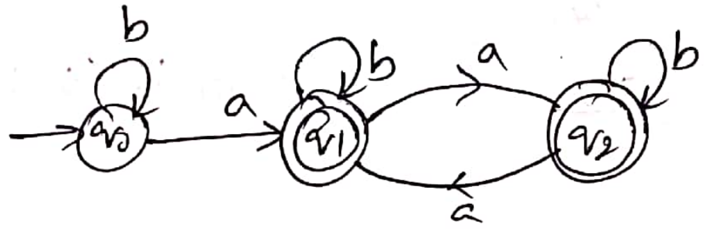
Final Diagram :



3 Minimize the FA.



Note: By using 'q_3', we can't reach to Final State (q_2), So ~~skip~~ remove 'q_3' State.



Transition Table:-

	a	b
$\rightarrow q_0$	$*q_1$	q_0
$*q_1$	$*q_2$	$*q_1$
$*q_2$	$*q_1$	$*q_2$

Equivalence:

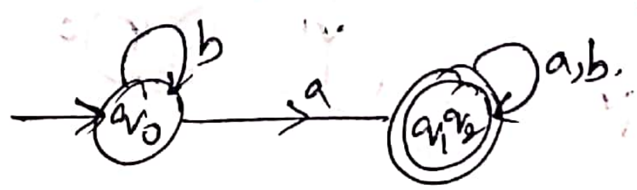
a) 0 - Equivalence:-

Final State , $\{q1, q2\}$

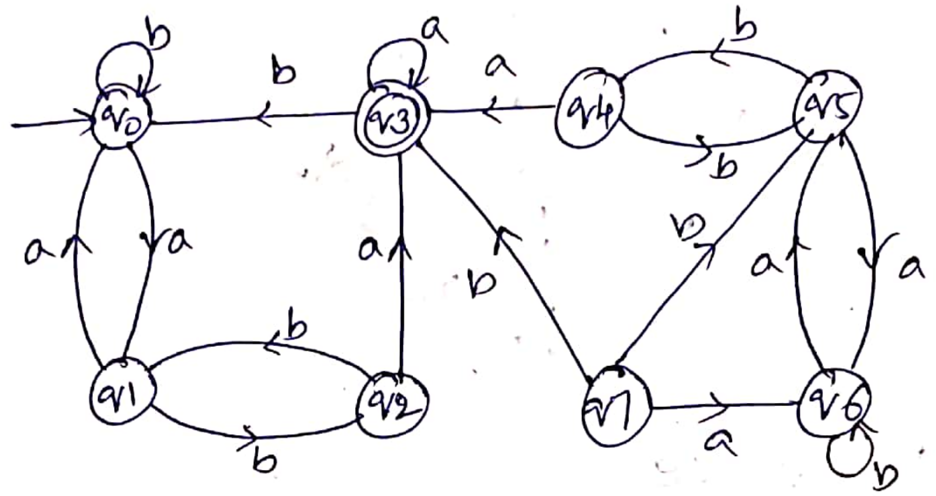
Non Final State , $\{q0\}$.

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

Final Diagram:

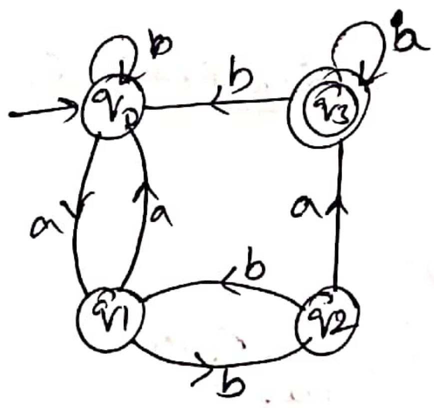


4. Minimize the Finite Automaton.



Sol: In the given Diagram $\{q_4, q_5, q_6, q_7\}$ are not reaching to Initial State and those are differed from Diagram not moving through $\{q_0, q_1, q_2, q_3\}$ and ViceVersa. remove q_4, q_5, q_6, q_7

After removing $\{q_4, q_5, q_6, q_7\}$ States, the Diagram is



Transition Table:-

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	$*q_3$	q_1
$*q_3$	$*q_3$	q_0

Equivalence List:

Final State = $\{q_3\}$

→ 0-Equivalence:-

$\{q_0, q_1, q_2\} \cdot \{q_3\}$

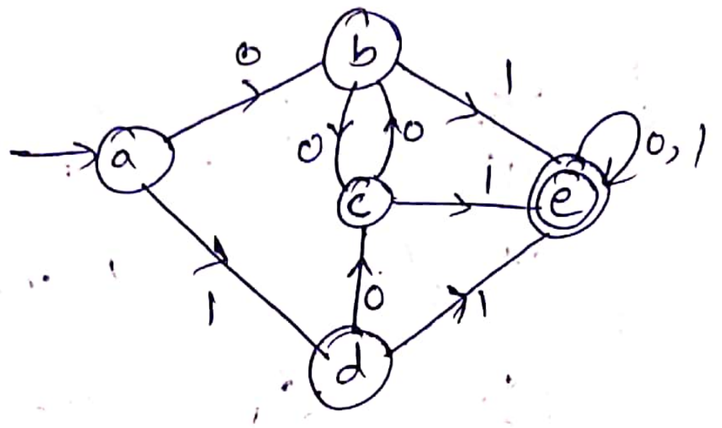
Non-Final States $\{q_0, q_1, q_2\}$

→ 1-Equivalence; $\{q_0, q_1\}\{q_2\}\{q_3\}$.

→ 2-Equivalence: $\{q_0\}\{q_1\}\{q_2\}\{q_3\}$.

Note: At last 4-States divided to the Diagram is equivalence to the above given Diagram.

⑤ Minimize the Finite Automaton given Diagram. (66)



Soln Final State = {e}

Non Final States = {a, b, c, d}.

Transition Table:

	0	1
→a	b	d
b	c	*e
c	b	*e
d	c	*e
*e	*e	*e

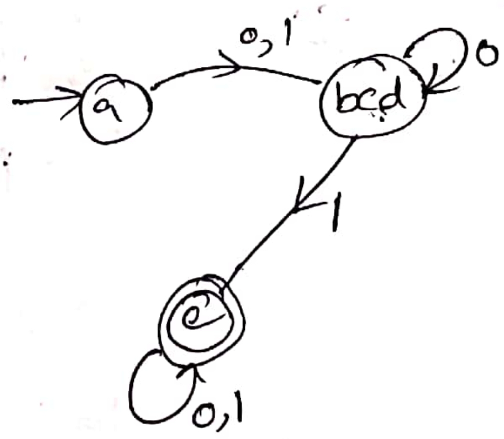
Equivalence List:

0-Equivalence: {a, b, c, d} {e}

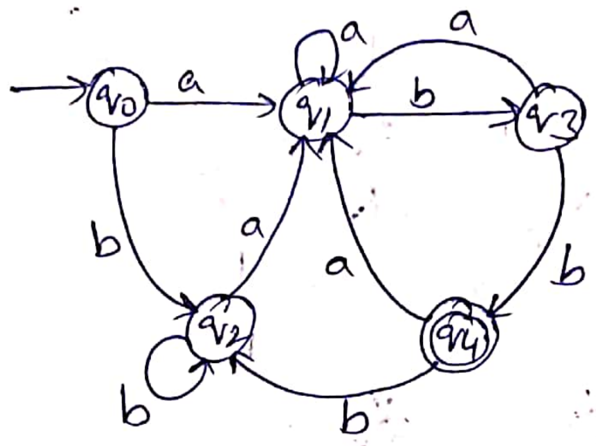
1-Equivalence: {a} {b c d} {e}

2-Equivalence: {a} {b c d} {e}.

Final Diagram:



6 Minimize the Finite Automata for the following DFA



Solution: Final State = $\{q_4\}$

Non Final-State = $\{q_0, q_1, q_2, q_3\}$

Transition Table:-

	a	b
→ q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	$*q_4$
$*q_4$	q_1	q_2

Equivalence List:-

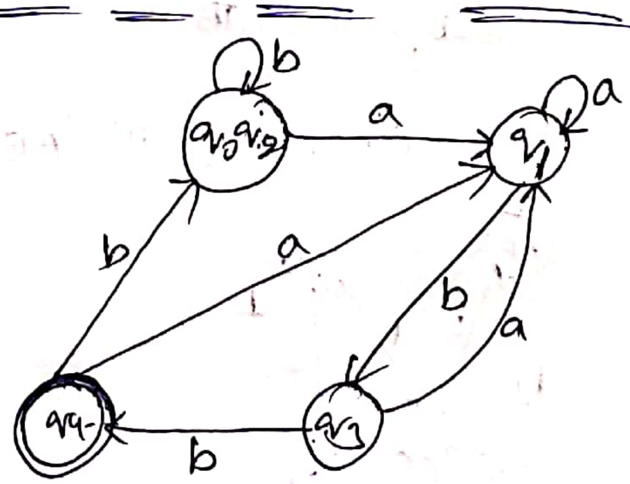
→ 0-Equivalence:- $\{q_0, q_1, q_2, q_3\}$ $\{q_4\}$

→ 1-Equivalence:- $\{q_0, q_1, q_2\}$ $\{q_3\}$ $\{q_4\}$

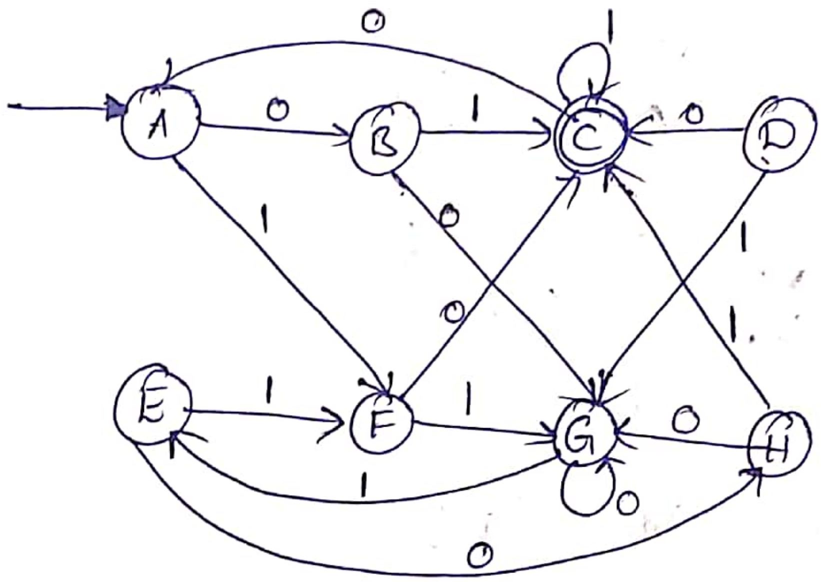
→ 2-Equivalence:- $\{q_0, q_2\}$ $\{q_1\}$ $\{q_3\}$ $\{q_4\}$

→ 3-Equivalence:- $\{q_0, q_2\}$ $\{q_1\}$ $\{q_3\}$ $\{q_4\}$

Final State Minimum Automata:



7. Minimize the DFA as given below:



Soln

Final State = {C}

Non-Final State = {A, B, D, E, F, G, H}

Transition Table:

	0	1
→ A	B	F
B	G	*C
*C	A	*C
D	*C	G
E	H	F
F	*C	G
G	G	E
H	G	*C

Equivalence List:

* 0-Equivalence (Π_0):

{C} {A, B, D, E, F, G, H}

→ 1-Equivalence (Π_1):

{C} {D, F} {B, H} {A, E, G}.

→ 2-Equivalence (Π_2):

{C} {D, F} {B, H} {A E} {G}.

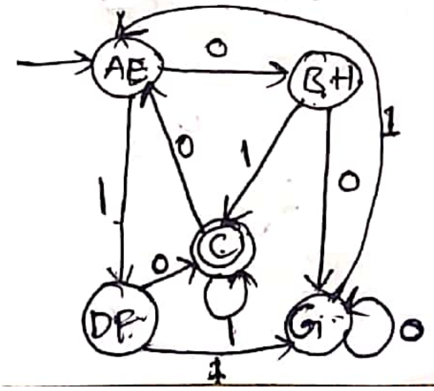
	a	b
D	B	
F	H	

	a	b
A	B	F
G	G	E

	a	b
A	B	F
E	H	F

AE

Final Diagram:



(8) Minimize the DFA for a given Transition Table. (69)

Sol:

	0	1
A	B	E
B	*C	*F
*C	D	H
D	E	H
E	*F	*I
*F	G	B
G	H	B
H	*I	*C
*I	A	E

Final state

{C,F,I}

Non Final

{A,B,D,E,G,H}

Sol: Equivalence

(a) 0-Equivalence:-

{C,F,I} {A,B, D, E,G,H}

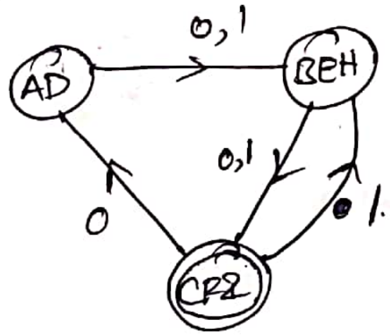
1-Equivalence:

[A D] [B E H] [C F I]

2-Equivalence:

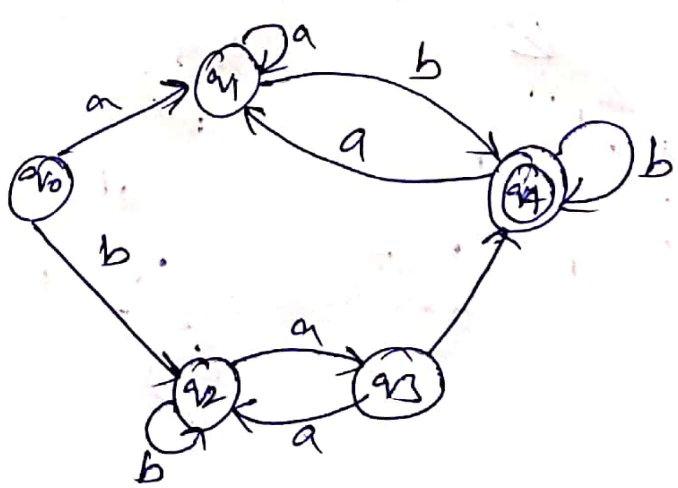
[A D][B E H] [C F I]

Final DFA:

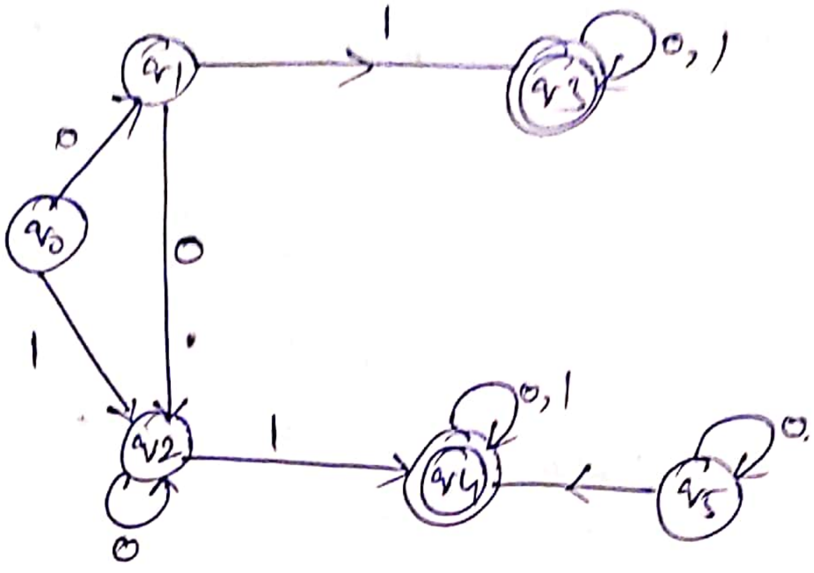


Assessment: on Minimization of FSM's

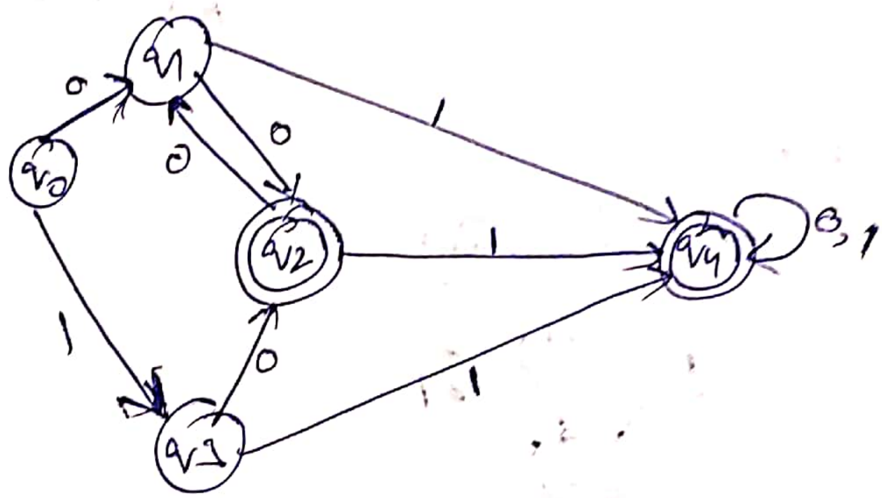
(1) DFA



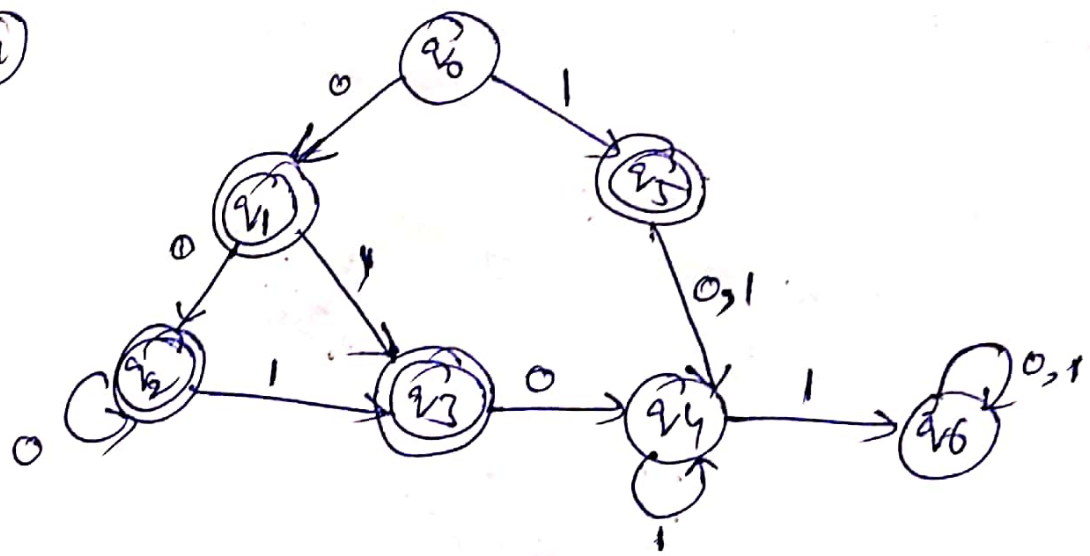
2


q1
q3
q0
q2
q4
q5
0
1
0, 1
0, 1
0.

3


q1
q0
q2
q3
q4
0
1
0, 1

4


q0
q1
q5
q2
q3
q4
q6
0
1
0, 1

Equivalence between Two FSM's :-

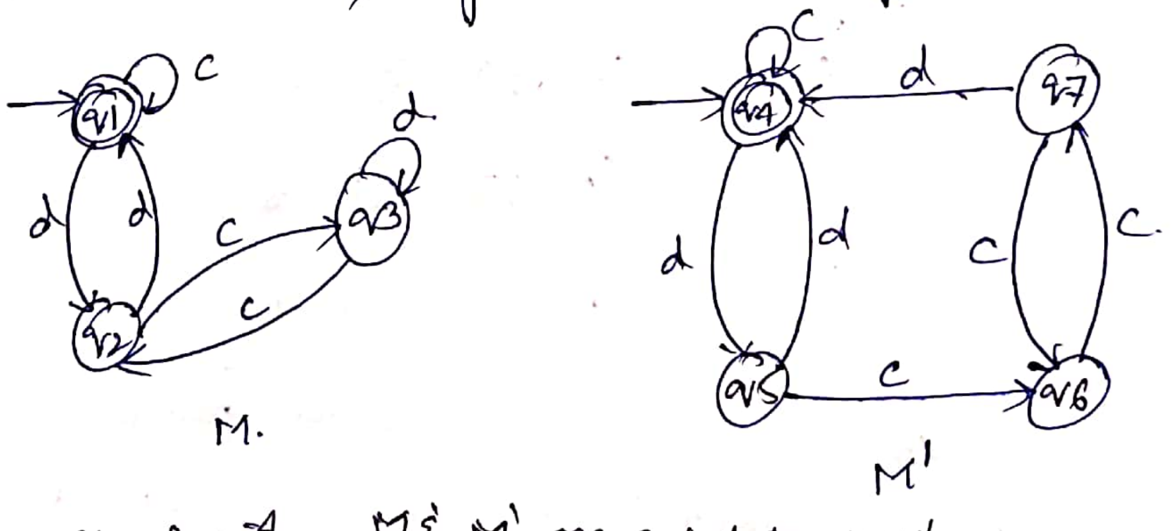
The two FA's are said to be equivalent, If both the automata accept the same set of Strings over an Input set "Σ"

Method for Comparing two FA's :-

Let M, M' be two FA's, and Σ is a set of Input Strings.

* we will Construct a transition table have pair wise entries (q, q') where $q \in M$ and $q' \in M'$ for each Input Symbol.
* If we get in a pair as one final State & other non-final State then we terminate Construction of transition table and declare that two FSM's are not equivalent.
* The Construction of transition Table gets terminated, when there is no new pair appearing in the transition Table.

① Consider the DFA's given below are they equivalent.



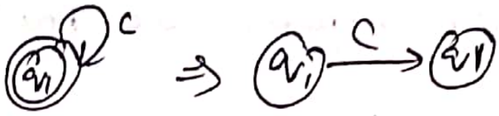
Soln Checking of M & M' are equivalent or not.

* Construct the Transition Table for each i/p c & d.

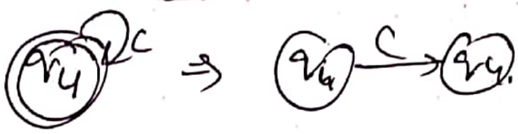
Transition Table:

Procedure:

a) First Machine M on receiving i/p = c in state q_1 we reach to state q_1 only.



b) From Second Machine M_1', for 'State q_4 on receiving 'c' we reach to State 'q_4'



$$\therefore (q_1, q_4) \xrightarrow{c} (q_1, q_4).$$

Similarly $(q_1, q_4) \xrightarrow{d} (q_2, q_5)$.

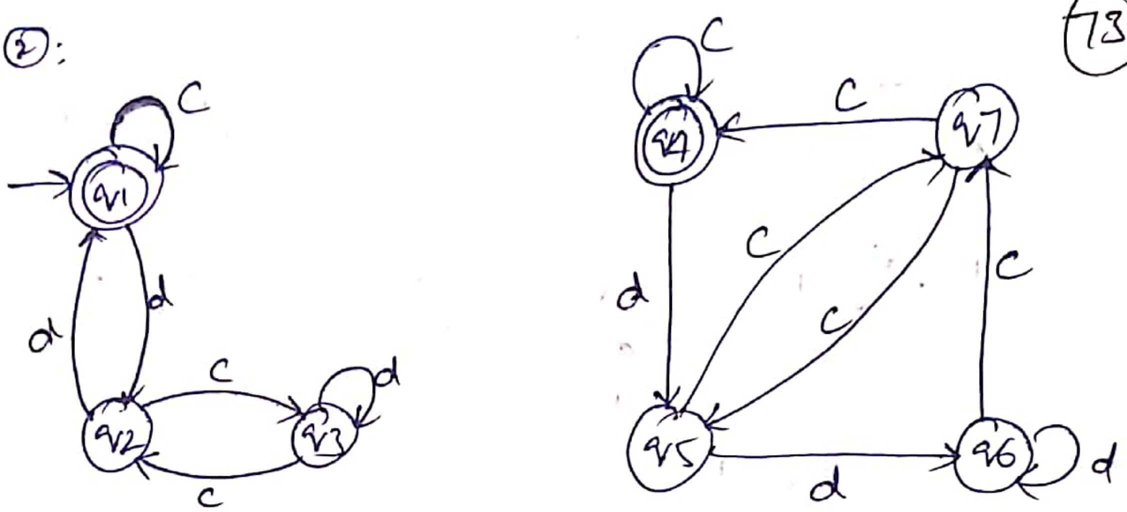
Transistion Table:

	c	d
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4).

→ If we observe, there is no new pair now. When new pair not forming we can stop the construction of Table.

→ From the above table note that we don't get Final State & other Non-Final State in pair. So, we can say Two DFA's are "Equivalent"

Problem (2):



Check the two FA's equivalent or not?

Solution: Let M & M'

Transition Table:-

	c	d
$q_1\, q_4$	(q_1, q_4)	(q_2, q_5)
$q_2\, q_5$	(q_3, q_7)	(q_1, q_6)

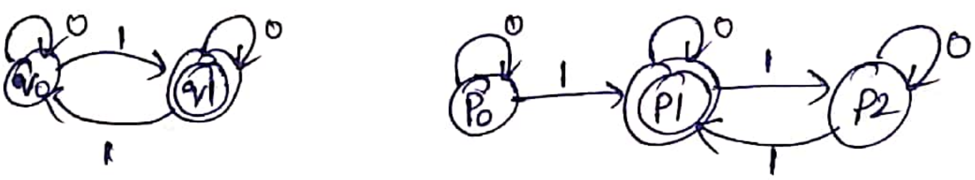
→ Stop the Construction of Transition Table. Because we get a pair (q_1, q_6) in which q_1 is Final, but q_6 is Non-Final State.

→ As per Equivalence Rule one Final & Non-Final States Can't form pairs.

→ Hence the Given FA's are not equivalent.

Assessment Questions on Two FA's are equivalent or Not?

(1)



(2)

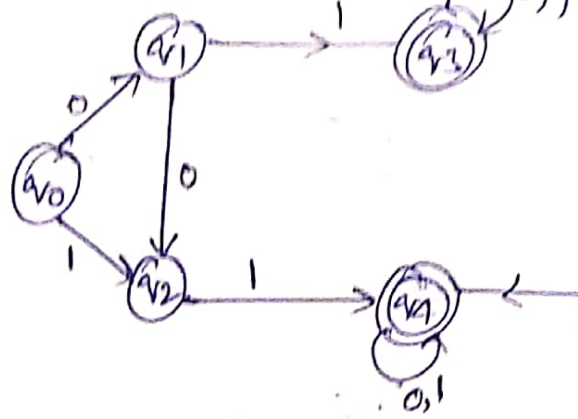


fig: M

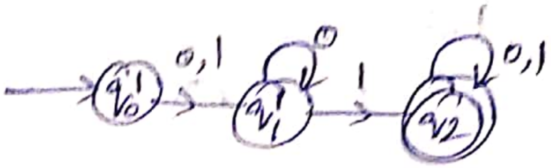
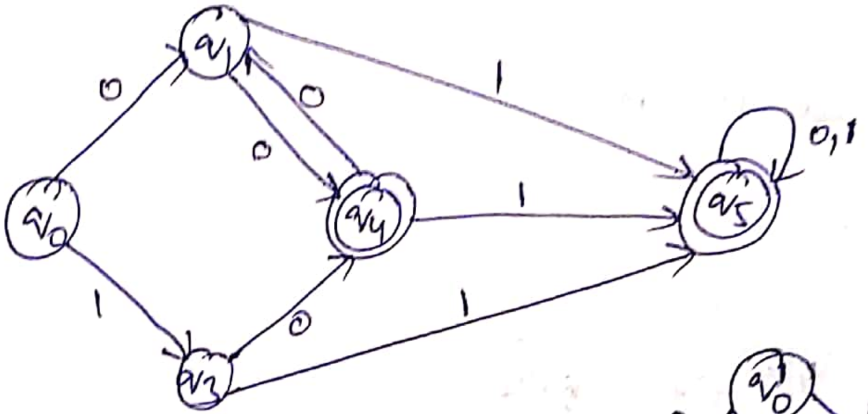
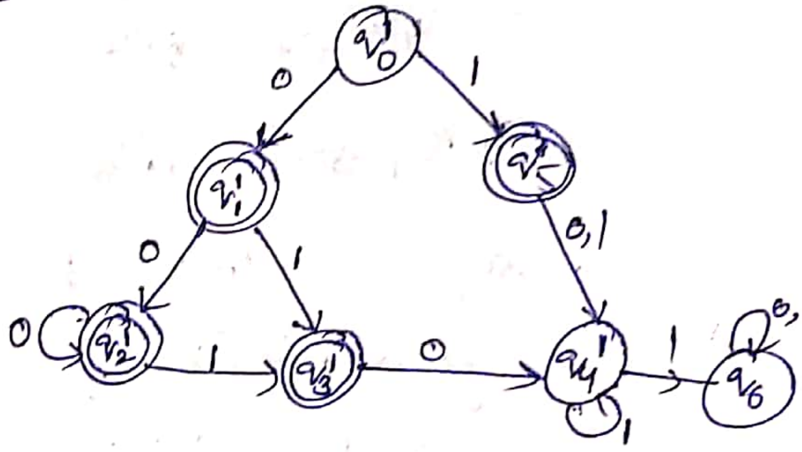


fig: M'

(3)





Finite Automata withoutput – Moore & Melay Machines (75)

Finite-AutoMata withoutput is of 2 types.

① Moore Machine ② Melay Machine

① Moore Machine

→ "Moore Machine is a FSM in which the next State is decided by Current State & Current Input Symbol.

→ The output Symbol at a given time depends only on present State of the Machine.

→ Length of Input String = n, then length of o/p String = n+1

→ Formal Definition

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

Q = Finite set of States

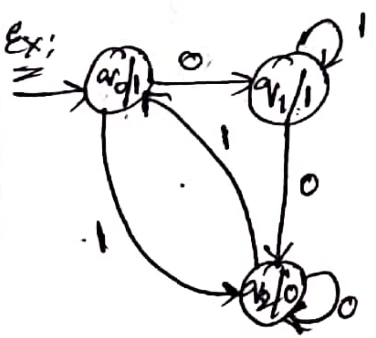
Σ = Finite set of i/p Symbols

Δ = Output Alphabet.

δ = Transistion function $Q \times \Sigma \to Q$

λ = o/p Function $Q \to \Delta$.

Ex:



Transistion Table

Current State	Next State (δ) 0	Next State (δ) 1	output (λ)
q_0	q_1	q_2	1
q_1	q_2	q_1	1
q_2	q_2	q_0	0

2) Melay Machine

→ Melay Machine is a Machine in which Output Symbol depends upon the present Input Symbol & present State of Machine.

→ Formal Definition

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

Q = Finite set of States

Σ = Finite set of i/p Symbols

Δ = is an o/p Alphabet

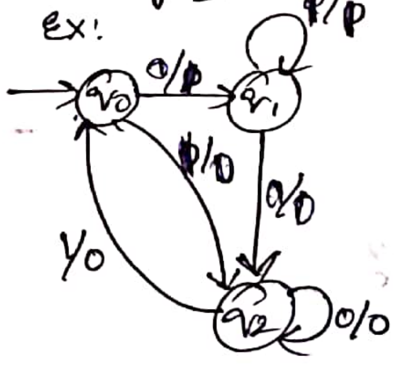
δ = State transition Function $Q \times \Sigma \to Q$

λ = Machine Function $Q \times \Sigma \to \Delta$.

q_0 = Initial State.

→ Length of Input String = n, then length of o/p String = n

Ex:



Problems;

① Construct a Moore Machine that takes set of all strings over {a,b} as input and print '1' as o/p for every occurrence of 'ab' as a substring.

Sol:- Given $\Sigma = \{a,b\}$
$\Delta = \{0,1\}$

For every occurrence of ab

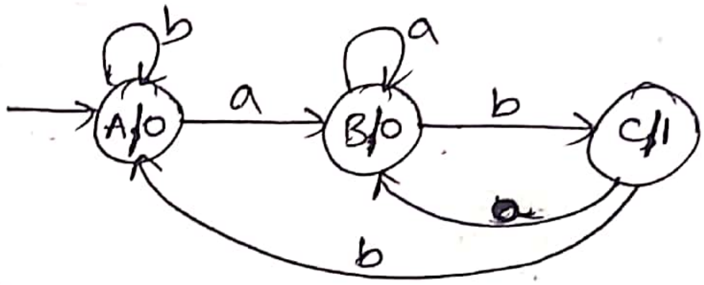
→ ab
→ ab, ab
→ ab bb ab

Steps;

① Start with ab. FA not count remaining ab's.

② Containing ab's not count first & last ab's.

③ That's the reason, FA ends with ab. That FA count no. of ab's in a string.



④ When ever you reach 'C' then print '1', the remaining state print '0'

① ab :

$$\begin{array}{ccccc} A & \xrightarrow{a} & B & \xrightarrow{b} & C \\ \downarrow & & \downarrow & & \downarrow \\ 0 & & 0 & & 1 \end{array}$$

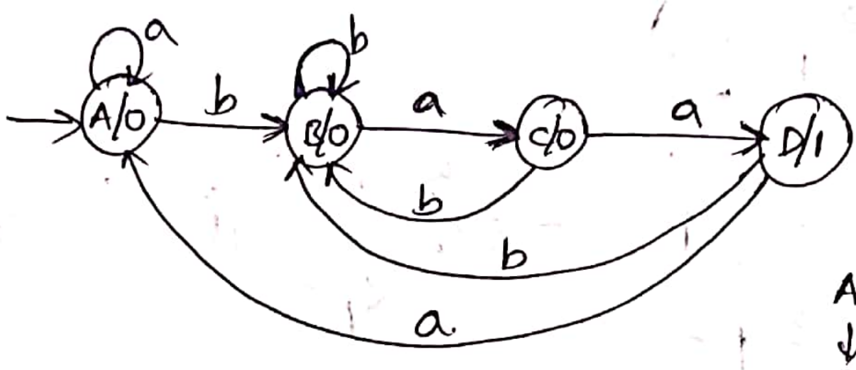
② abab :

$$\begin{array}{ccccccccc} A & \xrightarrow{a} & B & \xrightarrow{b} & C & \xrightarrow{a} & B & \xrightarrow{b} & C \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 0 & & 0 & & 1 & & 0 & & 1 \end{array}$$

(17)

② Construct Moore Machine Counting the occurence of substring 'baa' over $\Sigma = \{a, b\}$; $\Delta = \{0, 1\}$.

Soln



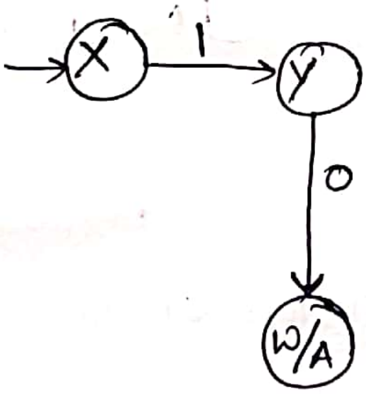
String
baa

A $\xrightarrow{b}$ B $\xrightarrow{a}$ C $\xrightarrow{a}$ D
↓ 0 ↓ 0 ↓ 0 ↓ 1

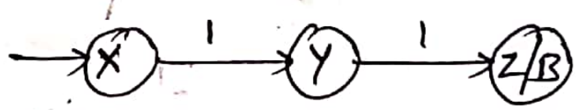
③ Construct a Moore Machine that takes set of all Strings over $\{0, 1\}$ and produces 'A' as o/p if i/p ends with '10' (or) Produces 'B' as if i/p ends with '11' otherwise 'C'.

Soln Given $\Sigma = \{0, 1\}$; $\Delta = \{A, B, C\}$

1) If i/p ends with '10' ⇒ o/p = 'A'

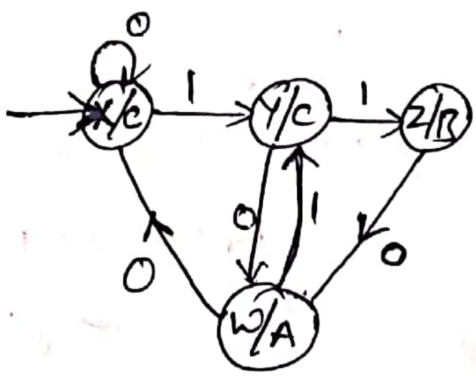


② If i/p ends with '11' ⇒ o/p = 'B'



③ Remaining all 'C'

⇒ Total Diagram:



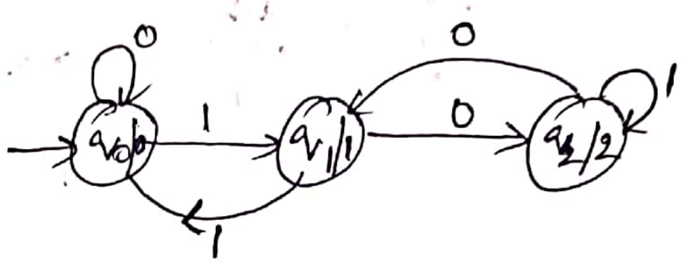
78

0
3 - 011
6 - 110
9 - 1001

(4) Construct a Moore Machine that takes binary no's as i/p and produces residue modulo '3' as o/p.

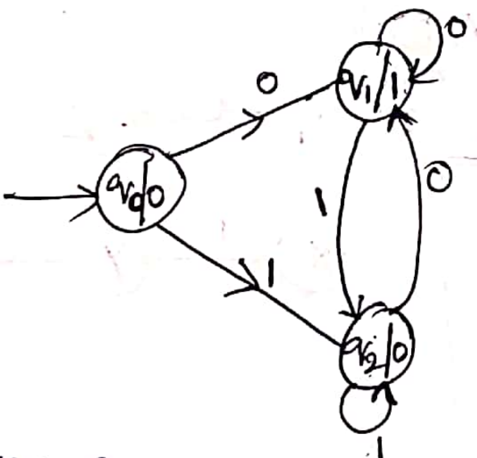
Sol: Given $\Sigma = \{0,1\}$; $\Delta = \{0,1,2\}$

	δ: 0	δ: 1	λ: Δ
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

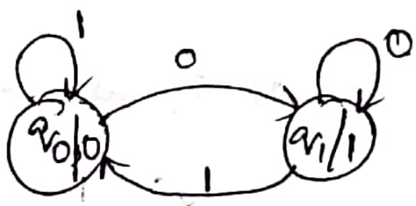


(5) Design a Moore Machine to generate 1's Complement of given binary Number?

Sol: I/p's $\Sigma = \{0, 1\}$; O/p's $\Delta = \{0,1\}$.



(or)



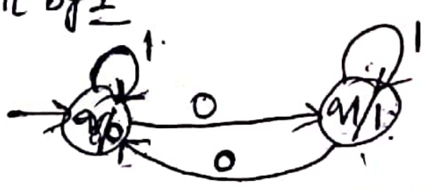
(6) Design a Moore Machine which will increment the given binary number by '1'.

Sol. → Read the binary Number from LSB one bit at a time

→ Replace each 1 by 0, until we get First '0'

→ Once we get First '0', Replace it by '1'

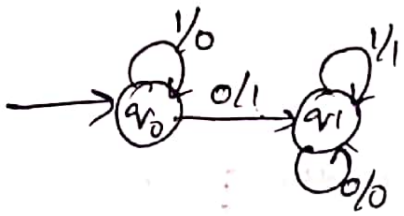
→ Keep remaining bits as it is



Scanned by CamScanner

Melay Machine

1001
1001



(7) Moore & Melay Machine For to find 2's Complement of a given Binary Number

Sol: * Read Input from LSB → MSB

* Will keep that binary string upto, we hit the First '1'.

* Keep that 1st '1' as it is, then change the Remaining 1's as → 0's & 0's as → 1's

Ex: 1011

MSB ← LSB

1011

→ Keep First '1' as it is, Remaining change the bits

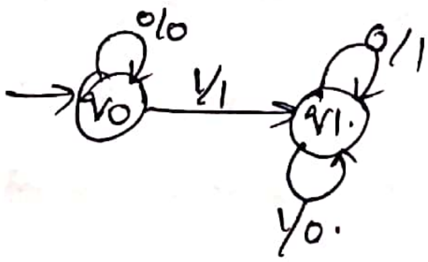
1011
↓
0101

∴ 2's Complement of 1011 is 0101.

Examply

1100 → 0100
11101100 → 00010100
1011 → 0101

Melay Machine:



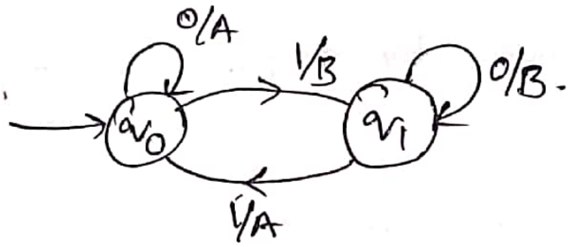
Moore Machine

(8) Construct Melay Machine which can output even/odd according as the total No. of 1's encountered is even (or) odd. The Input Symbols are 0 (or) 1.

(80)

Sol:

There is no restriction on number of 0's



Assessment:

(9) Design Moore Machine which Converts a ternary number into decimal form by using Modulo(5).

Hint: Ternary: i/p's = 0, 1, 2.

(10) Moore Machine & Melay Machine for i/p Integer Strings represented as binary and give the o/p '0' for Multiples of 2, and 1 - for others.

(11) Moore Machine i/p from $(0+1)^*$, if the i/p ends with 101 → o/p = A; If the i/p ends with 110 → o/p = B; For others it is C.

(12) Melay-Machine for Modulo(3) for binary language.

(13) Melay-Machine for Modulo(5). for ternary language

(14) Melay-Machine for which have the o/p symbol 'E' (or) 'O' - odd According to as total no. of one's encountered is even (or) odd. $\Sigma = \{0,1\}$

(15) Melay Machine for accepting String from a i/p $\Sigma = \{0,1\}$ ending with 2 consecutive zero's & one's.

Conversion of Moore Machine to Mealy Machine :-

Let $M = (Q, \Sigma, \delta, \lambda, q_0)$ be a Moore Machine. The equivalent Mealy Machine can be represented by $M' = (Q, \Sigma, \delta, \lambda', q_0)$.

The o/p Function λ' can be obtained as

$$\lambda'(q, a) = \lambda(\delta(q, a))$$

Problem ① The Moore Machine to determine residue mod 3 for binary Number is given below. Convert it to Mealy equivalent Machine.

Q \ Σ	0	1	o/p = λ
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

Soln Transition Diagram for the given problem is as below.

According to Formula: $\lambda'(q, a) = \lambda(\delta(q, a))$

Output for every Transition is

$\rightarrow \lambda'(q_0, 0) = \lambda(\delta(q_0, 0)) = \lambda(q_0) \rightarrow$ o/p of $q_0 \Rightarrow \boxed{\lambda'(q_0, 0) = 0}$

$\rightarrow \lambda'(q_0, 1) = \lambda(\delta(q_0, 1)) = \lambda(q_1) \rightarrow$ o/p of $q_1 \Rightarrow \boxed{\lambda'(q_0, 1) = 1}$

$\rightarrow \lambda'(q_1, 0) = \lambda(\delta(q_1, 0)) = \lambda(q_2) \rightarrow$ o/p of $q_2 \Rightarrow \boxed{\lambda'(q_1, 0) = 2}$

$\rightarrow \lambda'(q_1, 1) = \lambda(\delta(q_1, 1)) = \lambda(q_0) \rightarrow$ o/p of $q_0 \Rightarrow \boxed{\lambda'(q_1, 1) = 0}$

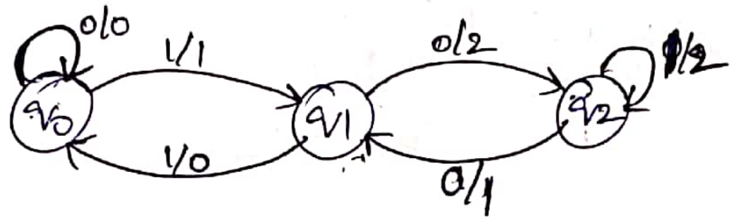
$\lambda'(q_2,0) = \lambda(\delta(q_2,0)) = \lambda(q_1) \Rightarrow \boxed{\lambda'(q_2,0)=1}$

$\lambda'(q_2,1) = \lambda(\delta(q_2,1)) = \lambda(q_2) \Rightarrow \boxed{\lambda'(q_2,1)=2}$

Hence the Transition Table for Melay Machine is as belao.

Q \ Σ	Input 0		Input 1	
	State	O/P	State	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

Transition Diagram of Melay Machine:-



String Acceptance:

String = 10011 ⇒ String length $|w|=5$

Melay Machine

Next State $q_1 \rightarrow q_2 \rightarrow q_2 \rightarrow q_1 \rightarrow q_0$

Output 1 2 2 1 0

length = 5

Consider = n

Moore Machine

Inputs: 1 0 0 1 1

Next State $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_2 \rightarrow q_1 \rightarrow q_0$

O/p 0 1 2 2 1 0

length = 6

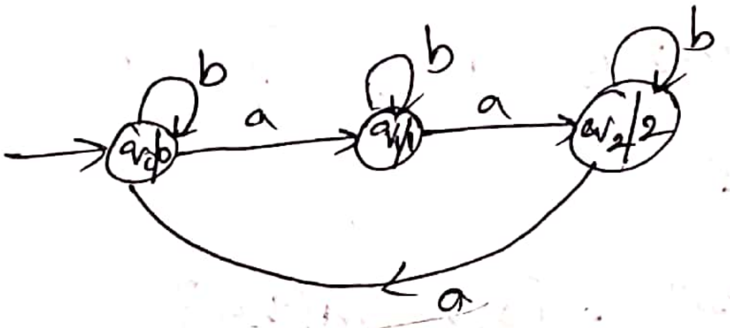
Consider "n+1"

(2) Assessment Problem: (Moore $\xrightarrow{to}$ Melay)

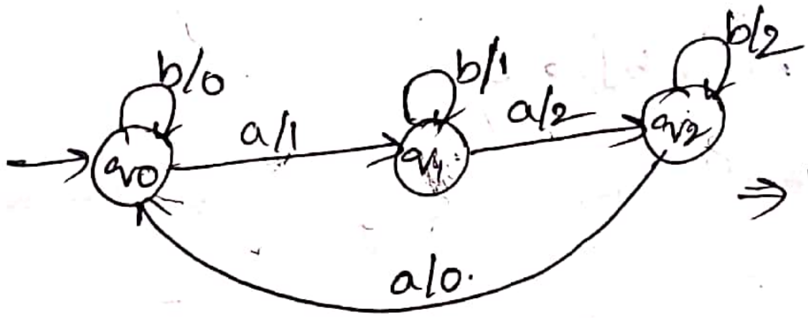
Ex $M = (\{q_0, q_1\}, \{a, b\}, \{0, 1\}, \delta, \lambda, q_0)$.

Problem :- (Moorey to Melay)

① Count No. of a's %3. (Moore to Melay)



⇒ Moore.



⇒ Melay

Transition Table:

Moore :

	a	b	Δ
→ q_0	q_1	q_0	0
q_1	q_2	q_1	1
q_2	q_0	q_2	2

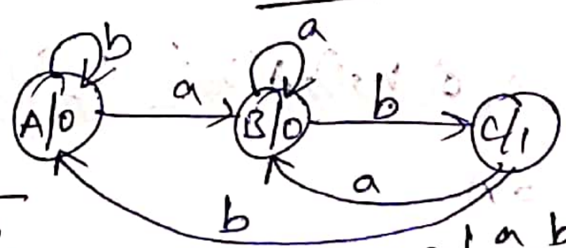
Melay :-

	a	b
q_0	$(q_1,1)$	$(q_0,0)$
q_1	$(q_2,2)$	$(q_1,1)$
q_2	$(q_0,0)$	$(q_2,2)$

② Moore – Melay :

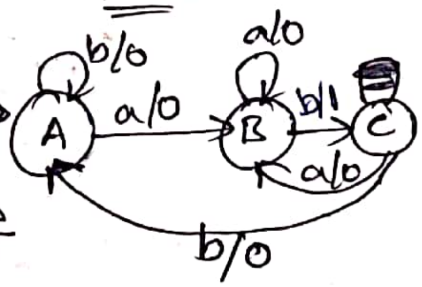
Given

Moore



⇒

Melay



	a	b
A	(B,0)	(A,0)
B	(B,0)	(C,1)
C	(B,0)	(A,0)

Melay

⟸ Moore !

	a	b	Δ
A	B	A	0
B	B	C	0
C	B	A	1

Conversion From Mealy Machine to Moore Machine

Let $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ — Mealy Machine

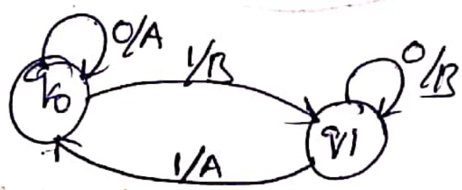
$M' = (Q \times \Delta, \Sigma, \Delta, \delta', \lambda', [q_0, b_0])$ — b_0 = O/P Symbol.

Calculate δ', λ':

Formula: $\delta'([q, b], a) = [\delta(q, a), \lambda(q, a)]$

$\lambda([q, b]) = b.$

Problem: Convert the following Mealy Machine into equivalent Moore Machine.



Soln: States for Moore Machine $[q_0, A], [q_0, B], [q_1, A]$, $[q_1, B]$

Calculation of δ', λ' :-

1) $\delta'([q_0, A], 0) = [\delta(q_0, 0), \lambda(q_0, 0)] = [q_0, A]$

$\lambda'([q_0, A]) = A.$

2) $\delta'([q_0, A], 1) = [\delta(q_0, 1), \lambda(q_0, 1)] = [q_1, B]$

$\lambda'([q_0, A]) = A$

3) $\delta'([q_1, A], 0) = [\delta(q_1, 0), \lambda(q_1, 0)] = [q_1, B] \Rightarrow \lambda'([q_1, A]) = A.$

4) $\delta'([q_1, A], 1) = [\delta(q_1, 1), \lambda(q_1, 1)] = [q_0, A] \Rightarrow \lambda'([q_1, A]) = A.$

5) $\delta'([q_1, B], 0) = [\delta(q_1, 0), \lambda(q_1, 0)] = [q_1, B]$

$\lambda'(q_1, B) = B.$

6) $\delta'([q_1, B], 1) = [\delta(q_1, 1), \lambda(q_1, 1)] = [q_0, B]$

$\lambda'[(q_1, B)] = B.$

7) $\delta'([q_0, B], 0) = [\delta(q_0, 0), \lambda(q_0, 0)] = [q_0, A]$

$\lambda'[(q_0, B)] = B.$

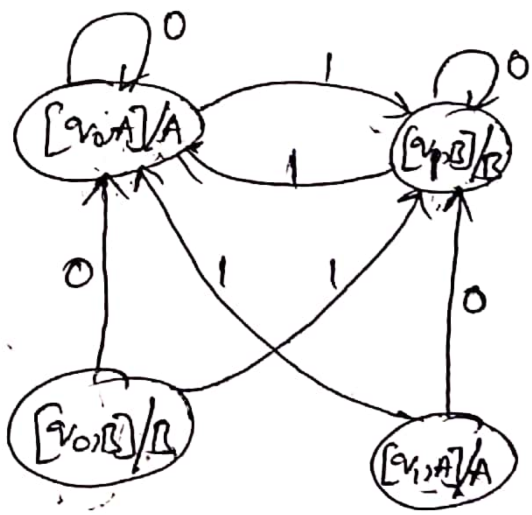
8) $\delta'([q_0, B], 1) = [\delta(q_0, 1), \lambda(q_0, 1)] = [q_1, B]$

$\lambda'(q_0, B) = B.$

Transition Table

State \ I/P	0	1	O/P
$[q_0, A]$	$[q_0, A]$	$[q_1, B]$	A
$[q_0, B]$	$[q_0, A]$	$[q_1, B]$	B
$[q_1, A]$	$[q_1, B]$	$[q_0, A]$	A
$[q_1, B]$	$[q_1, B]$	$[q_0, A]$	B

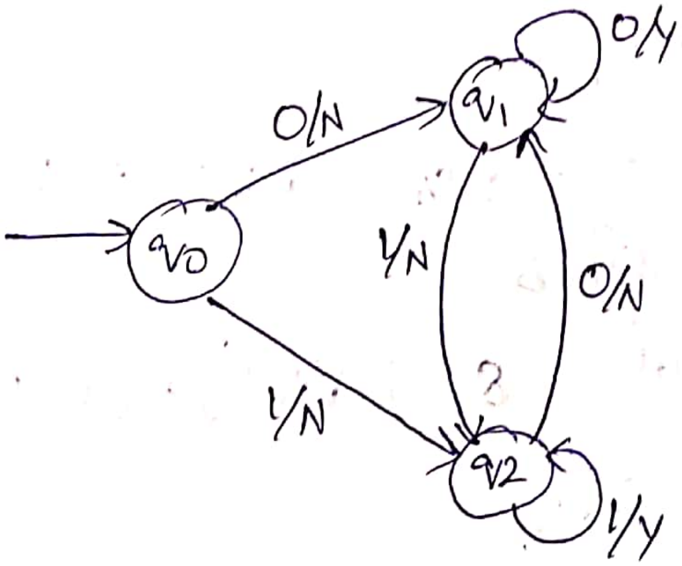
Transition Diagram



~~Assesment problem~~

Problem 2

Convert the following Mealy Machine into equivalent Moore Machine.



① Problems

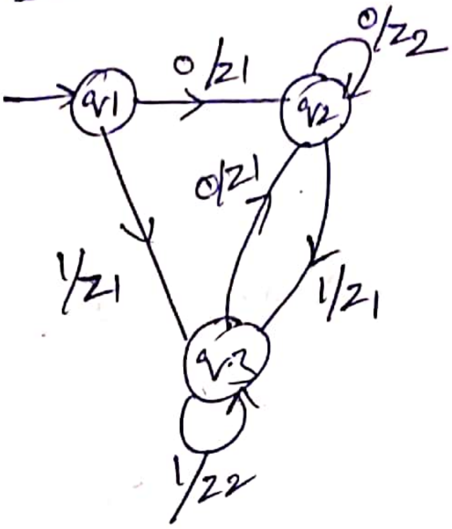


fig Mealy.

$\Rightarrow$

Moore Machine

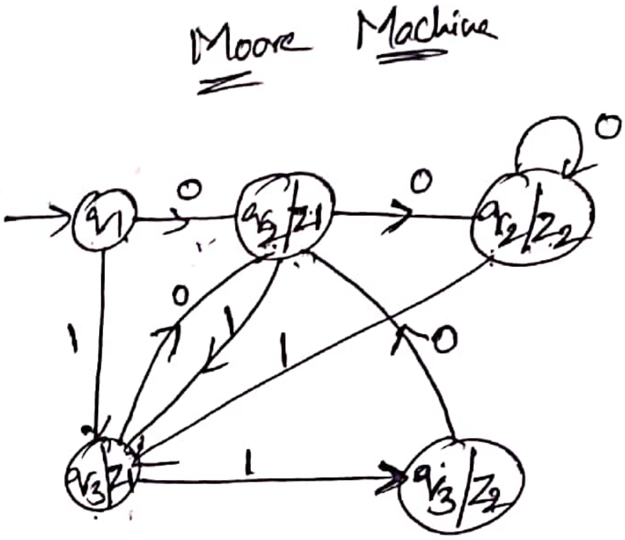
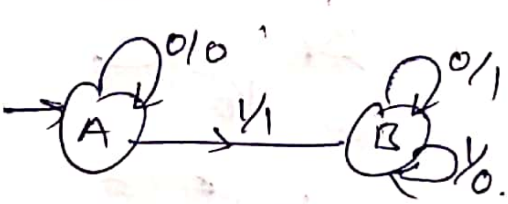


fig: Moore

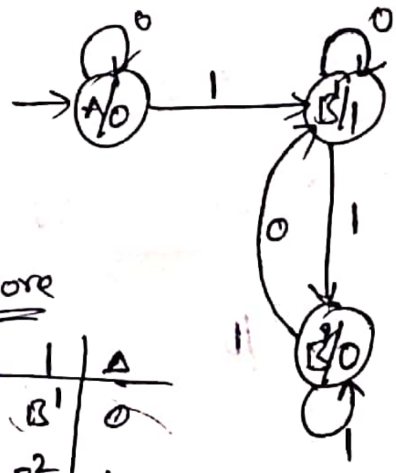
	0	1
q_1	(q_2, Z_1)	(q_3, Z_1)
q_2	(q_2, Z_2)	(q_3, Z_1)
q_3	(q_2, Z_1)	(q_3, Z_2)

	0	1	Δ
q_1	q_{21}	q_{31}	$Z_1 \rightarrow \varepsilon$
q_{21}	q_{22}	q_{31}	Z_1
q_{22}	q_{22}	q_{31}	Z_2
q_{31}	q_{21}	q_{32}	Z_1
q_{32}	q_{21}	q_{32}	Z_2

② Mealy $\rightarrow$ Moore :-



$\Rightarrow$



Mealy:

	0	1
A	(A,0)	(B,1)
B	(B,1)	(B,0)

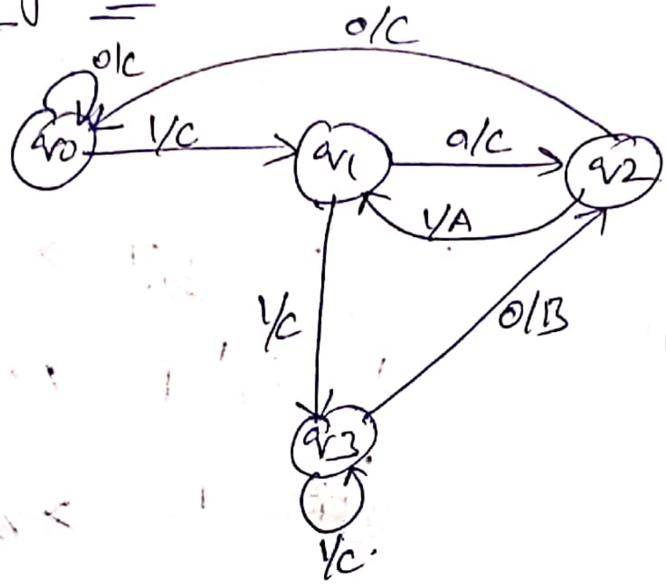
$\Rightarrow$

Moore

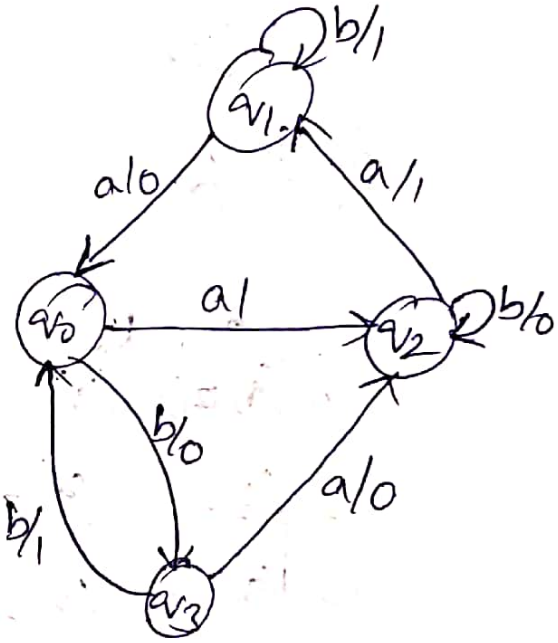
	0	1	Δ
A	A	B^1	0
B^1	B^1	B^2	1
B^2	B^1	B^2	0

Assessment Problems

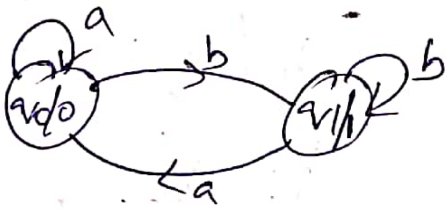
③ Melay-Moore



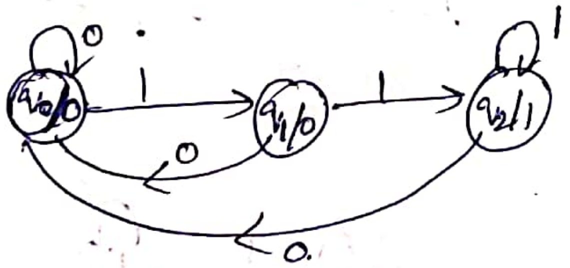
⑦ Melay - Moore



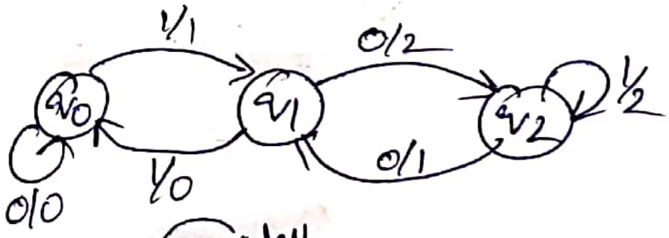
④ Moore-Melay



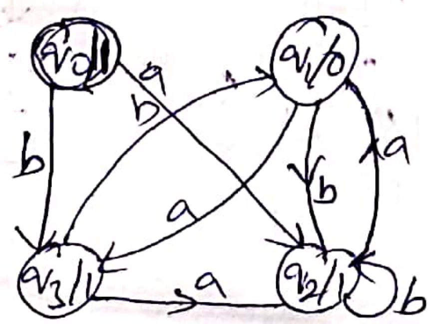
⑤ Moore → Melay:



⑧ Melay-Moore



⑥ Moore-Melay;



⑨ Moore-Melay