TOC

notes by :- Sonam

DATE
PAGE

Finite Automation :-

↳ abstract computing device.

↳ mathematical model of a system with discrete I/P's, O/P's, states & set of transitions from state to state that occurs on I/P symbols from alphabet Σ.

Representation :-

↳ Graphical (Trans. Diag)
↳ Tabular (Trans. Table)
↳ Mathematical (Trans. func.)

Finite Automation formal Def. :- consists of 5 tuples.

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

where :-

$Q \rightarrow$ set of states
$\Sigma \rightarrow$ set of (finite) alphabets
$\delta \rightarrow (Q \times \Sigma \rightarrow Q)$ transition func.
$q_0 \rightarrow$ initial state
$F \rightarrow$ set of accepted or finite states.

DATE
PAGE

Powers of Sigma :-

$\Sigma^0 \rightarrow$ Set of all strings with length '0' = $\{\lambda, \epsilon\}$

$\Sigma^1 \rightarrow$ " " " " " '1' = $\{a, b\}$

$\Sigma^2 \rightarrow$ " " " " " '2' = $\{aa, ab, ba, bb\} - 2^2$

Σ^3

:

$\Sigma^* \rightarrow$ (Kleene Closure) $\rightarrow$ set of all strings of all possible length.

Eg:- $(a+b)^* \rightarrow$ Infinite language.

$\Sigma^+ \rightarrow$ (Positive Closure) $\rightarrow$ all strings possible except epsilon

$\Rightarrow \boxed{\Sigma^* = \Sigma^+ + \Sigma^0}$ ☆

$\Rightarrow \boxed{\Sigma^+ = \Sigma^* - \Sigma^0}$ ☆

null is also known as identity element.

Grammar :-

A grammar is defined as quadruple.

$$G = \{V, T, P, S\}$$

$V \rightarrow$ Variable
$T \rightarrow$ Terminal
$P \rightarrow$ Production Rule
$S \rightarrow$ Start

eg:- $S \rightarrow aSb / \epsilon$

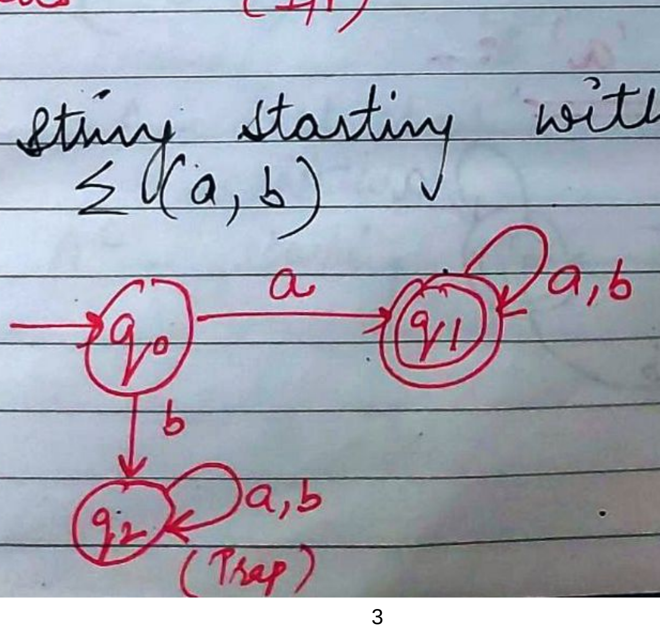
Generated strings could be
ϵ, aSb, $aaSb$, ab, $aaSbb$
$aaaSbbb$, $a^n b^n \ldots a^n b^n$ where $n \geq 0$

DFA (Deterministic Finite Automata)

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

Q → set of finite states
Σ → a, b (I/P)
δ → transitions
q_0 → start
F → set of final states.

Eg: string starting with a in $\Sigma(a, b)$



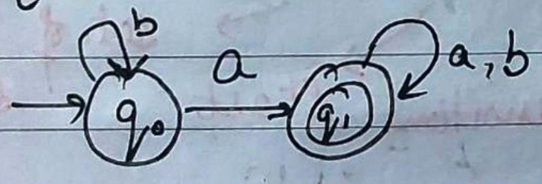
δ :- $\boxed{Q \times \Sigma \rightarrow Q}$

$q_0 \quad (a, b)$
q_1
q_2

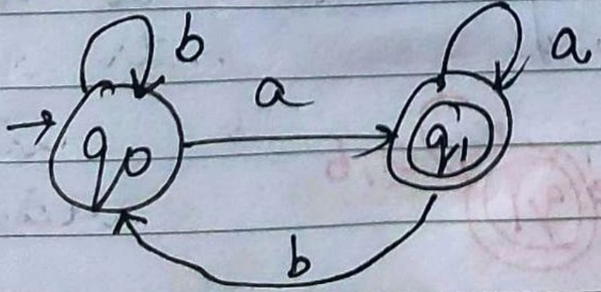
$q_0\ a \rightarrow q_1$
$q_0\ b \rightarrow q_2$
$q_1\ a \rightarrow q_1$
$q_1\ b \rightarrow q_1$
$q_2\ a \rightarrow q_2$
$q_2\ b \rightarrow q_2$

Ques:-
Construct a DFA which accept a language of all strings containing 'a'. $\Sigma(a, b)$

$\Rightarrow \{a, aa, aaa, ba, ab \ldots\}$



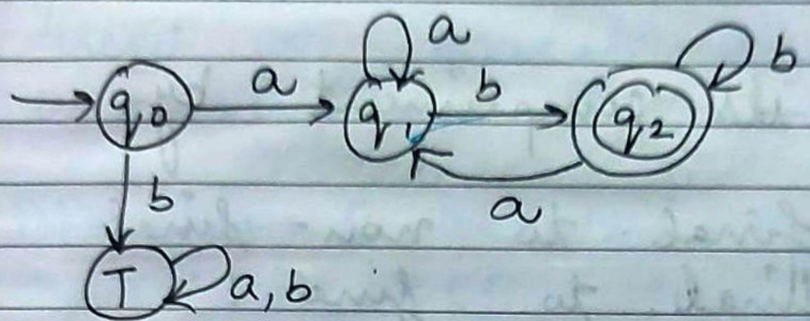
$\Rightarrow$ End with 'a' :-



DATE
PAGE

⇒ Const. a DFA which accept a lang. of all strings starting with 'a' and ending with 'b'.

L = {ab, abab, ababab, aaabbb ...



⇒ not starting with 'a' or not ~~#~~ ending with 'b'.

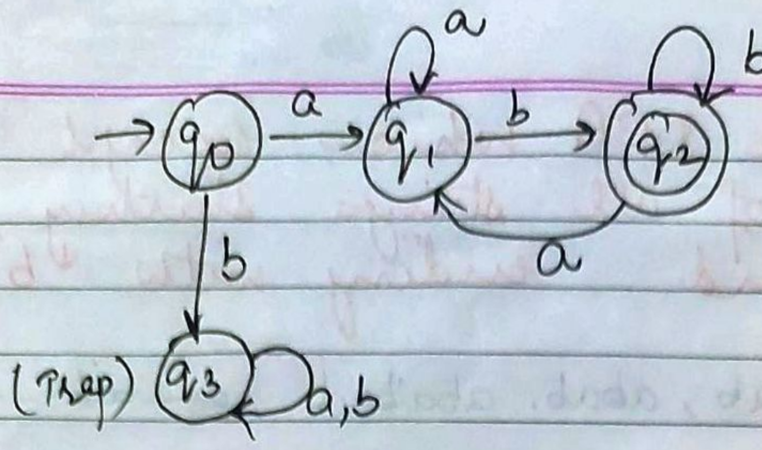
L = {ε, a, b, ba}...

⇒ We should use compliment here by using De-morgan theorem.

$$(A \cup B)^c = \underbrace{A^c}_{} \cap \underbrace{B^c}_{}$$

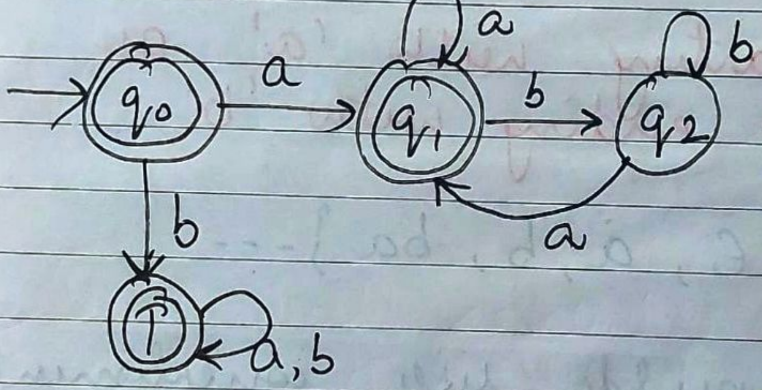
A = not starting with a
B = not ~~starting~~ ending with b

A^c = starting with a
B = ending with b



Starting with a & ending with b

So take its compliment by making
final to non-final
non-final to final.



Properties of DFA:-

- A finite automata is called deterministic finite automata if the machine reads an I/P string one symbol at a time.
- There is only one path for specific I/P from current state to the next state.

DATE
PAGE

↳ DFA does not accept the null move i.e DFA cannot change state without any I/P character.

↳ DFA can contain multiple final states.

↳ Used in lexical analysis phase in compiler.

DFA has 5 tuples :-

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

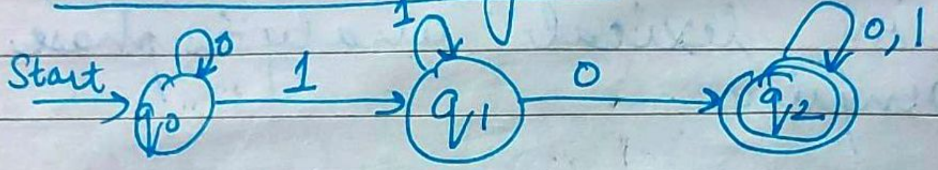
$Q \rightarrow$ finite set of states
$\Sigma \rightarrow$ " " " I/P symbols
$q_0 \rightarrow$ initial state
$F \rightarrow$ Final state
$\delta \rightarrow$ Transition function.
$(\delta : Q \times \Sigma \rightarrow Q)$

Graphical Representation :-

⇒ State → vertices
⇒ arc with I/P char → transitions
⇒ initial state is marked with an arrow.
⇒ final state is denoted by double circle.

Eg:- $Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
$F = \{q_2\}$

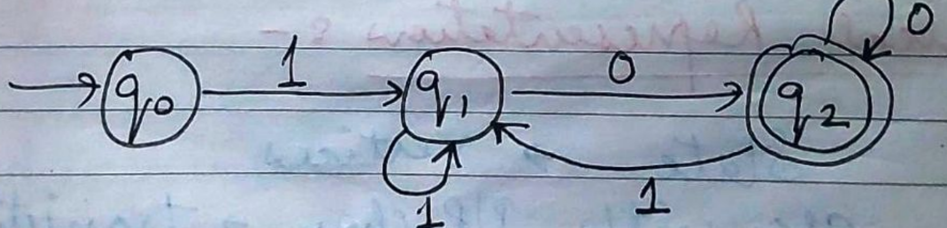
Transition Diagram :-



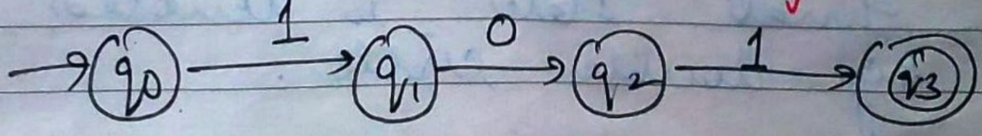
Transition Table:-

	0	1
→ q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

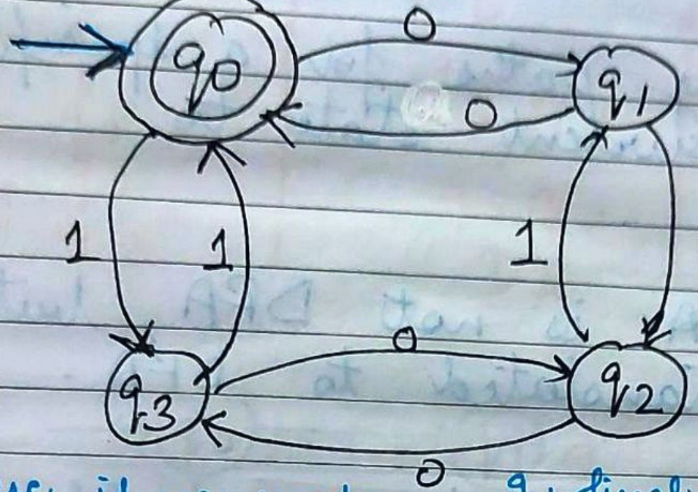
Eg:- ① FA with $\Sigma = \{0, 1\}$ accepts the string that start with 1 and ends with 0.



② FA that accepts only I/P 101.



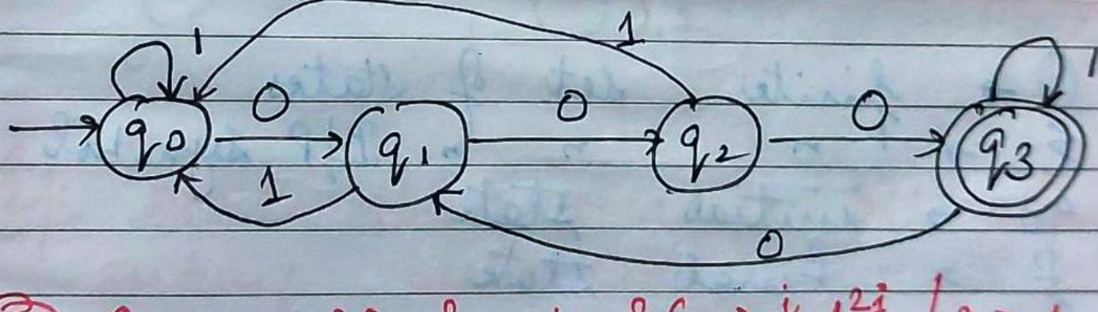
(3) FA with even no. of 0's & even no. of 1's.



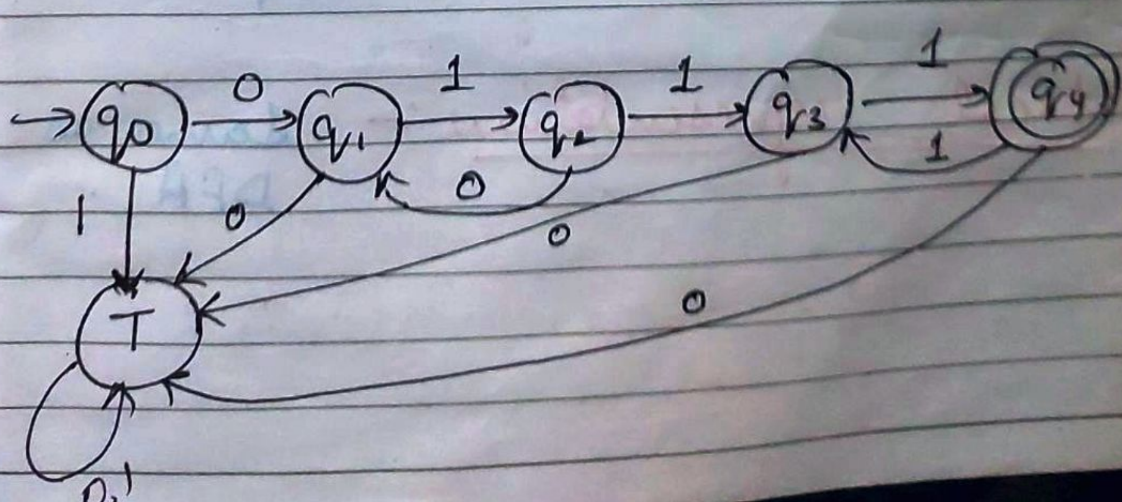
q0 will be both start & final state.

Here if we make q1 final → odd 0's & even 1's
q3 " → even 0's & odd 1's
q2 " → odd 0's & odd 1's

(4) FA with all strings with 3 consecutive 0's.



(5) Design FA for $L = \{(01)^i 1^{2j} \mid i \geq 1, j \geq 1\}$
means string should start with 01 & end with even no. of 1's.



NFA or NDFA

(Non-Deterministic Finite Automata)

⇒ Here many paths for a specific I/P from current state to next state.

⇒ Every NFA is not DFA, but can be translated to DFA.

⇒ contains ε transitions or null moves.

NDFA has 5 tuples :-

Q → finite set of states
Σ → " " " I/P symbol
q_0 → initial state
F → final state
δ → transition function

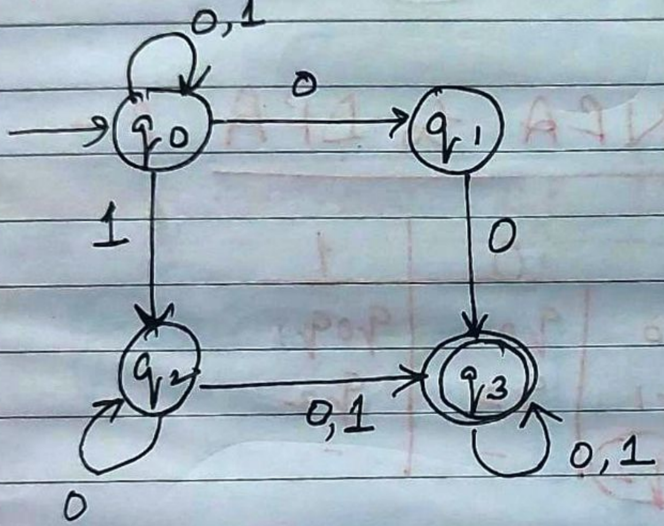
$$\boxed{\delta : Q \times \Sigma \rightarrow 2^Q}$$

Graphical representation :- same as DFA.

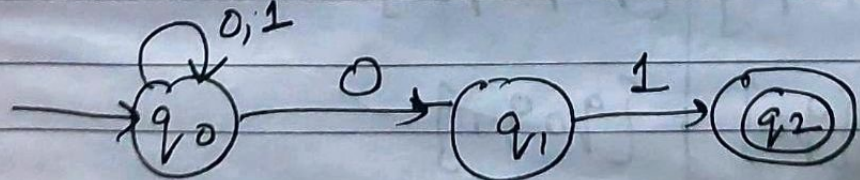
DATE
PAGE

Eg → Design an NFA for transition table:-

I/P	0	1
→ q_0	$q_0 \, q_1$	$q_0 \, q_2$
q_1	q_3	ε
q_2	$q_2 \, q_3$	q_3
q_3	q_3	q_3



Eg: NFA that accepts all strings ending with 01:

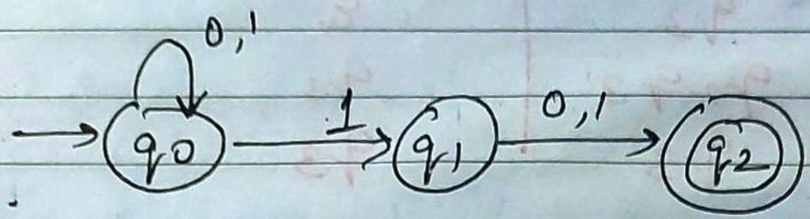


DATE________
PAGE________

Q:- NFA of binary string in which 2nd last bit is 1.

10
11

$$(0+1)^* \, 1 \, (0+1)$$



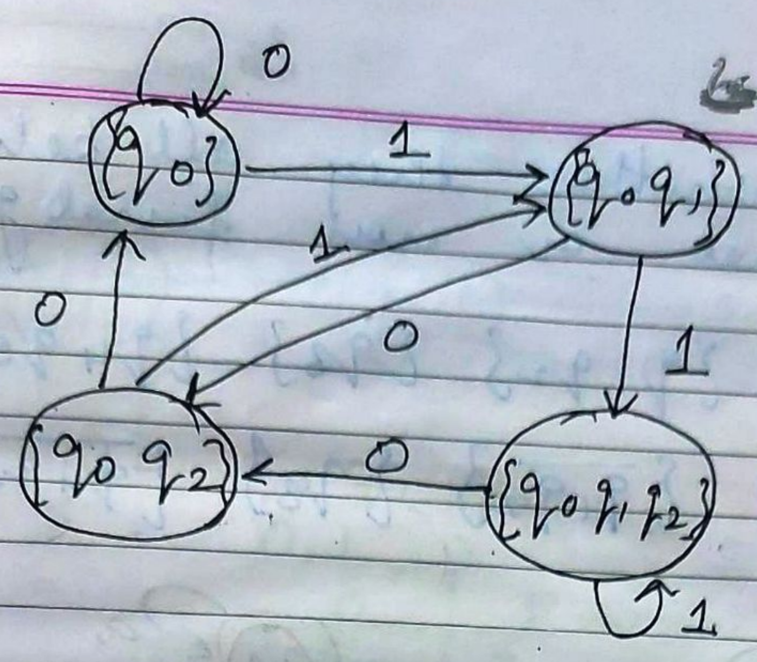
Conversion of NFA to DFA :-

Given :-

	0	1
→ q_0	q_0	$q_0 q_1$
q_1	q_2	q_2
q_2	–	–

	0	1
→ q_0	$\{q_0\}$	$\{q_0 q_1\}$
$q_0 q_1$	$\{q_0 q_2\}$	$\{q_0 q_1 q_2\}$
$q_0 q_2^*$	$\{q_0\}$	$\{q_0 q_1\}$
$q_0 q_1 q_2^*$	$\{q_0 q_2\}$	$\{q_0 q_1 q_2\}$

The states which will have q_2 would be treated as final states in NFA.



Resultant DFA

Minimization of DFA :-

Given :-

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
q_3^*	q_3	q_1
q_4^*	q_4	q_5
q_5^*	q_5	q_4

Separate the final & non final states in different groups and label them as classes represented by π.

$\pi_0 = \{q_0 q_1 q_2\} \; \{q_3 \; q_4 \; q_5\}$

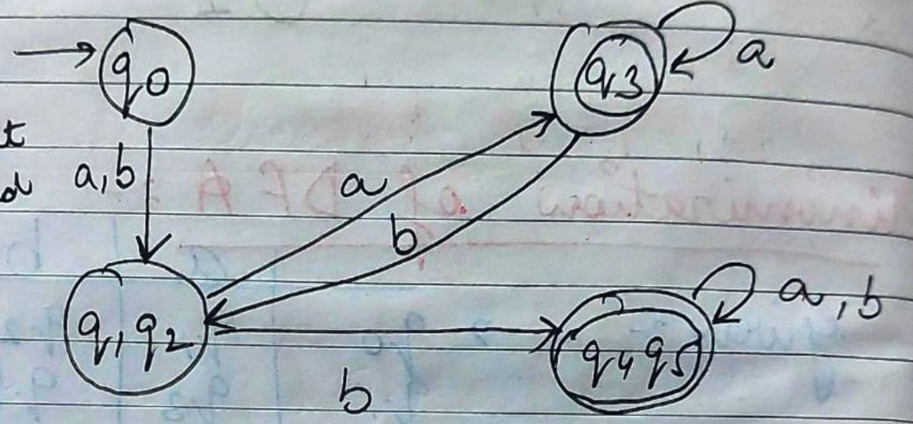
Chk the equivalance of states by checking each state gp with other.

Chk whether they all belong to final or non final grp.

$\pi_1 = \{q_0\}\ \{q_1, q_2\}\ \{q_3\}\ \{q_4 q_5\}$

$\pi_2 = \{q_0\}\ \{\overline{q_1 q_2}\}\ \{q_3\}\ \{\overline{q_4 q_5}\}$

Resultant Minimized DFA



Moore Machines :- called finite automata with O/P.

$M_0 = \{Q, \Sigma, \Delta, \delta, \lambda, q_0\}$

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ I/P symbol

$\delta \rightarrow$ transition func. $Q \times \Sigma$

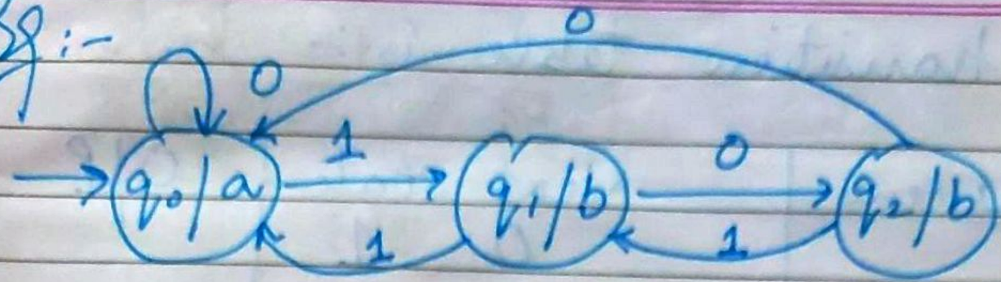
$q_0 \rightarrow$ Start state

$\Delta \rightarrow$ O/P symbol

$\lambda \rightarrow$ O/P function

$\lambda : Q \rightarrow \Delta$

Eg:-



$Q = q_0 \; q_1 \; q_2$

$\Sigma = 0, 1$

$\delta =$

	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_0	q_1

$q_0 = q_0$

$\Delta = (a, b)$

$\lambda =$

$q_0 \to a$
$q_1 \to b$
$q_2 \to b$

★ If we give I/P of 'n' length then O/P length = 'n+1'

Eg:- if a string is passed like
00110 then O/P would be :-
aaabaa
↓
This is O/P of initial state & rest depends like for 0 at q_0 we get a O/P & so on..
So 5 length I/P is giving 6 length O/P.

Final transition table is :-

Current	Next state		O/P
	0	1	
q_0	q_0	q_1	a
q_1	q_2	q_0	b
q_2	q_0	q_1	b

Mealy Machine :- FA with O/P.

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

$Q = q_0\ q_1$

$\Sigma = (a, b)$

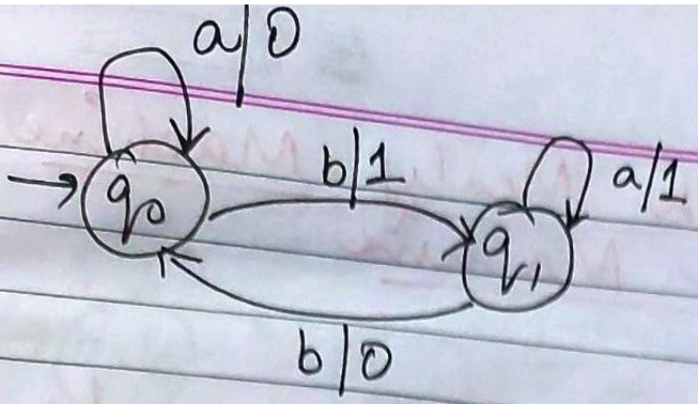
$\delta =$

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0

$q_0 = q_0$

$\Delta = (0, 1)$

$$\boxed{\lambda = Q \times \Sigma \rightarrow \Delta}$$



$\lambda =$

	a	b
q_0	0	1
q_1	1	0

★ If we give I/P of length 'n' then O/P length will also be 'n'!

Eg:- if a string is given like abba so acc. to it O/P would be:-

abba

Chk a O/P at q_0

$\overset{q_0}{a}$ $\overset{q_0}{b}$ $\overset{q_1}{b}$ $\overset{q_0}{a}$

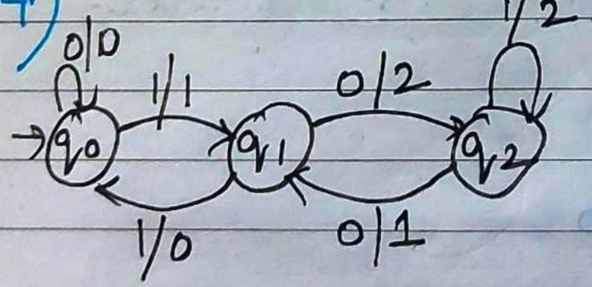
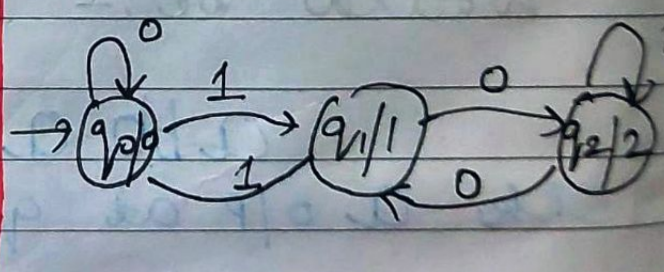
↓ ↓ ↓ ↓

0 1 0 0

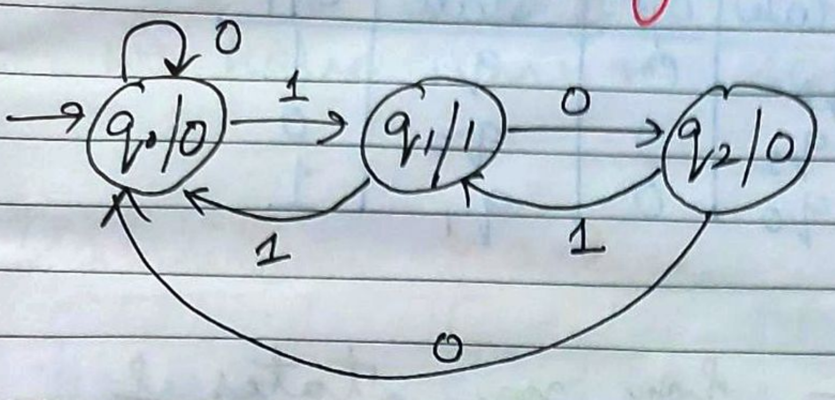
Final transition table is:-

Current	Next State			
	I/P = a		I/P = b	
	State	O/P	State	O/P
q_0	q_0	0	q_1	1
q_1	q_1	1	q_0	0

Difference b/w Mealy Machine & Moore Machine

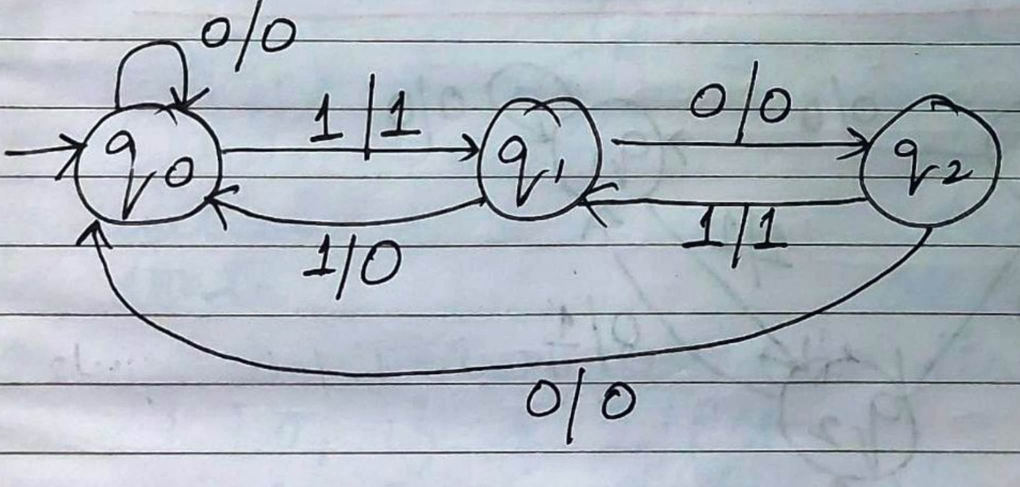
Mealy m/c	Moore m/c
1) O/P depends on present state & present I/P.	O/P depends on present state ~~only~~ I/P only.
2) I/P string of length 'n' → O/P length is also 'n'.	I/P string length 'n' → O/P string 'n+1'
3) $\lambda : Q \times \Sigma \rightarrow \Delta$	$\lambda : Q \rightarrow \Delta$
4) 	
5.) Its O/P is asynchronous	O/P is synchronous with clock.

Moore to Mealy Conversion :-



Moore

States	Next State 0	Next State 1	O/P
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_0	q_1	0



Mealy

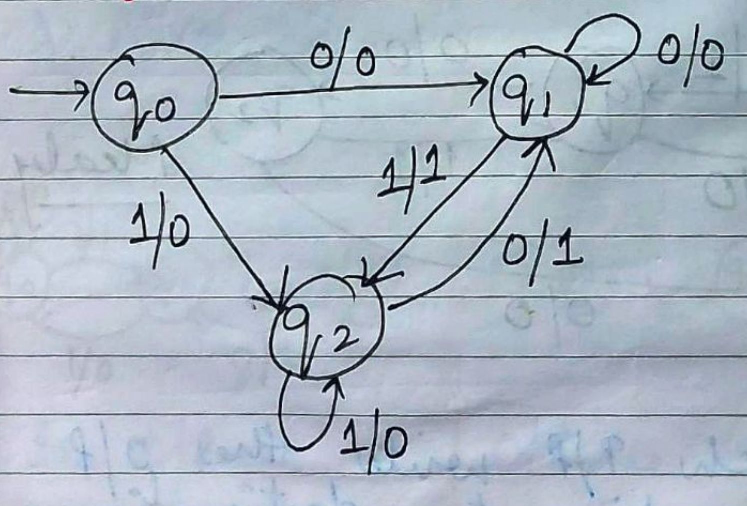
(For each I/P write the O/P corresponding to destination state. Eg:- q_0 is taking 1 to q_1 & q_1 has O/P as 1 so we write $\underbrace{1}_{I/P} / \underbrace{1}_{O/P}$.)

DATE ______
PAGE ______

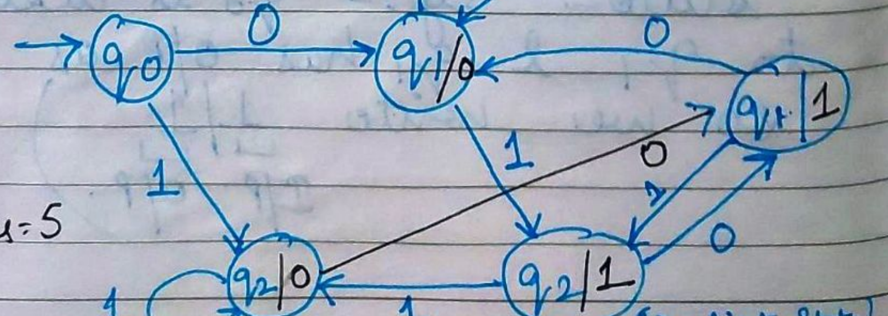
current	Next State 0		Next State 1	
	State	O/P	State	O/P
q_0	q_0	0	q_1	1
q_1	q_2	0	q_0	0
q_2	q_0	0	q_1	1

In Moore :- for m states & n O/P's. we have m states in mealy m/c.

Mealy to Moore Machine :-



⇒



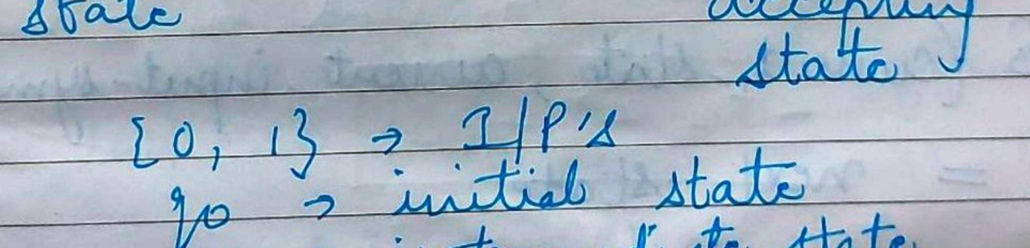
Now states = 5

In Mealy :- for m state, n O/P
we have $m \times n$ states

Transition Graph :-

Also called transition diagram.

It is a directed graph associated with the vertices of graph corresponds to the state of finite automata.



initial state

final or accepting state

$\{0, 1\} \rightarrow$ I/P's
$q_0 \rightarrow$ initial state
$q_1 \rightarrow$ intermediate state
$q_2 \rightarrow$ final state.

Transition Table :- is basically a tabular representation of transition function that takes two arguments (a state & a symbol) and returns a value called next state.

⇒ Rows corresponds to states
⇒ Columns " " I/P symbols
⇒ Enteries " " next states
⇒ Start state is marked with an arrow
⇒ Accept state is marked with * or circle.

$$\boxed{\delta : Q \times \Sigma \to Q}$$

Transition Function :- returns a state which is called next state.

δ (current_state, current_input-symbol) = next state.

Eg → $Q \times \Sigma \to Q$

$$\delta(q_0, a) = q_1$$

DATE
PAGE

Alphabets :- defined as finite set of symbols.
↳ Represented by Σ.

eg:- $\Sigma = \{0, 1, 2, 3, 4\}$
$\Sigma = \{0, 1\}$
$\Sigma = \{A, B - - - Z\}$
$\Sigma = \{a, b - - - z\}$

Strings :- finite set of symbols selected for some alphabet.

Eg:- if $\Sigma \{a, b\}$ is an alphabet then abab is the string over alphabet Σ.
⇒ It is denoted by w.

⇒ length of string is denoted as $|w|$

Eg:- $w = 01101$ has length
$|w| = 5$

⇒ An empty string can be represented by $\in$, ϵ or λ.

Language :- set of strings chosen from Σ^* where Σ is a particular alphabet is called a language.

DATE

PAGE

If Σ is an alphabet and $L \subseteq \Sigma^*$, then L is said to be language over alphabet Σ.

Kleene closure :-

is denoted by Σ^*

$\Sigma^* = \{$ set of all words over $\Sigma \}$

$= \underbrace{\Sigma^0}_{\text{empty}} \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots\ldots$

eg:- if $S = \{a\}$ then $S^* = \{\varepsilon, a, aa, aaa, aaaa, \ldots\ldots\}$

Positive Closure :-

denoted by Σ^+

$$\boxed{\Sigma^+ = \Sigma^* - \{\epsilon\}}$$

$\Sigma^+ = \{$set of all words over S excluding empty string $\varepsilon\}$

eg:- if $\Sigma = \{a\}$ then

$\Sigma^+ = \{a, aa, aaa, aaaa \ldots\}$