THEORY OF COMPUTATION

DATE
PAGE

Moore Machine :

An automata in which the output depends only on the states of the machine, is called as the moore machine.

Mealy Machine :

Automata in which output depends on the state & input both, is called mealy machine.

finite State Machine:

A model of computation consisting of a set of

states, a start set, and input alphabet & a transition fn that maps input symbols & current state to a next state.
Computation begins in the start state with an input string. It changes to new state depending on the transition function.

⇒ A ~~FSM~~ finite state automata is a model of behaviour composed of finite no. of states, transition b/w these states and actions which implies that FSM is an abstract model of a machine with a primitive internal memory.

DFA : Deterministic Finite Automata

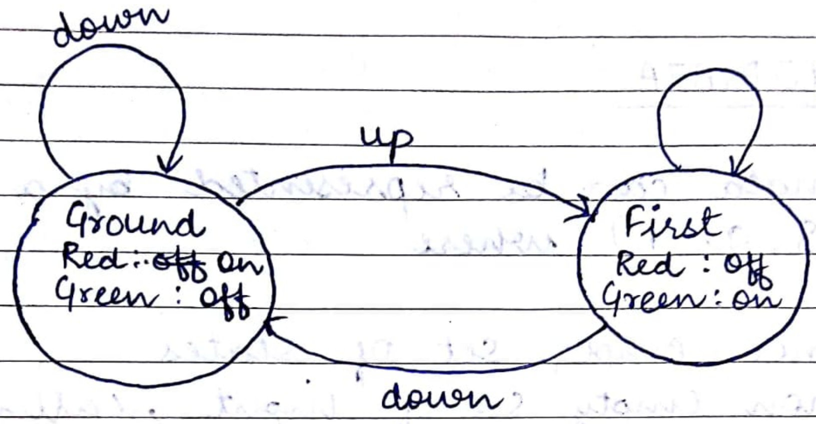
A FSM with atmost one transition for each symbol & state.

NFA : Non-deterministic finite Automata

A FSM whose transition function maps input symbols & states to a set of next states. The transition fn also may map the null symbol & states to next states.

Illustrn: To design a FSM ⇒

The elevator can be at one of the two floors. There is one button that contains the control mech. of the elevator (Top/Down). Also there are 2 heights in the elevator denoting the current floor - Red (ground), Green (first). At each step, controller checks the current floor & input, changing the floor & light correspondingly.



~~Green: 0~~ Ground : 0 , Down : 0 , off : 0
~~Red:~~ first : 1 , up : 1 , on : 1

current	Input	Next	Red	Green
0	0	0	1	0
0	1	1	0	1
1	0	0	1	0
1	1	1	0	1

FSM limitations:

→ usable for simple instructions
→ graphical represent^n is cumbersome
→ Translation to logic is prone to error
→ It doesn't have the capacity to remember large amount of info, so there are only fixed no. of states
→ If the initial state & the input sequence is

Ullmann 2nd
CJ Martin 3rd

known, then we can find the current state of the machine, but its reverse is not true.
→ It cannot recognise the centre.
→ Can't ensure well-formedness (brackets).

FINITE AUTOMATA :

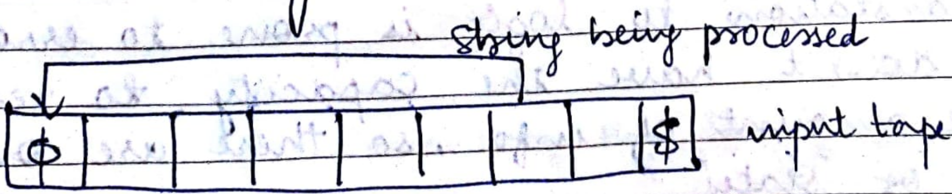
finite automata can be represented by a 5 tuple set $(Q, \Sigma, \delta, q_0, F)$ where

Q : finite non-empty set of states
Σ : finite non-empty set of input, called Input Alphabet
δ : function which maps $Q \times \Sigma \rightarrow Q$, called as the direct transition fn

this mapping usually represented by transition graph / table

q_0 : $\in Q$, is the initial state
F : subset of Q, is a set of ~~finite~~ final states
↓
$\subseteq Q$

It is assumed here that there may be more than one final state



DATE
PAGE

Transition system / graph is a finite directed labelled graph, in which the nodes represent the state & the directed edges indicate the transition of the state, and the edges are labelled as input / output.

Properties :

1) $\delta(q, \Lambda) = q$
This means the state can only be changed by a input symbol

2) $\delta(q, aw) = \delta(\delta(q,a), w)$
$\delta(q, wa) = \delta(\delta(q,w), a)$

for all strings w and input symbol a :
This property gives the state after the automata consumes / reads the first symbol of the string aw & the state after the automata consumes the prefix of the string wa

Acceptability of a String by a finite automata:

A string x is accepted by a finite automata
$M = (Q, \Sigma, \delta, q_0, F)$
if $\delta(q_0, x) = q$ for some $q \in F$

A final state is also called as accepting state.

Non-Deterministic finite automata:

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

The transition function δ does mapping from

$$Q \times \Sigma \rightarrow \underbrace{2^Q}_{\text{Power Set}}$$

Illustration :

Consider a FSM whose transition function δ is given as :

State	Input 0	Input 1
$\rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

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

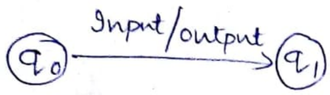
$\Sigma = \{0, 1\}$

$F = \{q_0\}$

$$\underbrace{\delta(q_0, 110101)}_{\text{To find}}$$

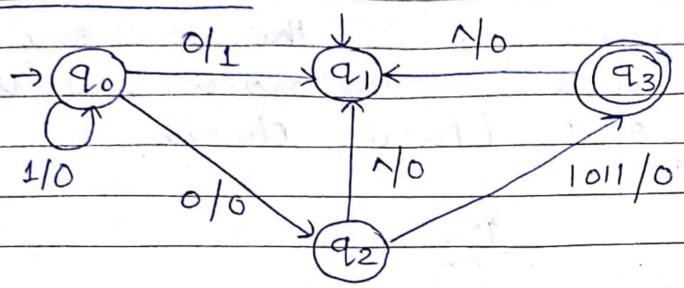
$$\begin{aligned}
\delta(q_0, 110101) &= \delta(q_1, 10101) \\
&= \delta(q_0, 0101) \\
&= \delta(q_2, 101) \\
&= \delta(q_3, 01) \\
&= \delta(q_1, 1) \\
&= \delta(q_0, \Lambda) \\
&= q_0 \in F
\end{aligned}$$

$\Rightarrow$ This string is accepted by this automata.



DATE
PAGE

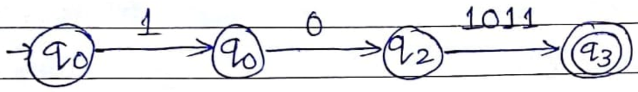
Illustration:



$q_0 = \{q_0, q_1\}$
$F = \{q_3\}$

Determine the acceptability of the string:

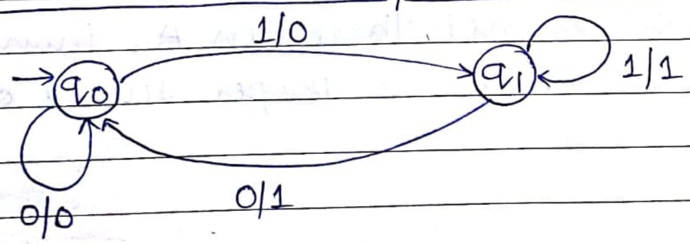
1) 101011



2) 111010

Since we are not able to reach the final state from any of the paths, this string isn't accepted.

ONE-MOMENT DELAY MACHINE:

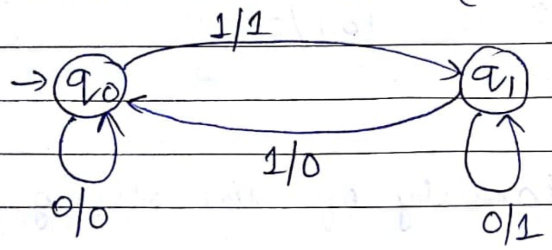


The first output is 0 & afterwards, the output is the symbol read at previous instance.

Input: 011010011
Output: 0011010011

→ Two moment delay me initially two zeroes.

Ques: Construct a FSM which reads a binary string and outputs a '0' if the no. of ones it has read is even & 1 if the no. of ones it has read is odd. (Parity checker)

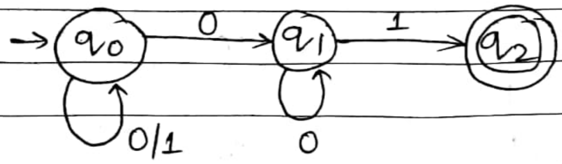


SETS :

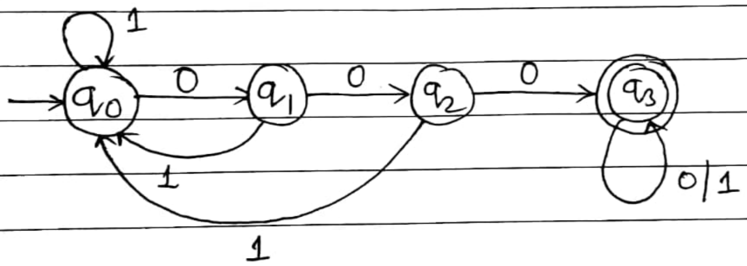
1) If every member of a set A is a member of set B, then $A \subseteq B$
2) ~~If~~ set A & set B are equal if they have same members ie $A = B$ if $A \subseteq B$ & $B \subseteq A$
3) If $A \subseteq B$ & $A \neq B$ ie if every member of A is in B and there is atleast one member of B which is not there in A, then we write $A \subset B$, ie A is a proper subset of B.

Tutorial :

1 Construct an NFA to accept all strings terminating in 01

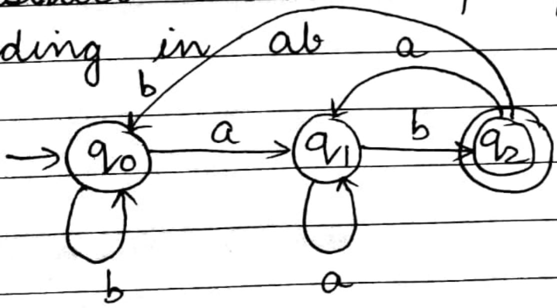


2 Construct an NFA to accept those strings containing three consecutive zeroes



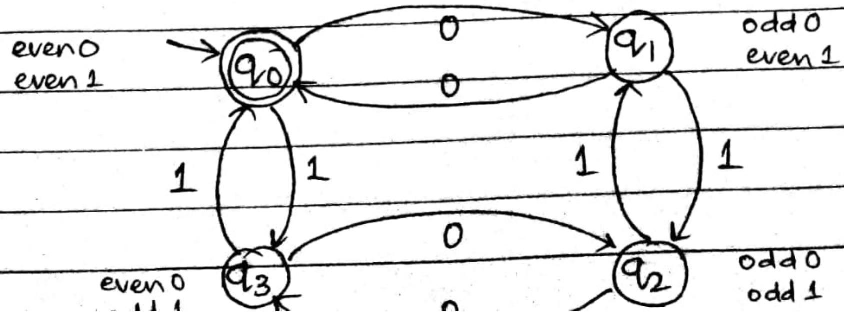
Doubt

3 Construct a DFA accepting all strings over {a,b} ending in ab



Doubt

4 Construct a DFA to accept all strings containing even no. of '0' & even no. of '1'.



Curr State		Input	Next-state
(q_0)	00	0	01
(q_1)	01	0	00
(q_2)	10	0	11
(q_3)	11	0	10
	00	1	11
	01	1	10
	10	1	01
	11	1	00

DATE
PAGE

State \ Σ	a	b
q_0		

DATE

PAGE

Ques: Construct a DFA equivalent to

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

State Table (δ)

State	Input(0)	Input(1)
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0, q_1

(o) The states are subsets of Q
ϕ, $[q_0]$, $[q_0, q_1]$, $[q_1]$

(o) Initial state : $[q_0]$

(o) final state : $[q_0]$, $[q_0, q_1]$

(o) δ :

state / Σ	0	1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0 q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

Ques: Construct a DFA equivalent to

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

δ

State / Σ	a	b
$\rightarrow q_0$	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	ϕ	q_2

(o) All states are subsets of Q.

(o) Initial state : $[q_0]$

(o) final state : All the subsets in δ, having q_3.

ϕ :

State \ Σ	a	b
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$

FINITE AUTOMATA WITH OUTPUTS:

The mathematical model of FSM is defined by:
$M = \{Q, \Sigma, \delta, q_0, F\}$
which doesn't include the information about output.
But now we will be considering FSM with outputs:

1) Moore machine
2) Mealy machine

Both of them are not working as finite automata.
There are no final states present in the machine.
($\Rightarrow$ These can't be used as language acceptors)
i.e. These machines provide some output only.

The FA which we have considered till now have binary outputs i.e they accept or don't accept the string.
This acceptability was decided on the basis of reachability of the final state from initial state.

Now, we remove this restriction, consider the model where outputs can be chosen from some other alphabets.
The value of output fn $Z(t)$ in the most general case is a function of present state q & the present input $x(t)$ i.e.

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

where λ is the o/p function

This generalised model is usually called as the **MEALY MACHINE**.

DATE
PAGE

If the output fn $Z(t)$ depends only on the present state & is independent of the current function, then it is called as the MOORE MACHINE

$$Z(t) = \lambda(q(t))$$

Moore machine

A moore m/c is a 6-tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

where Δ : output alphabet
λ : Mapping from $Q \to \Delta$ giving the output associated with the state

Mealy machine

A Mealy m/c is a 6-tuple set $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

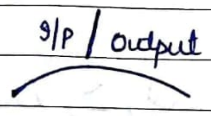
where λ: output fn mapping $Q \times \Sigma \to \Delta$

NOTE:

- In Mealy m/c, we get an output only on the applcn of an i/p ~~sto~~ symbol.
  So, for input string Λ, the output is Λ
- A finite automata can be converted into a Moore machine by introducing $\Delta = \{0, 1\}$ and defining $\lambda(q) = \begin{cases} 1, & \text{if } q \in F \\ 0, & \text{if } q \notin F \end{cases}$
- For Moore machine, if the i/p string is of length n, the o/p string is of length n+1. The first

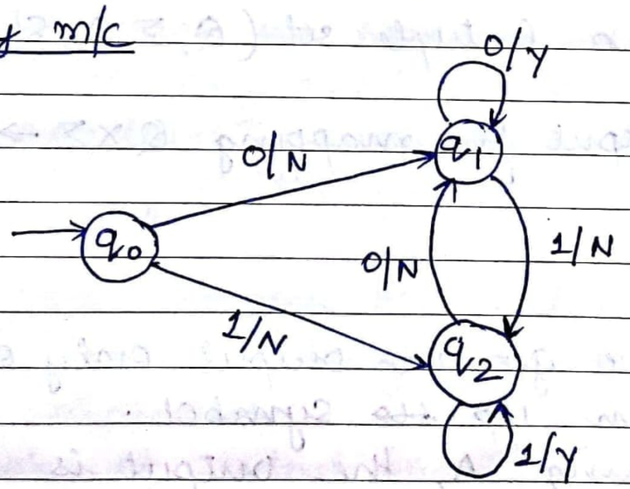
output is $\lambda(q_0)$ for all output strings.
In case of Mealy m/c, if i/p string is of length n the output string is also of the same length 'n'.

Mealy m/c	Moore m/c
i/p / output	q_0/o/p
with input	with state

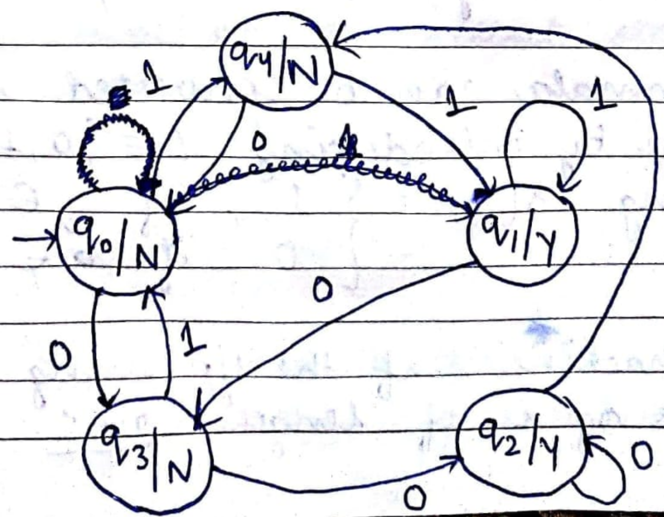


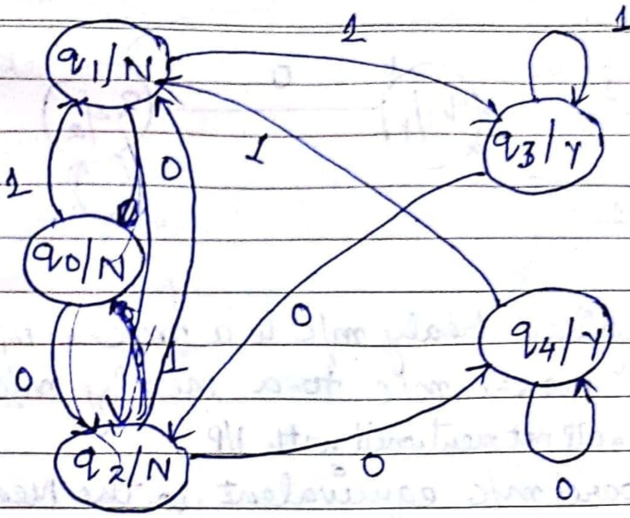
Ques: Construct a Mealy & Moore machine that produces y if input ends with 00 or 11 and N otherwise.

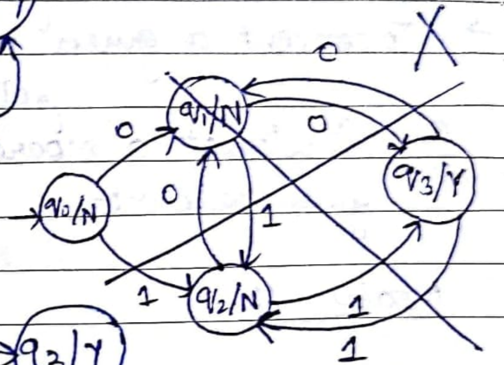
Mealy m/c



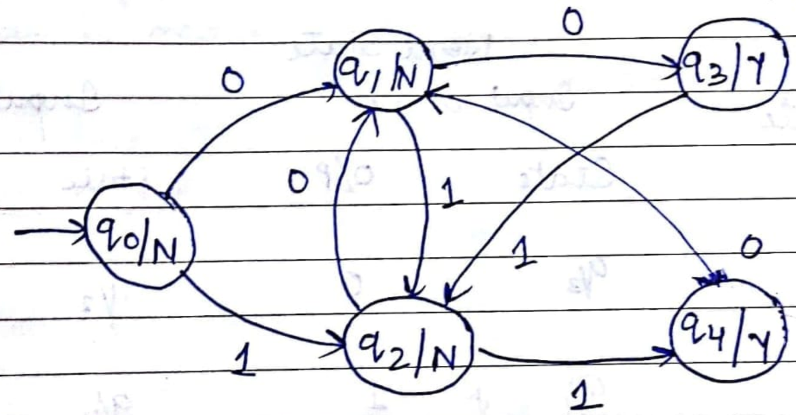
Moore m/c







How about this?



Q. Residue % 3 of binary numbers using mealy & moore machine

$i \rightarrow 0 \longrightarrow 2i$

$i \rightarrow 1 \longrightarrow 2i+1$

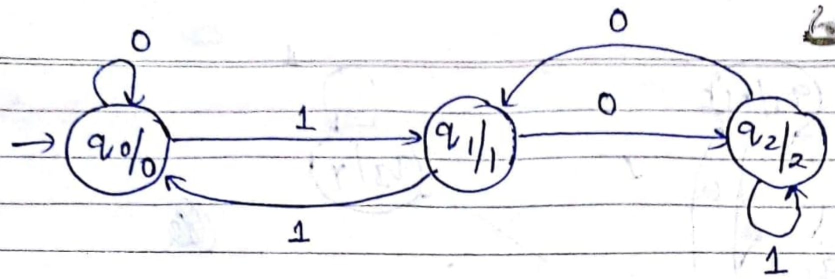
$i/3 \Rightarrow p = 0, 1, 2$ ($p \rightarrow i \% 3$)

$0 \rightarrow 2i/3 \rightarrow 2p \bmod 3$

$1 \rightarrow 2i+1/3 \rightarrow 2p+1 \bmod 3$

$p=0$:	0	0	$p=1$:	0	2	$p=2$:	0	1
	1	1		1	0		1	2

DATE
PAGE



→ Procedure to convert a given Mealy m/c to a moore m/c
→ To convert a given moore m/c to a mealy m/c

o/p not mentioned with I/P

Q. Construct a moore m/c equivalent to the Mealy machine given below:-

Mealy m/c

Present state	Next State Input a = 0 State	O/P	Input a = 1 State	O/P
→ q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

At the first stage we develop the procedure so that both machines accept exactly the same set of I/P sequences. We look into the next state coloumn for any state say q_i and determine the no. of different o/p assosciate with q_i in that col.

We split q_i into several diff states, the no. of such states being equal to the no. of diff o/p assosciated with q_i

Scanned by CamScanner

	a=0		a=1	
Present state	State	O/P	State	O/P
$\rightarrow q_1$	q_3	0	q_{20}	0
q_{20}	q_1	1	q_{40}	0
q_{21}	q_1	1	q_{40}	0
q_3	q_{21}	1	q_1	1
q_{40}	q_{41}	1	q_3	0
q_{41}	q_{41}	1	q_3	0

~~Now we~~ So, The pair of states and the outputs in the next col can be rearranged as

Present state	Next state a=0	a=1	O/P
$\rightarrow q_1$	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

Q Let $M_1 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ be a Moore m/c Then following procedure may be executed to construct eq mealy m/c M2.

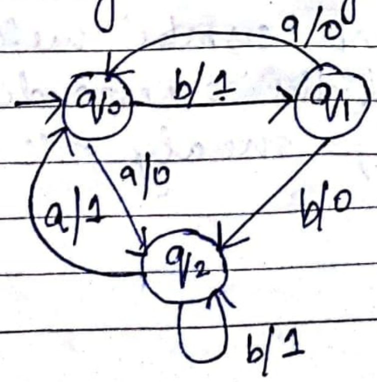
In the case of moore m/c for every input symbol, we form the pair consisting of the next state & the corresponding o/p for and reconstruct the table for mealy m/c

Present state	Next state a = 0	Next state a = 1	O/P
→ q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

Present state	Next state a = 0		Next state a = 1	
	State	O/P	State	O/P
→ q_0	q_3	0	q_1	~~0~~ 1
q_1	q_1	1	q_2	~~1~~ 0
q_2	q_2	0	q_3	0
q_3	q_3	0	q_0	0

If 2 states have identical transitions (rows corresponding to these 2 states are identical) then we can delete one of them

Q Convert given Mealy M/C to Moore m/C



P.S	N.S I/P ⇒ a		I/P = b	
	State	O/P	State	O/P
→ q_0	q_2	0	q_1	1
q_1	q_0	0	q_2	0
q_2	q_0	1	q_2	1

P.S	a		b	
	state	o/p	state	o/p
q_{00}	q_{20}	0	q_1	1
q_{01}	q_{20}	0	q_1	1
q_1	q_{00}	0	q_{20}	0
q_{20}	q_{01}	1	q_{21}	1
q_{21}	q_{01}	1	q_{21}	1

P.S	N.S I/P = a	I/P = b	O/P
→ q_{00}	q_{20}	q_1	0
q_{01}	q_{20}	q_1	1
q_1	q_{00}	q_{20}	1
q_{20}	q_{01}	q_{21}	0
q_{21}	q_{01}	q_{21}	1

Q Construct a Mealy m/c which is eq to the moore m/c given in table:-

P.S	N.S a=0	N.S a=1	O/P
$\rightarrow q_0$	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

P.S	N.S a = 0 State	a = 0 O/P	a = 1 State	a = 1 O/P
$\rightarrow q_0$	q_1	$\cancel{1}$ 0	q_2	1
q_1	q_3	$\cancel{0}$ 1	q_2	$\cancel{0}$ 1
q_2	q_2	1	q_1	$\cancel{1}$ 0
q_3	q_0	1	q_3	1

What is Dead state?

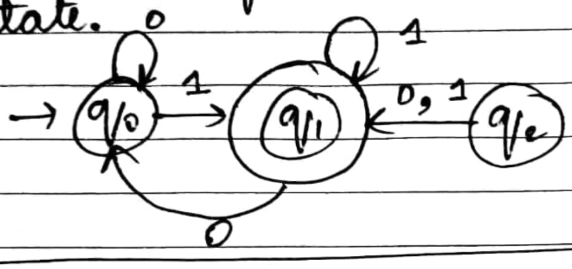
All those nonfinal states which transit to itself for all input symbols in ~~as~~ Σ are called as dead state

If $\Sigma = \{0, 1\}$ q_4 is said to be a dead state if q is not member of F and.

$\delta(q, 0) = q$

$\delta(q, 1) = q$

Inaccessible State : All those states which can never be reached from initial state are called inaccessible state.



q_2 is inaccessible