AUTOMATA:-

It is the machine that decides whether it should hault (or) not by itself.

→ It can also be called as automaton.

String:- A finite sequence of symbols from some alphabet is called string. Strings are denoted by 'w'.

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

$w = 0, 1, 00, 11, 01, 10, 000, 001, \ldots$

→ A string consisting of zero symbols is called empty string and is denoted by 'ϵ'.

→ The no. of symbols in the string is the length of the string.

Ex:- $w = 0011$

$|w| = 4$.

Alphabet:- A finite non-empty set of symbols is called as alphabet. It is denoted by Σ.

→ $\Sigma^* \Rightarrow$ the set of all strings over an alphabet Σ. Called as Kleene Closure.

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

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$.

Language:- A language is a set of strings all of which all choosen from Σ^* where Σ is particular alphabet. It is subset of Σ^*.

$L \subseteq \Sigma^*$

Operations:-

1. Concatenation:-

Concatenation of two strings is a string formed by writing the first string followed by second string.

Ex:- $x = 0101$, $y = 1001$

$xy = 01011001$, $yx = 10010101$

→ It is not commutative.

$xy \neq yx$.

→ It is associative.

$(xy)z = x(yz)$

→ Identity element is ϵ.

$x\epsilon = \epsilon x = x$.

→ Cancellation property.

$zx = zy \Rightarrow x = y \Rightarrow$ left cancellation.

$xz = yz \Rightarrow x = y \Rightarrow$ right cancellation.

→ Let $x, y \in \Sigma^*$, then $|xy| = |x| + |y|$.

2. Transpose:-

For all $x \in \Sigma^*$, $a \in \Sigma$, then

$(a.x)^T = a.x^T$

Finite State Machines:-

The finite automaton is a mathematical modal of a system with discrete inputs and outputs.

→ It is used to know what is computable and what is not.

→ It is applied in compiler writing.

Finite Automata (FA)

- without output
  - Deterministic FA (DFA)
  - Non-deterministic FA (NFA)
  - NFA with ϵ moves.
- with output.
  - Mealey machine.
  - Moore machine.

Deterministic finite automata:-

A finite automata can be represented by 5 tuples, ie.,

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

Q → finite non-empty set of states.

Σ → finite non-empty set of input symbols.

δ → transition function.

$$\delta : Q \times \Sigma \rightarrow Q.$$

q_0 → initial state, $q_0 \in Q$.

F → set of final states.

$$F \subseteq Q.$$

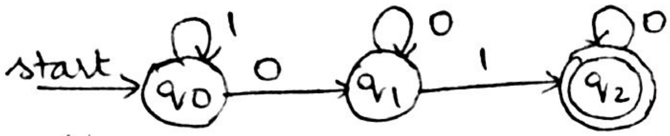
Ex:- ① $L = \{ w \mid w \text{ is of the form } x01y \}$

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

$x = \{ \epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots \}$

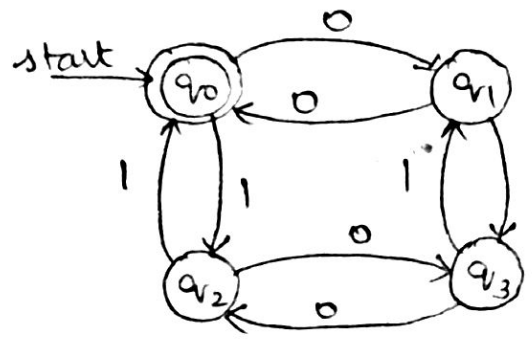
$y = \{ \epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots \}$

$L = \{ 01, 010, 001, 011, 101, \ldots \}$



② Even no. of 0's and Even no. of 1's.

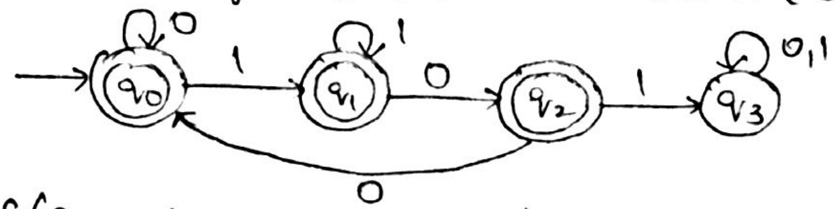
$L = \{ \epsilon, 00, 11, 0000, 1111, 0011, 1100, 0101, \ldots \}$



→ Transition diagram.

→ Transition table is a conventional tabular representation of transition function (δ).

Ex:-



$\delta(q_0, 0) \rightarrow q_0$ $\quad$ $\delta(q_0, 1) \rightarrow q_1$

$\delta(q_1, 0) \rightarrow q_2$ $\quad$ $\delta(q_1, 1) \rightarrow q_1$

$\delta(q_2, 0) \rightarrow q_0$ $\quad$ $\delta(q_2, 1) \rightarrow q_3$

$\delta(q_3, 0) \rightarrow q_3$ $\quad$ $\delta(q_3, 1) \rightarrow q_3$

Transition table:-

Q \ Σ	0	1
→ q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3	q_3	q_3

→ **Extendend Transition function:-**

It is a function which takes two arguments i.e., a state and an input string and returns only one value. denoted by $(\hat{\delta})$.

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

(i) $\hat{\delta}(q_0, \epsilon) \rightarrow q_0$

(ii) $\hat{\delta}(q_0, w) = \hat{\delta}(q_0, xa) = \delta(\hat{\delta}(q, x), a)$

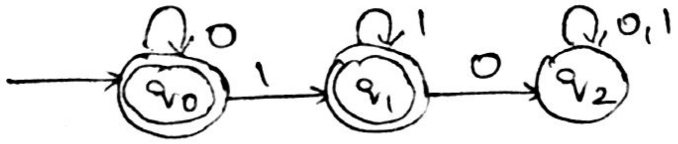
$= q_f$ (any state).

→ **Acceptance of a string:-**

A string 'w' is accepted by a DFA $M = \{Q, \Sigma, \delta, q_0, F\}$ if $\delta(q_0, w) = q_f$ where q_0 is initial state and q_f is final state.

Method-1:- Transition path:-

Ex:-



Let $w = 011$

$\rightarrow q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \in F$

∴ w is accepted.

let $w = 0100$.

$$\rightarrow q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_2 \notin F$$

$\therefore w$ is not accepted.

<u>Method - 2</u>:- Using transition function (δ).

Ex:- $\delta(q_0, 011) \overset{\text{moves to}}{\vdash} \delta(q_0, 11)$ $[\delta(q_0,0) \rightarrow q_0]$

$\vdash \delta(q_1, 1)$ $[\delta(q_0,1) \rightarrow q_1]$

$\vdash q_1 \in F$

$\delta(q_0, 010) \vdash \delta(q_0, 10)$

$\vdash \delta(q_1, 0)$

$\vdash q_2 \notin F$.

<u>Method - 3</u>:- Extended Transition function ($\hat{\delta}$).

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

$$\hat{\delta}(q_0, w) = \hat{\delta}(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = q$$

If $q \in F$, then w is accepted.

Ex:- $\hat{\delta}(q_0, 011)$.

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

$$\hat{\delta}(q_0, \epsilon 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(q_0, 0) = q_0$$

$$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_0, 1) = q_1$$

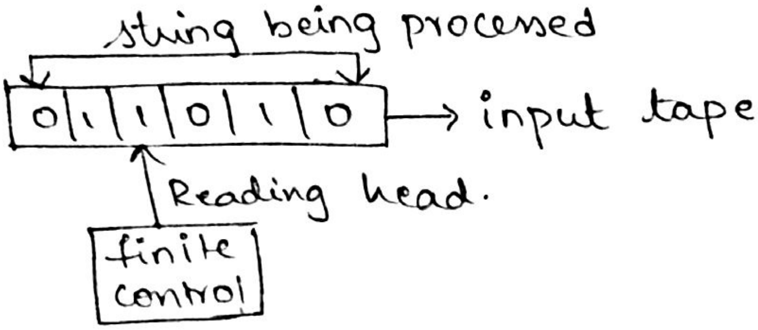
$$\hat{\delta}(q_0, 011) = \delta(\hat{\delta}(q_0, 01), 1) = \delta(q_1, 1) = q_1 \in F$$

$\therefore w$ is accepted.

Q: With the help of symatic diagram, explain the function of DFA. what are the reasons to say it is deterministic?

A: A DFA is a 5 tuple machine $M = \{Q, \Sigma, \delta, q_0, F\}$ is described as below:

We picture a FA as a finite control which is in some state from 'Q'. Reading a sequence of symbols from Σ, written on a tape as shown in the following figure.



1. **Input tape:-**

The input tape is divided into squares, each square containing single symbol from the input alphabet Σ.

2. **Reading head:-**

The head examines only one square at a time and can move one square either to the left (or) to the right. For further analysis, we restrict the movement of reading head only to the right side.

3. **Finite Control:-**

The input to the finite control will be usually symbol under the reading head.

Say a (or) 1 are the present state of machine 'q' to gives the following outputs.

* A motion of reading head along the tape to the next square.
* The next state (or) the final state machine is given by $\delta(q, a)$.

→ There are three methods for processing a machine.
1. Transition path.
2. Transition function.
3. Extended transition function.

→ There are two predefined notations for describing automata.
1. Transition diagram.
2. Transition table.

→ Reasons to state it is deterministic :-

1. for DFA, the outcome is a state that is an element of 'Q' but for NFA, the outcome is subset of Q.
2. DFA is in a single state after reading any sequence of inputs.
3. The term deterministic refers to the fact that on each input, there is one and only one state to which the automaton can move from its current state.

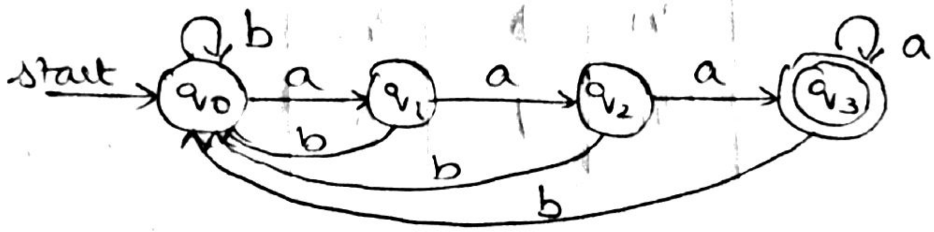
4. The smallest DFA can have 2^n states where as the smallest NFA (for the same language) has only 'n' states.

5. DFA has finite set of states and finite set of input symbols.

6. In a DFA, for a given input string 'w', and state 'q', there will be exactly one path labelled 'w' starting from 'q'.

7. To determine if the string is accepted by a DFA, it is sufficient to check this one path.

Ex:- Design a DFA which reads string from $\Sigma = \{a,b\}$ and ends with aaa.



Non-deterministic Finite Automata:-

An NFA can be represented by a 5 tuple $M = \{Q, \Sigma, \delta, q_0, F\}$ where

$Q \rightarrow$ finite non-empty set of states.

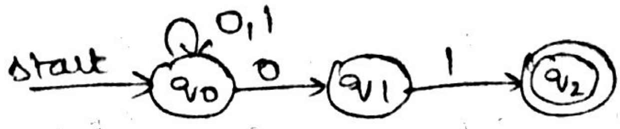
$\Sigma \rightarrow$ Alphabet.

$\delta \rightarrow$ Transition function.

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

$q_0 \rightarrow$ initial state.

$F \rightarrow$ set of final states.

Ex:- 

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

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

$\delta \Rightarrow \delta(q_0, 0) \to \{q_0, q_1\}$

$\delta(q_1, 0) \to \phi$

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

$\delta(q_0, 1) \to q_0$

$\delta(q_1, 1) \to q_2$

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

Q \ Σ	0	1
$\to q_0$	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	ϕ	ϕ

→ The entries of transition table in NFA will be set of states (or) black (ϕ).

→ A state can have more than zero transitions on input symbol 'a'.

→ <u>Extended Transition Function ($\hat{\delta}$)</u> :-

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

(ii) $\hat{\delta}(q_0, w) = \hat{\delta}(q_0, xa) = \delta(\hat{\delta}(q_0, x), a)$

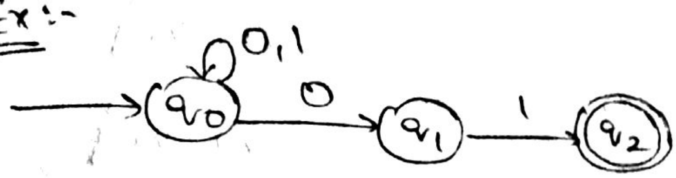
(iii) for each s, $S \subseteq Q$

$$\delta(s, a) = \bigcup_{\forall p \text{ in } s} \delta(p, a)$$

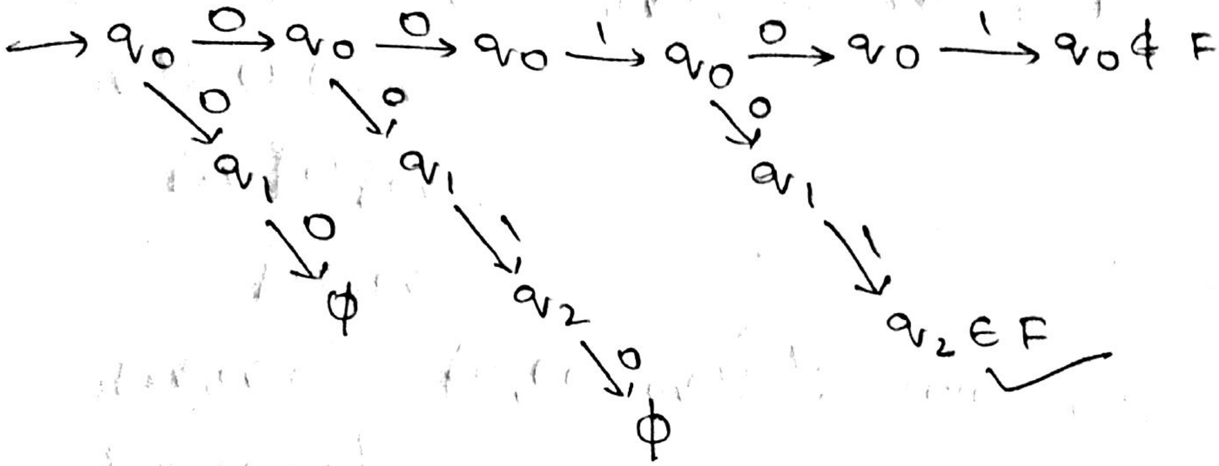
→ **Acceptance of a string by NFA:-**

1. **Transition path:-**

Ex:-

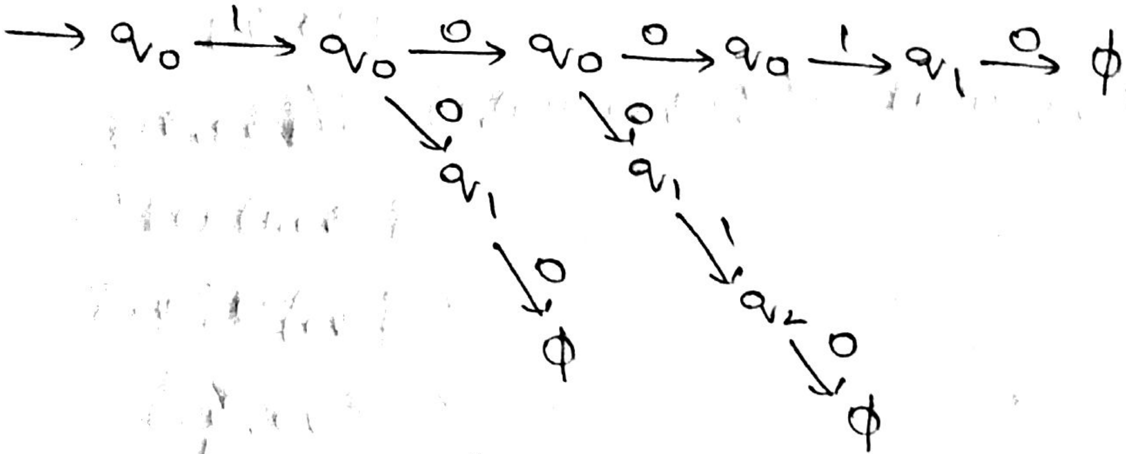


Let $\omega = 00101$



$\therefore$ ω is accepted.

Let $\omega = 10010$



$\therefore$ ω is not accepted.

2. **Extended Transition function ($\hat{\delta}$):-**

$\omega = 00101$

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

$\hat{\delta}(q_0, 0) = \hat{\delta}(q_0, \epsilon 0) = \delta(\hat{\delta}(q_0, \epsilon), 0)$

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

$$\hat{\delta}(q_0, 00) = \delta(\hat{\delta}(q_0, 0), 0) = \delta(\{q_0, q_1\}, 0)$$
$$= \delta(q_0, 0) \cup \delta(q_1, 0)$$
$$= \{q_0, q_1\} \cup \phi$$
$$= \{q_0, q_1\}$$

$$\hat{\delta}(q_0, 001) = \delta(\hat{\delta}(q_0, 00), 1) = \delta(\{q_0, q_1\}, 1)$$
$$= \delta(q_0, 1) \cup \delta(q_1, 1)$$
$$= \{q_0\} \cup \{q_2\}$$
$$= \{q_0, q_2\}$$

$$\hat{\delta}(q_0, 0010) = \delta(\hat{\delta}(q_0, 001), 0) = \delta(\{q_0, q_2\}, 0)$$
$$= \delta(q_0, 0) \cup \delta(q_2, 0)$$
$$= \{q_0, q_1\} \cup \phi$$
$$= \{q_0, q_1\}$$

$$\hat{\delta}(q_0, 00101) = \delta(\hat{\delta}(q_0, 0010), 1) = \delta(\{q_0, q_1\}, 1)$$
$$= \delta(q_0, 1) \cup \delta(q_1, 1)$$
$$= \{q_0\} \cup \{q_2\}$$
$$= \{q_0, \underset{\in F}{\underset{\downarrow}{q_2}}\}$$

$\therefore$ ω is accepted.

Language of a DFA:-

The language of DFA, $M = \{Q, \Sigma, \delta, q_0, F\}$ is denoted by $L(M)$ and is defined as

$$L(M) = \{ \omega \mid \hat{\delta}(q_0, \omega) = q_f \in F \}$$

ie., the language of M is a set of all strings ω, that take initial states q_0 to one to the final (or) accepting states.

$$L(M) = \{ \omega \in \Sigma^* \mid \hat{\delta}(q_0, \omega) \in F \}$$

→ If 'L' is L(M) for some DFA M, then 'L' is called regular language.

Language of an NFA:-

$$L(M') = \{ \omega \mid \hat{\delta}(q_0, \omega) \cap F \neq \phi \}$$

ie., $L(M')$ is a set of all states in Σ^* such that $\delta(q_0, \omega)$ contains atleast one final state of 'F'.

Conversion of NFA to DFA:-

Let the NFA be $M = \{Q, \Sigma, \delta, q_0, F\}$
and the DFA be $M' = \{Q', \Sigma', \delta', q_0', F'\}$

* M' is defined as follows:

1. $Q' = 2^Q$ i.e., Q' will have 2^Q states. If Q has 'n' states, then Q' will have 2^n states.

Note:- Not all the states are accessable from the start state of DFA. Inaccessable states can be thrown away so the no. of states of DFA can be much smaller than 2^n

* The i/p alphabet of two automata are same.
* $\delta'(q_1, a) = \delta(q_1, a)$

$$\delta'([q_0, q_1], a) = \delta(q_0', a) \cup \delta(q_1, a)$$

$$\delta'([q_1, q_2, \ldots q_i], a) = [P_1, P_2, \ldots P_j]$$

iff $\delta([q_1, q_2, \ldots q_j], a) = \{P_1, P_2 \ldots P_j\}$

4. $q_0' = [q_0]$

q_0 – state of NFA.

q_0' – state of DFA.

5. F': final state of DFA is a set of all states that include atleast one accepting states of NFA.

Steps to convert NFA to DFA:-

1. Write all transitions from initial state on every i/p symbol in Σ.

2. Repeat step-1 for every new state.

3. If a transition on some i/p symbol resultant in a set of states, then it is considered as a new single state.

4. Repeat step-2 and step-3 untill we do not get a new state.

5. The final states of equivalent DFA are all those states which consists of one accepting state of given NFA.

Q: Construct a DFA equivalent to

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

TT:

Q \ Σ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3	ϕ	$\{q_2\}$

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

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

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

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

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

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

$\delta(q_3, 0) = \phi$, $\delta(q_3, 1) = \{q_2\}$

$q_0 = q_0 \in Q$

$F = \{q_3\}$

Let M' be the DFA for the given NFA.

$$M' = \{Q', \Sigma_1, \delta', q_0', F'\}$$

$$Q' = 2^Q = \{\phi, [q_0], [q_1], [q_2], [q_3], [q_0, q_1], [q_1, q_2], [q_2, q_3], [q_0, q_2], [q_0, q_3], [q_1, q_3], [q_0, q_1, q_2], [q_0, q_1, q_3], [q_0, q_2, q_3], [q_1, q_2, q_3], [q_0, q_1, q_2, q_3]\}$$

$\Sigma_1 = \{0, 1\}$

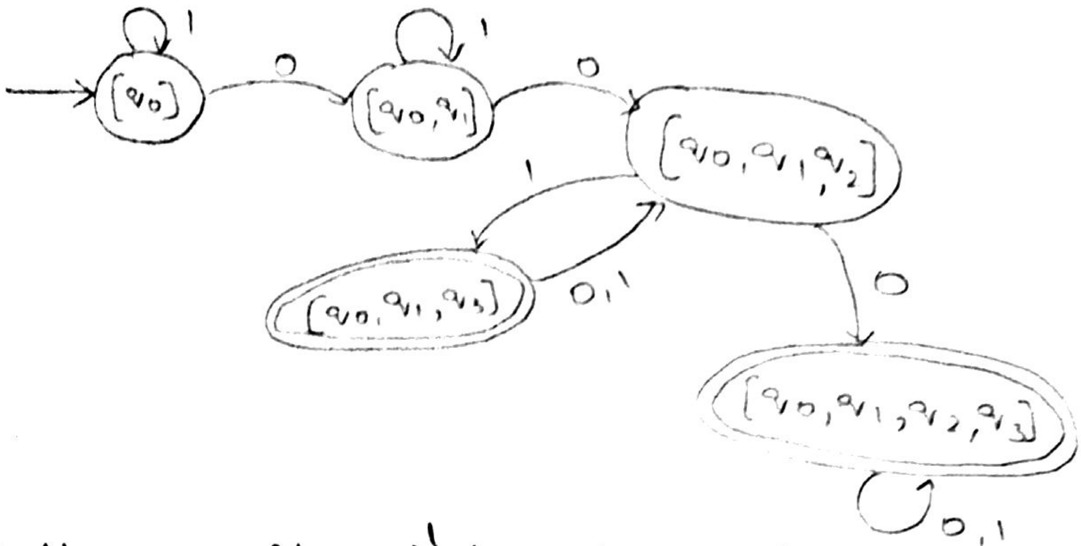
$\delta' : Q' \times \Sigma_1 = Q'$

$q_0' = \{q_0\}$

$$F' = \{[q_3], [q_2, q_3], [q_0, q_3], [q_1, q_3], [q_0, q_1, q_3], [q_1, q_2, q_3], [q_0, q_2, q_3], [q_0, q_1, q_2, q_3]\}$$

$Q \backslash \Sigma_1$	0	1
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_1]$	$[q_2]$	$[q_1]$
$[q_2]$	$[q_3]$	$[q_3]$
$[q_3]$	ϕ	$[q_2]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_2]$	$[q_0, q_1, q_3]$	$[q_0, q_3]$
$[q_0, q_3]$	$[q_0, q_1]$	$[q_0, q_2]$
$[q_1, q_2]$	$[q_2, q_3]$	$[q_1, q_3]$
$[q_1, q_3]$	$[q_2]$	$[q_1, q_2]$
$[q_2, q_3]$	$[q_3]$	$[q_2, q_3]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$

$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_1, q_2, q_3]$	$[q_2, q_3]$	$[q_1, q_2, q_3]$
$[q_0, q_2, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$



→ If we write δ' transitions for all i/ps on all i/p symbols, we got 30 transitions.

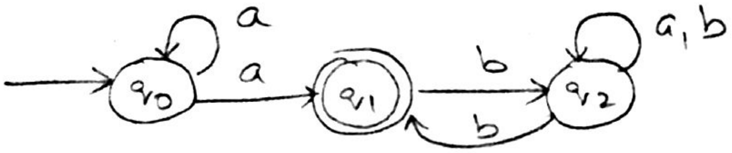
→ To reduce the no. of transitions, see the o/p state, if it is odd state, leave it otherwise for that new state, write the transitions on 0 & 1. Now the transition table for DFA is as follows:

Q \ Σ	0	1
$[q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$

Accessable state :-

If there is a path (direct (or) indirect) from initial state to any other state, then it is called accessable state. Otherwise it is inaccessable state.

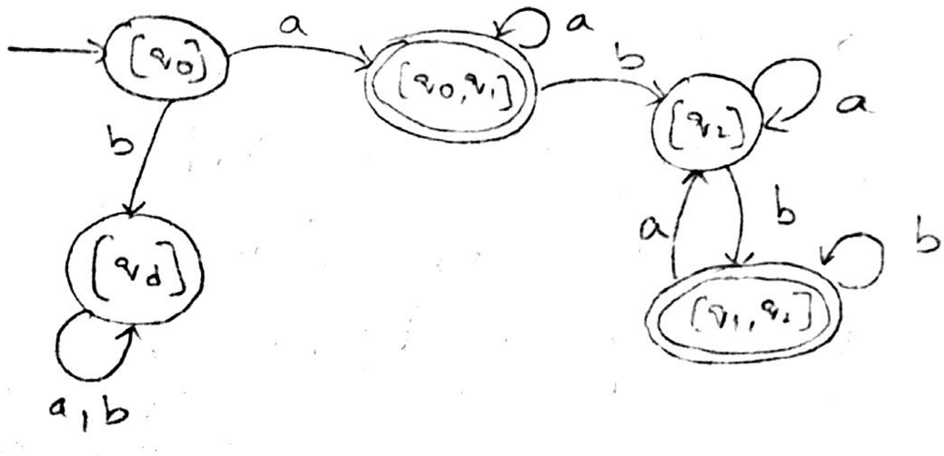
Q:



Q \ Σ	a	b
→ q_0	$\{q_0, q_1\}$	ϕ
q_1	ϕ	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_1, q_2\}$

$$Q' = \{\phi, [q_0], [q_1], [q_2], [q_0, q_1], [q_1, q_2], [q_0, q_2], [q_0, q_1, q_2]\}$$

Q \ Σ	a	b
→ $[q_0]$	$[q_0, q_1]$	$[q_d]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_2]$
$[q_2]$	$[q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_2]$	$[q_1, q_2]$
$[q_d]$	$[q_d]$	$[q_d]$



Non-deterministic finite Automata with ε-moves:-

NFA-ε is given by $M = \{Q, \Sigma, \delta, q_0, F\}$

$Q \rightarrow$ a finite set of states.

$\Sigma \rightarrow$ alphabet including ε.

$\delta \rightarrow$ Transition function.

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

$q_0 \rightarrow$ initial state.

$F \rightarrow$ set of final states.

→ **Acceptance of strings:-**

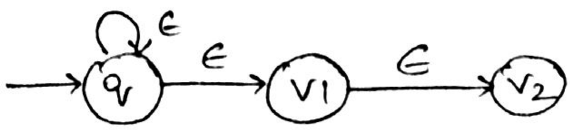
* **Transition path:-**

The input symbols includes 'ε' although they donot appear explicitly in 'w'.

→ **ε-closure (q):-**

We use ε-closure (q) to denote the set of all vertices 'v' such that there is a path from q to v labelled ε (direct or indirect)

Ex:-



$$\epsilon\text{-closure}(q) = \{q, v_1, v_2\}$$

Extended transition function ($\hat{\delta}$) :-

In order to extend δ to $\hat{\delta}$,

$$\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q.$$

(i) $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-closure}(q_0)$.

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

$= \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), a))$.

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

(iii) for ω in Σ^* and a in Σ,

$\hat{\delta}(q_0, \omega a) = \epsilon\text{-closure}(P)$,

where $P = \{p \mid$ for some r, in $\hat{\delta}(q, \omega)$ and p in $\delta(r, a)\}$.

(iv) $\hat{\delta}(s, a) = \bigcup_{q \text{ in } s} \hat{\delta}(q, a)$; $\forall q \in s$.

(v) $\hat{\delta}(s, \omega) = \bigcup_{q \text{ in } s} \hat{\delta}(q, \omega)$; $\forall q \in s$.

Procedure to convert NFA-ϵ to NFA :-

Let M be NFA-ϵ and M' be NFA.

(i) Find ϵ-closure (q) for all states of M.

(ii) Determine the extended transition funct^n ($\hat{\delta}$) as follows:

* $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-closure}(q_0)$.
* $\hat{\delta}(q_0, a) = \hat{\delta}(q_0, \epsilon a)$
  $= \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), a))$
  $= \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), a))$
* $\epsilon\text{-closure}(q_1, q_2, \ldots q_n) = \epsilon\text{-closure}(q_1) \cup \epsilon\text{-closure}(q_2) \cup \ldots$

☆(iii) Final states of M' include all states whose ϵ-closure contains a final state.

Direct conversion from NFA-ϵ to DFA:-

Procedure:-

1. Find ϵ-closure(q) for all states of NFA-ϵ.
2. $q_0' = \epsilon$-closure(q_0) where q_0 – initial of NFA-ϵ (M)
   & q_0' – initial of DFA (M')
3. Calculate the δ transitions using the following formulae:

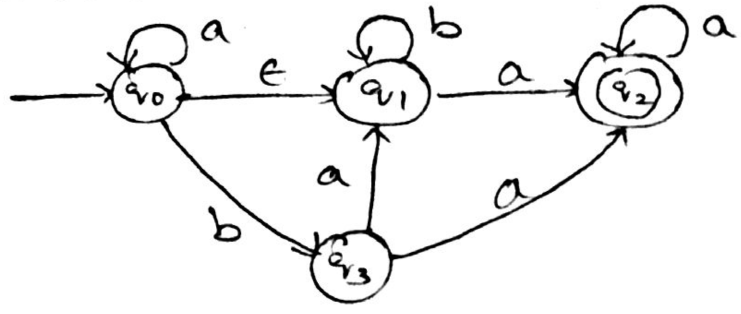
$$\delta'(q_0', a) = \epsilon\text{-closure}\left(\delta(q_0, a)\right).$$

$$\delta'([q_0, q_1, \dots], a) = \epsilon\text{-closure}\left(\delta\{q_0, q_1, \dots q_n\}, a\right)$$

4. Write all transitions from initial state on every i/p symbol Σ.
5. Repeat step-4 for each new state obtained in step-4.

6. F' is a set of all states of DFA consisting of atleast one final state of M.

Q: Convert NFA-ϵ to DFA.



Sol:

Q \ Σ	a	b	ϵ
$\rightarrow q_0$	$\{q_0\}$	$\{q_3\}$	$\{q_1\}$
q_1	$\{q_2\}$	$\{q_1\}$	—
q_2	$\{q_2\}$	—	—
q_3	$\{q_1, q_2\}$	—	—

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

ϵ-closure $(q_0) = \{q_0, q_1\}$

ϵ-closure $(q_1) = \{q_1\}$

ϵ-closure $(q_2) = \{q_2\}$

ϵ-closure $(q_3) = \{q_3\}$

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

Q \ Σ	a	b
$\rightarrow [q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_1, q_3]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_1, q_3]$
$[q_1, q_3]$	$[q_1, q_2]$	$\{q_1\}$

$$\hat{\delta}'([q_0, q_1], a) = \epsilon\text{-closure}\left(\delta([q_0, q_1], a)\right)$$

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

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

$$\hat{\delta}[[q_0, q_1], b] = \epsilon\text{-closure}\left(\hat{\delta}'([q_0, q_1]), b\right)$$

$$= \epsilon\text{-}c[q_1, q_3]$$

$$= [q_1, q_3]$$

$$\hat{\delta}([q_0, q_1, q_2], a) = \epsilon\text{-}c\left[\delta([q_0, q_1, q_2]), a\right]$$

$$= \epsilon\text{-}c[\{q_0, q_2\}]$$

$$= [q_0, q_1, q_2]$$

$$\hat{\delta}([q_0, q_1, q_2], b) = \epsilon\text{-}c\left[\delta([q_0, q_1, q_2]), b\right)$$

$$= \epsilon\text{-}c[\{q_1, q_3\}]$$

$$= [q_1, q_3]$$

$$\hat{\delta}([q_1, q_3], a) = \epsilon\text{-}c\left[\delta([q_1, q_3]), a\right)$$

$$= \epsilon\text{-}c[\{q_1, q_2\}, a)$$

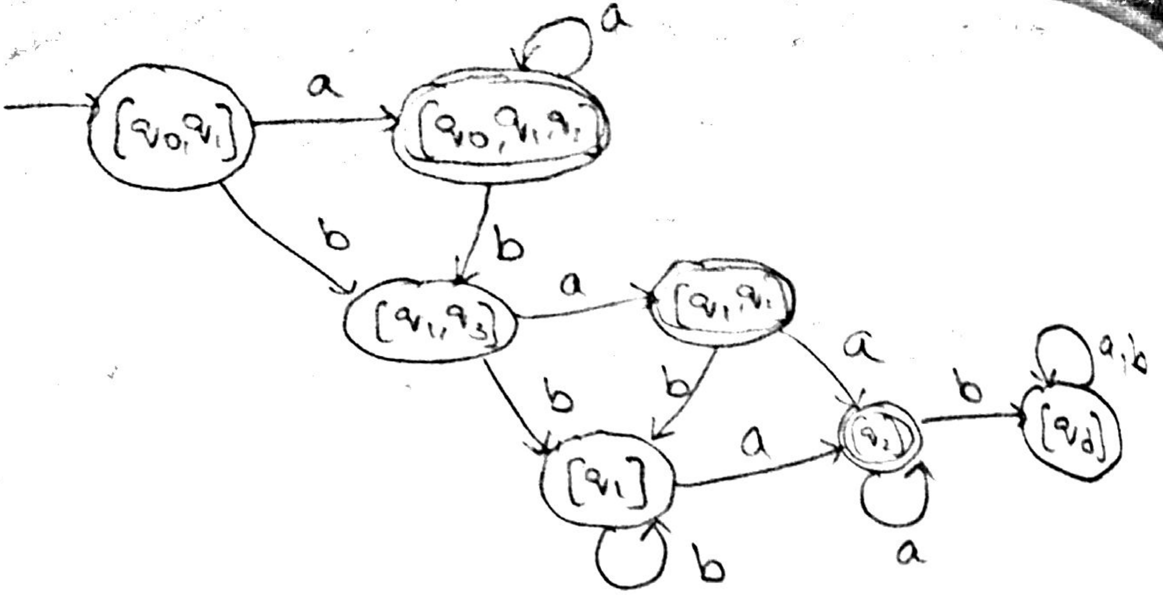
$$= [q_p, q_2]$$

$$\hat{\delta}([q_1, q_3], b) = \epsilon\text{-}c\left[\delta([q_1, q_3], b)\right]$$

$$= \epsilon\text{-}c\begin{bmatrix}\{q_1\} \\ \cancel{\phi}\end{bmatrix}$$

$$= \cancel{\phi}\ \{\cancel{q},\ q_1\}$$

$[q_1, q_2]$	$[q_2]$	$[q_1]$
$[q_1]$	$[q_2]$	$[q_1]$
$[q_2]$	$[q_2]$	$[q_d]$
$[q_d]$	$[q_d]$	$[q_d]$



10/1/13 ||

Minimization of finite Automata (DFA):-

Procedure:-

1. Prepare the transition table for the given DFA M.

2. Consider 'Q' of M and partition Q into set of final and non-final states.

3. Take the set of final and non-final states as group-I & group-II. ~~Consider each~~

4. Consider each group and write transitions for each group on each i/p symbol.

5. Partition the group when on an i/p symbol, if the states of a group goes to a different group.

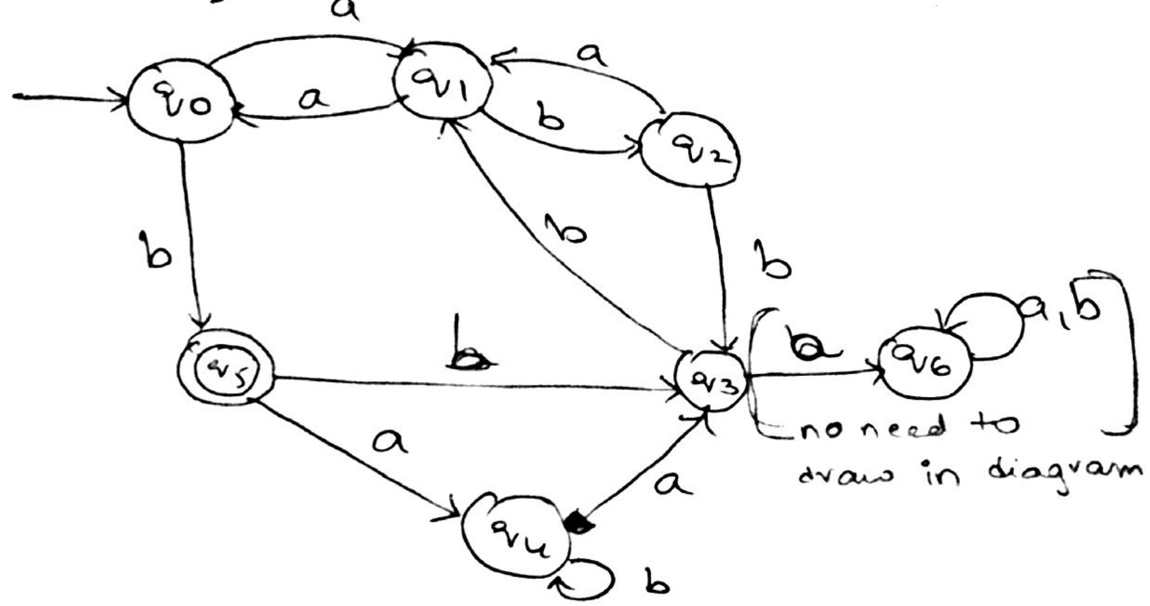
6. Don't partition the group when on an i/p symbol, if the states of a group goes to the state group.

7. Repeat steps-5 & 6 till there is no partitioning.

8. Finally we will get the minimized DFA M'

$M' = \{Q', \Sigma, \delta', q_0', F'\}$, which is equivalent to the given DFA.

Q: Minimize the finite automaton:



$Q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_1	q_5
q_1	q_0	q_2
q_2	q_1	q_3
q_3	q_6	q_1
q_4	q_3	q_4
q_5	q_4	q_3
q_6	q_6	q_6

In the above table, there is no transition for (q_3, a) but in DFA, for each and every symbol of Σ should have one and only one transition to the other state. So we are introducing new state q_6 called dead state.

$G1 - q_5$

$G2 - q_0, q_1, q_2, q_3, q_4, q_6$

G2:- $\left.\begin{array}{l} \delta(q_0,a) \to q_1 \\ \delta(q_1,a) \to q_0 \\ \delta(q_2,a) \to q_1 \\ \delta(q_3,a) \to q_6 \\ \delta(q_4,a) \to q_3 \\ \delta(q_6,a) \to q_6. \end{array}\right\} \in G2$

$\because$ all belongs to same group, consider 'b'.

$\delta(q_0,b) \to q_5 \to G1$

$\left.\begin{array}{l} \delta(q_1,b) \to q_2 \\ \delta(q_2,b) \to q_3 \\ \delta(q_3,b) \to q_1 \\ \delta(q_4,b) \to q_4 \\ \delta(q_6,b) \to q_6 \end{array}\right\} \in G2$

$\therefore$ $\underset{G_1}{\boxed{q_5}}$ $\underbrace{\underset{G2}{\boxed{q_0}} \quad \underset{G3.}{\boxed{q_1 \mid q_2 \mid q_3 \mid q_4 \mid q_6}}}_{}$

since, $(\delta 0, b) \to q_5$, which is in different group.

G_3:-

$$\delta(q_1, a) \rightarrow q_0 \in G_2$$

$$\left.\begin{array}{l}\delta(q_2, a) \rightarrow q_1 \\ \delta(q_3, a) \rightarrow q_6 \\ \delta(q_4, a) \rightarrow q_3 \\ \delta(q_6, a) \rightarrow q_6\end{array}\right\} \in G_3.$$

$$\underset{G_1}{\boxed{q_5}} \quad \underset{G_2}{\boxed{q_0}} \quad \underset{G_3}{\boxed{q_1}} \quad \underset{G_4}{\boxed{q_2 \mid q_3 \mid q_4 \mid q_6}}$$

Here, no need ~~to~~ of writing transitions on 'b' as we get partition on 'a'.

G_4:-

$$\delta(q_2, a) \rightarrow q_1 \in G_3$$

$$\left.\begin{array}{l}\delta(q_3, a) \rightarrow q_6 \\ \delta(q_4, a) \rightarrow q_3 \\ \delta(q_6, a) \rightarrow q_6\end{array}\right\} G_4$$

$$\underset{G_1}{\boxed{q_5}} \quad \underset{G_2}{\boxed{q_0}} \quad \underset{G_3}{\boxed{q_1}} \quad \underset{G_4}{\boxed{q_2}} \quad \underset{G_5}{\boxed{q_3 \mid q_4 \mid q_6}}$$

G_5:-

$$\left.\begin{array}{l}\delta(q_3, a) \rightarrow q_6 \\ \delta(q_4, a) \rightarrow q_3 \\ \delta(q_6, a) \rightarrow q_6\end{array}\right\} \in G_5 \text{ (same grp).}$$

$$\delta(q_3, b) \rightarrow q_1 \in G_3$$

$$\left.\begin{array}{l}\delta(q_4, b) \rightarrow q_4 \\ \delta(q_6, b) \rightarrow q_6\end{array}\right\} \in G_5.$$

$$\underset{G_1}{\boxed{q_5}} \quad \underset{G_2}{\boxed{q_0}} \quad \underset{G_3}{\boxed{q_1}} \quad \underset{G_4}{\boxed{q_2}} \quad \underset{G_5}{\boxed{q_3}} \quad \underset{G_6}{\boxed{q_4 \mid q_6}}$$

<u>G_6</u> :- $\delta(q_4, a) \rightarrow q_3 \in G_5$

$\delta(q_6, a) \rightarrow q_6 . \in G_6.$

$\boxed{q_5}$	$\boxed{q_0}$	$\boxed{q_1}$	$\boxed{q_2}$	$\boxed{q_3}$	$\boxed{q_4}$	$\boxed{q_6}$
G_1	G_2	G_3	G_4	G_5	G_6	G_7

Here the no. of states and no. of groups are same. ∴ the no. of states for given DFA = 7. The no. of states for reduced DFA = 7.

∴ The given DFA is the reduced DFA.

Q.

Finite Automata with outputs :-

→ DFA, NFA, NFA−ε have binary outputs ie., they accept (or) reject the string.

→ This accessability is based on final state and initial state. Now, we consider the model where the outputs can be choosen from some other alphabet.

→ The value of the output function $Z(t)$ is a function of the present state $q(t)$ and present input $x(t)$.

$$Z(t) = \lambda(q(t), x(t)).$$

This is called mealy machine.

→ If the output function $Z(t)$ depends only on the present state $q(t)$ and is independent on the current input, then the output function may be written as

$$Z(t) = \lambda[q(t)].$$

This is called moore machine.

Moore machine :-

It is a 6 tuple form,

$M = \{Q, \Sigma, \Delta, \delta, \lambda, q_0\}$

$Q \rightarrow$ set of all states.

$\Sigma \rightarrow$ alphabet (input)

$\Delta \rightarrow$ output alphabet.

$\delta \rightarrow$ Transition function. $\delta : Q \times \Sigma \rightarrow Q$

$\lambda \rightarrow$ Output function.

$\lambda : Q \longrightarrow \Delta$

$q_0 \rightarrow$ initial state.

Mealy machine:-

$$M = \{Q, \Sigma, \Delta, \delta, \lambda, q_0\}$$

$Q \rightarrow$ set of states.

$\Sigma \rightarrow$ input alphabet.

$\Delta \rightarrow$ output alphabet.

$\delta \rightarrow$ Transition function.

$\delta : Q \times \Sigma \longrightarrow Q$

$\lambda \rightarrow$ Output function

$\lambda : Q \times \Sigma \longrightarrow \Delta$

$q_0 \rightarrow$ initial state.

28/1/13

Conversion of moore to mealy machine :-

Procedure :-

1. Define the o/p funct^n. λ' for the mealy machine as follows:

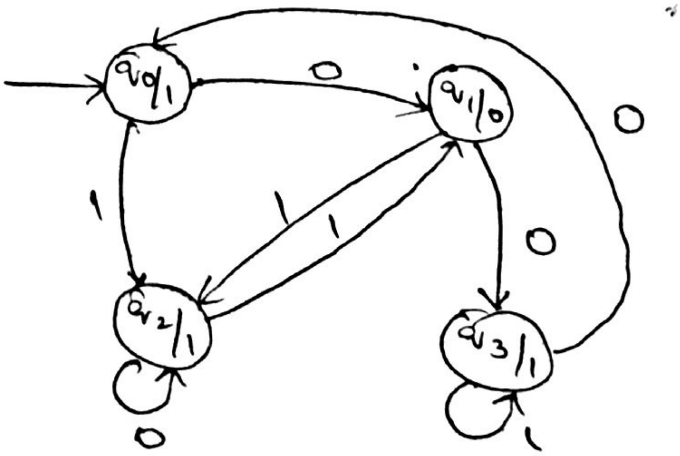
$$\lambda'(q, a) = \lambda(\delta(q, a)),$$

for all states and input symbols.

2. The transition funct^n (δ) for the mealy is same as that of the moore machine.

Q:

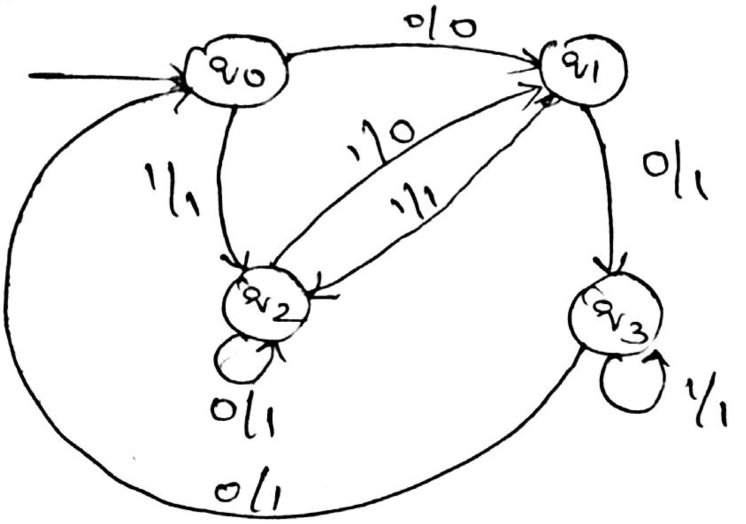
PS (q_i)	NS (q_{i+1}) $a=0$	NS (q_{i+1}) $a=1$	λ
→ q_0	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_1	1
q_3	q_0	q_3	1



$$\lambda'(q_0, 0) = \lambda(\delta(q_0, 0)) = \lambda(q_1) = 0$$
$$\lambda'(q_0, 1) = \lambda(\delta(q_0, 1)) = \lambda(q_2) = 1$$
$$\lambda'(q_1, 0) = \lambda(\delta(q_1, 0)) = \lambda(q_3) = 1$$
$$\lambda'(q_1, 1) = \lambda(\delta(q_1, 1)) = \lambda(q_2) = 1$$

$$\lambda'(q_2,0) = \lambda(\delta(q_2,0)) = \lambda(q_2) = 1$$
$$\lambda'(q_2,1) = \lambda(\delta(q_2,1)) = \lambda(q_1) = 0$$
$$\lambda'(q_3,0) = \lambda(\delta(q_3,0)) = \lambda(q_0) = 1$$
$$\lambda'(q_3,1) = \lambda(\delta(q_3,1)) = \lambda(q_3) = 1$$

PS	NS			
	$a=0$	λ'	$a=1$	λ'
$\rightarrow q_0$	q_1	0	q_2	1
q_1	q_3	1	q_2	1
q_2	q_2	1	q_1	0
q_3	q_0	1	q_3	1



Note:-

Acceptance of the null input string (ε):-

(i) In case of moore machine, it accepts the null i/p string (ε).

(ii) In the case of mealy machine, we get an o/p only on the application of an i/p symbol. So, for the i/p string ε, the o/p is also 'ε'. ∴ It does not accept 'ε'.

Conversion of mealy to moore:-

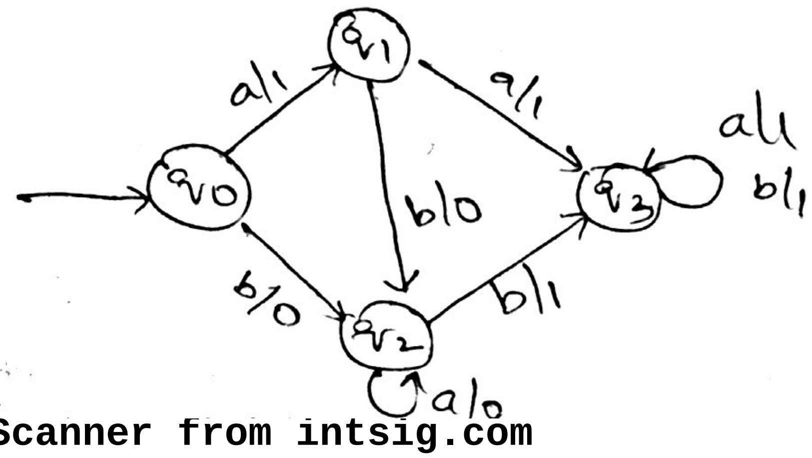
Procedure:-

1. For any state q_i, determine the no. of different o/ps 'n' associated with "q_i" in the next state column.
2. Split q_i into 'n' different states.
3. Construct the transition table for new state.
4. Arrange the pair of states and o/p's in the next state column. The resulting table is the moore machine.
5. In the moore machine, if the o/p of the initial state q_0 is 1, then add a new state say, "q" with the same transitions as "q_0" whose o/p is zero. The transition table is the moore machine.

Note:-

In the mealy machine, in the next state column, some states are associated with different o/ps. Then split the states which are associated with different o/ps.

Q:



PS	NS: a	λ	NS: b	λ
→ q_0	q_1	1	q_2	0
q_1	q_3	1	q_2	0
q_2	q_2	0	q_3	1
q_3	q_3	1	q_3	1

q_0 is associated with o/p symbol 'ε'.

$q_1 \rightarrow 1$ (single o/p symbol)

$q_2 \rightarrow 0$

$q_3 \rightarrow 1$

Here, we can observe that all the states are associated with single o/p symbol. ∴ there is no need to split the state. The required moore machine is

PS	NS: a	NS: b	λ
→ q_0	q_1	q_2	ε
q_1	q_3	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_3	1