Theory of Computation CSE302

TOC is a branch of theoretical computer Science that deals with the study of algorithms & computational complexity.

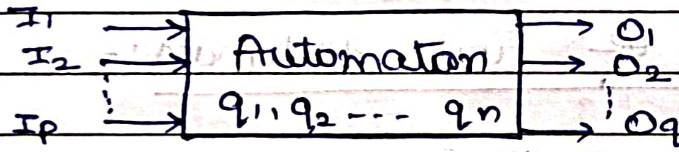
It aims to answer Fundamental questions about:-

(i) what can be computed.

(ii) how efficiently it can be done

(iii) what limitations exist in terms of computational power

Automaton :- It is defined as a system where energy, ~~or~~ material and information are transferred or transformed or transmitted & is used for performing some Functions without the direct participation of man.



Characteristics of an Automata :-

- Input :- At discrete instance of time the I/P values $I_1, I_2 --- I_p$, each of which can take a finite number of fixed values from the Input alphabet sigma Σ
- Output :- $O_1, O_2 --- O_q$ are the outputs of model, each of which can take infinite no. of fixed values from an o/p 'O'.
- States :- At any instant of time, the automaton can be in one of the states $q_1, q_2, --- q_n$
- States relation :- Next state of an automaton at any instant of time is determined by present state & present I/P.

- O/P relation :- O/P is related to either state only or to both input and state

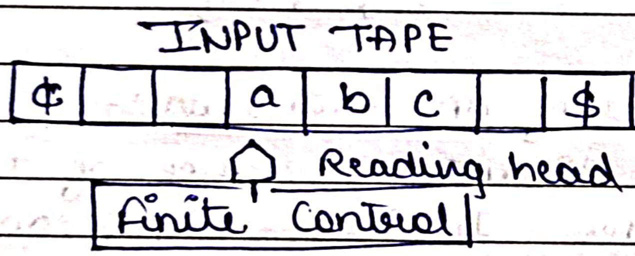
NOTE :- (i) An automaton in which the O/P depends only on O/P is called Automata without a memory

(ii) An automaton in which the O/P depends on State is called automaton with finite memory.

(iii) An automaton in which the O/P depends only on states of machine is called moore machine.

(iv) An automaton in which the O/P depends on State & inputs ~~is called~~ at any instant of time is called mealy machine.

Block diagram of Finite Automata :-



(i) ¢ and $:- These are called end markers. Absence of end markers means tape is of infinite length. Each Square containing a single symbol from the I/P alphabets Sigma Σ & represent I/P string to be processed.

(ii) Reading head :- It examines only one square at a time and can move from left to right or right to left.

(iii) Finite control :- The I/P of finite control will be usually the symbol under the reading head. It is a set of possible states that the machine can be in.

Finite state machine :-

- SYMBOL :- 0, 1, 2, a, b, c (alphabet or no)
- ALPHABET :- (Σ) It is a collection of symbols
  $\{a, b\}$, $\{1, 0, 1\}$, $\{a, b, c\}$
- STRING :- It is a sequence of symbols. (a, b, ab, a
  00, 01, 11
- LANGUAGE :- It is a set of strings

$\Sigma(0,1) \rightarrow$

L_1 = Set of all strings of length 2.

0,0, 01, 10, 11

Power of sigma :-

$\{\epsilon\}$ Σ^0 :- It is set of all string of length zero
empty set

Σ^1 :- It is set of all string of length one
$\hookrightarrow \{0, 1\}$

Σ^2 :- $\{00, 01, 10, 11\}$

Cardinality :- No. of elements in a set is called cardinality

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup ----$$

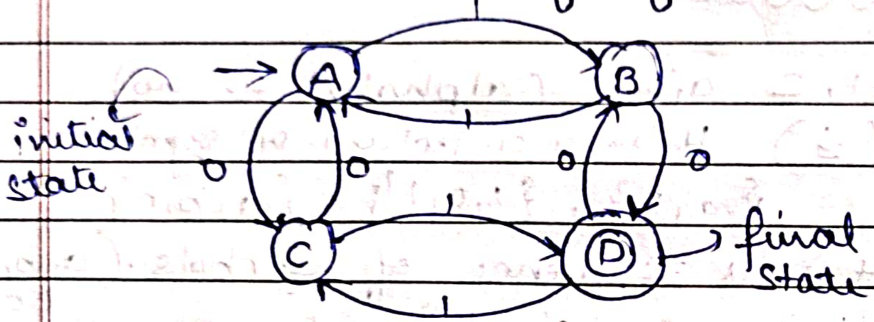
$$\Sigma^* = \{\epsilon\} \cup \{0, 1\} \cup \{00, 01, 10, 11\} \cup ----$$

Defination of finite Automaton :-

- Q = It is a finite non-empty set of states
- Σ = It is a finite non-empty set of I/P. called input alphabet
- δ = It is a Function which maps $Q \times \Sigma \times Q$ which is usually called as direct transaction function. This is the Function

which describe the change of state during transition. This mapping is usually represented by transition table or transition diagram

- q_0 :- q_0 belongs to Q is the initial state
- F :- It is the subset of Q which represent the set of final states



Q = A, B, C, D
$\Sigma = \{0, 1\}$
$q_0 = A$
F = D

δ =

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

Finite Automata :- It is the simplest form of model of computation. It has a very limited memory.

Finite Automata
- FA with output
  - Moore Machine
  - Mealy Machine
- FA without output
  - DFA
  - NFA
  - E-NFA

- **Determintics Finite Automata (DFA)**:-
A finite Automata is said to be determintics if for every state there is a unique I/P

symbol which takes state to required next unique state. This means that given a state S_j, the same I/P symbol doesnot cause finite automata to move into more than one state. There is always a unique next state for all I/P symbol.

- Non-Deterministic Finite Automata (NFA) :-

In NFA, it is possible to have more than one transition on reading same I/P symbol from a given state. Such a machine is not probabilistic as no weights are assigned to different possible transitions from states for same symbol, this is known as probabilistic machine. And NFA is denoted by 5 tuple notation. A string x is said to be accepted by finite automata if

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

(ii) $\delta(q_0, x) = Q$

As we know, δ is a state transition function that maps $Q \times \Sigma$ to 2^Q thus each individual can be denoted by $\delta(q_0, a_1) = q_1$

where q_0 is the current state, a_1 is the current I/P symbol and q_1 is the Next state

→ Acceptance of language :-

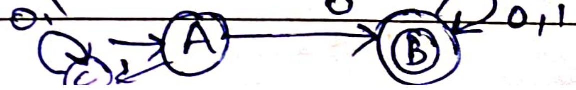
A language is a set of string over some alphabet. If there is a language 'L' such that

$$L = \{x \mid \delta(q_0, x) = Q \text{ for some } Q = F\}$$

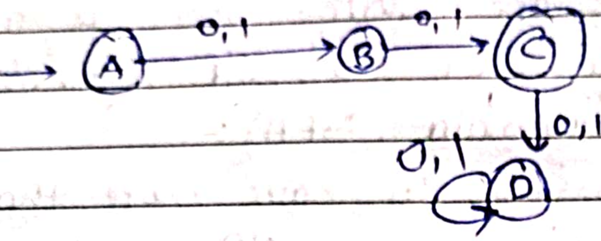
then it is said to be accepted by finite automata 'M' and is denoted by L(M). In such case the language is accepted by automata (M) and is also called as regular set or regular language.

Ex:- of DFA

L_1 = set of all strings that start with 0.

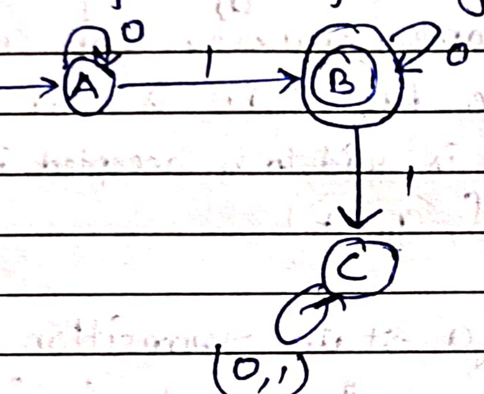


Ex 2:- Construct a DFA that accept set of all strings over 0,1 of length 2.
possible = (00, 01, 10, 11)



Lyods Clan

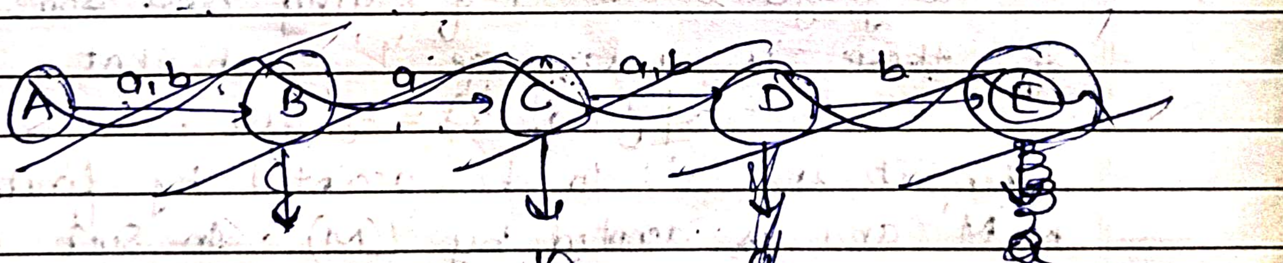
3. Design a finite automata which accepts strings containing exactly single one over alphabet (0,1).

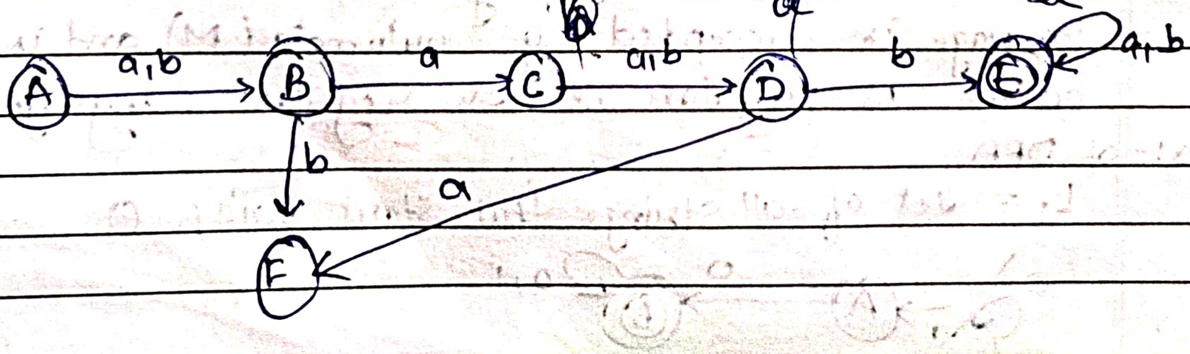


01000
00100
only 1.

4. Design a finite automata which accepts the language $L = \{ w \in (a,b)^* \}$, Second symbol of w is ~~zero~~ a and fourth is b.

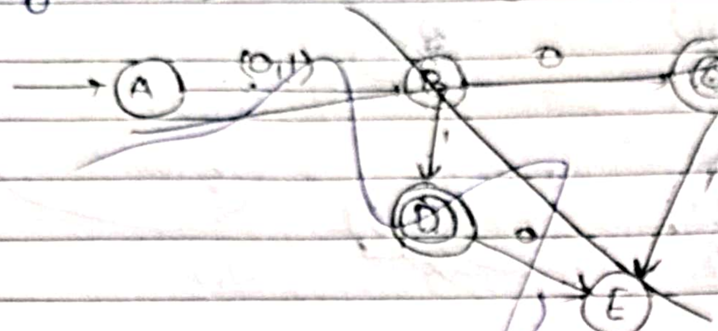
$\underset{(a,b)}{\_}\ \underset{}{\overset{a}{\_}}\ \underset{(a,b)}{\_}\ \underset{}{\overset{b}{\_}}$ length = 4. aaab



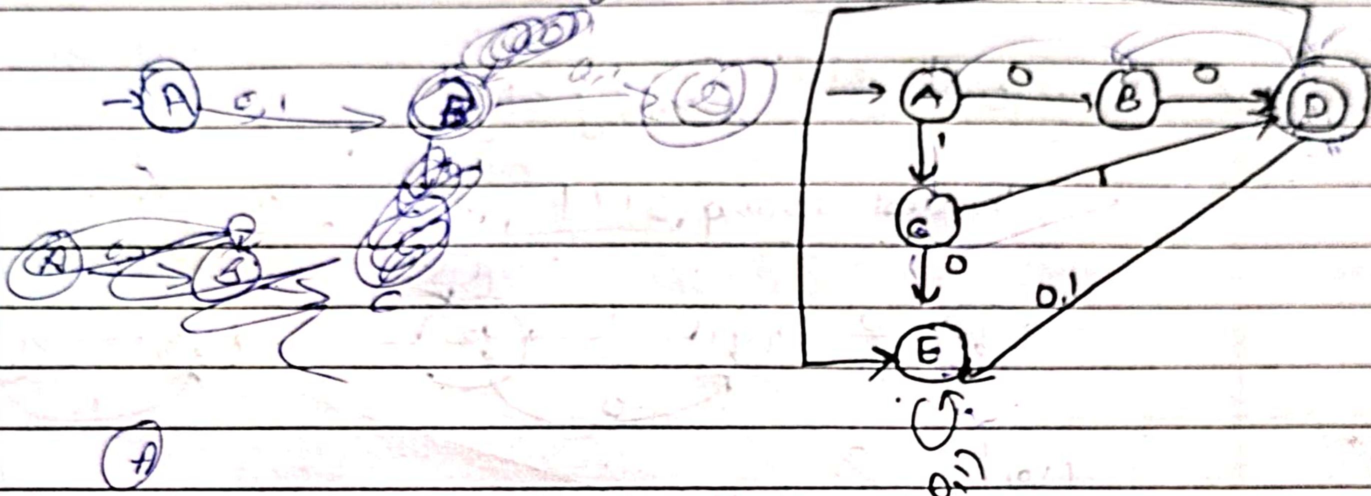


length ②

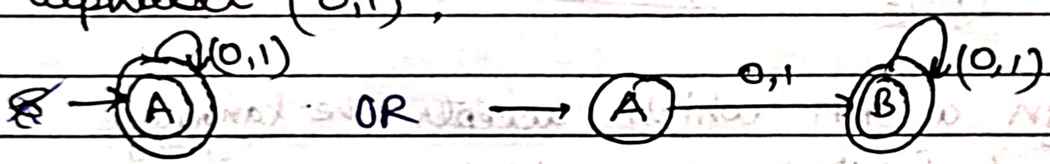
5. Design a DFA that accepts either 00 or 11



A = 0	→ B
A = 1	→ B.
B → 0	→ C
B = 1	→ D



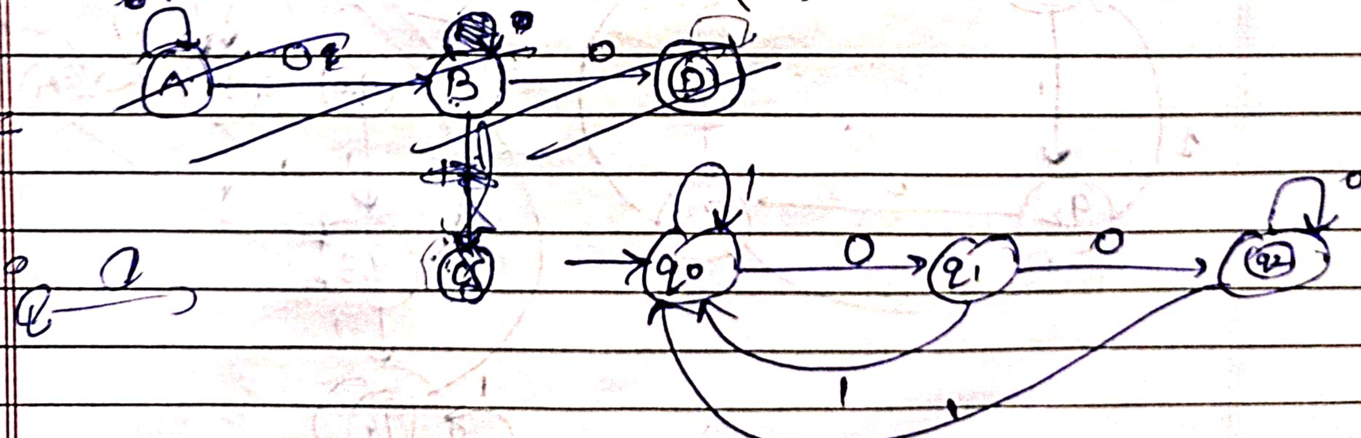
6. Design a DFA which accepts all strings over the alphabet (0,1).



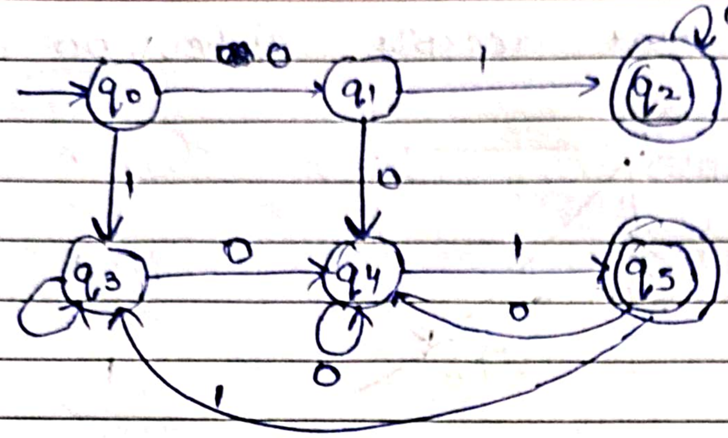
7. Design a DFA which accepts strings that ends with 00 over the alphabet (0,1)

8. Design a DFA which accepts set string which over aphabet (0,1) either start with 01 or end with (01)

7.



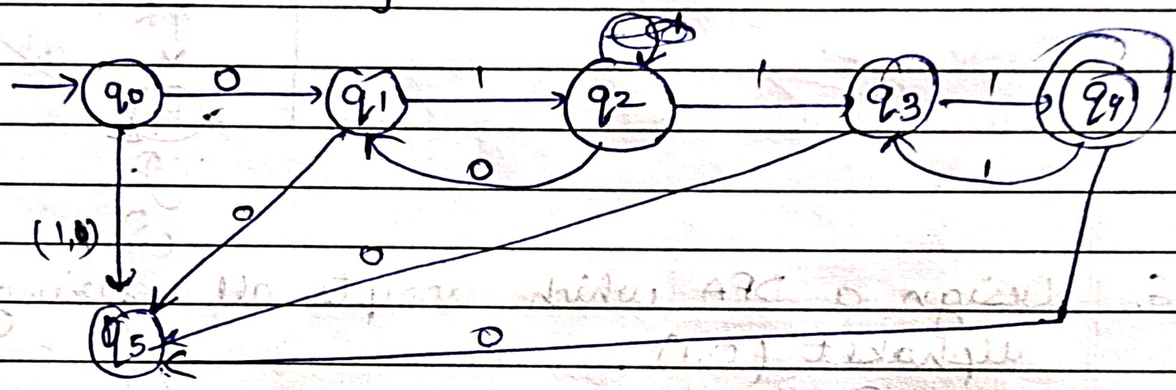
8.



1001011
010010

9. $L = \{(01)^i 1^{2j},\ i \geq 1, j \geq 1\}$.

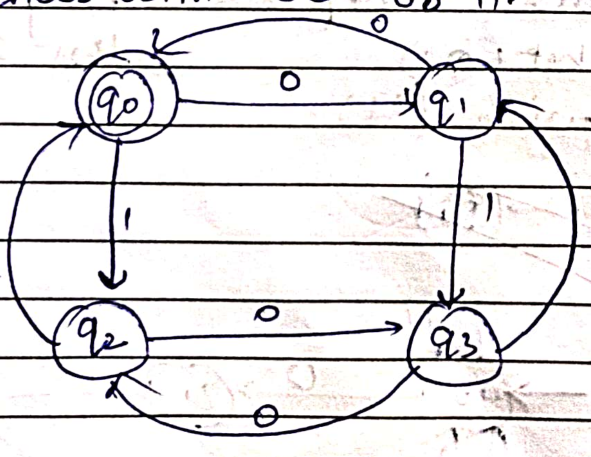
Shortest string $\underbrace{01}\underbrace{11}$, $\underbrace{0101}\underbrace{1111}$



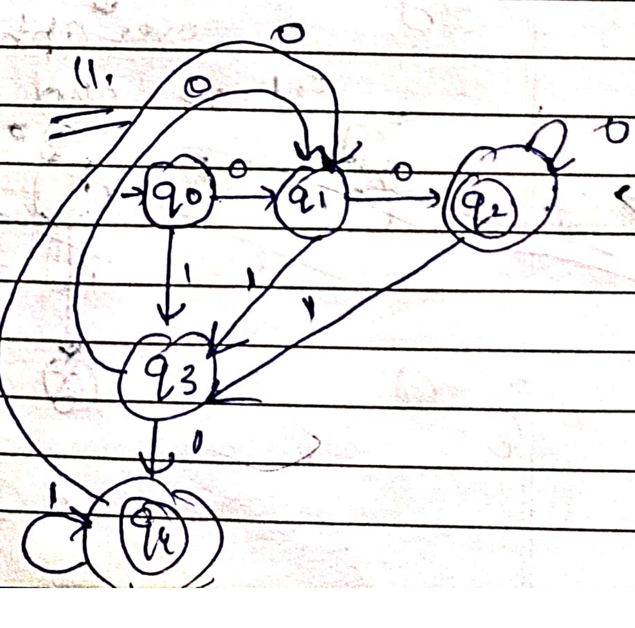
10. Design a DFA which accepts the language $L = \{w \mid w$ should have both, an even no. of zeros and an even no. of one's.

11. Design a DFA which accepts a string that either ends with 00 or 11.

10.



11.



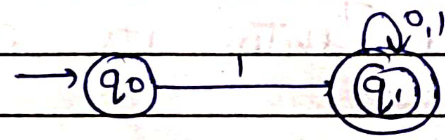
Properties of DFA:-

1. In DFA, given the current state we know what the next state will be
2. It has only one unique next state
3. It has no choices or randomness.
4. It is simple & easy to design.
5.

Properties of NFA:-

1. In NFA, given the current state there could be multiple next states.
2. The next state may be choosen at Random
3. All the next states may be choosen in parallel.

Ex:-

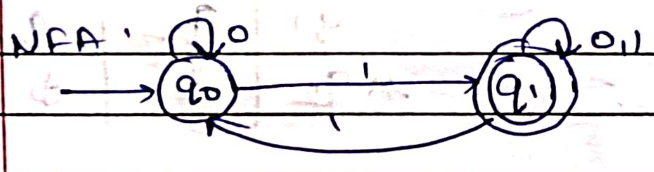


ε\δ	0	1
q0	q0	q1
*q1	q1	(q0,q1)
q0,q1	q0,q1	q0,q1

NFA to DFA:-

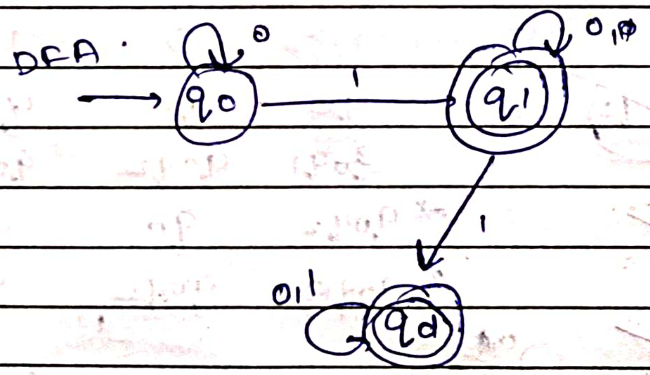
NFA:

1.

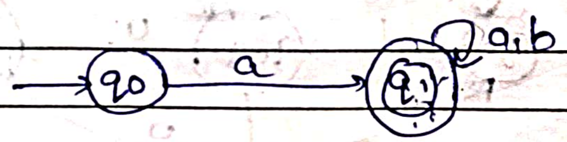


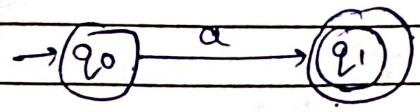
—Answer

DFA:



2.

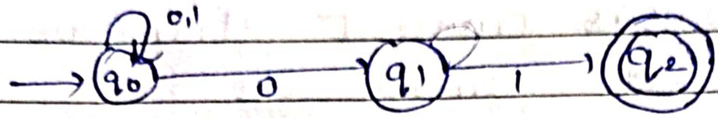




ε\δ	a	b
q0	q1	-
q1	q1	q1

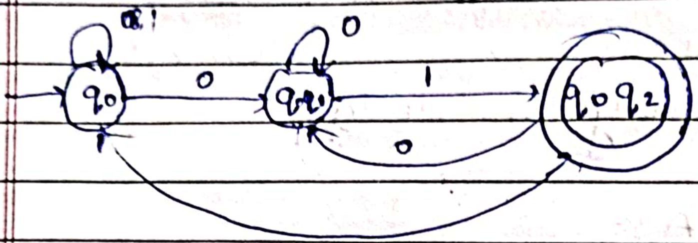
1 Q:- Design a NFA which accepts a set of string over 0,1 that ends with (0,1)

NFA



	0	1
q_0	$q_0 q_1$	q_0
q_1	ϕ	q_2
* q_2	ϕ	ϕ

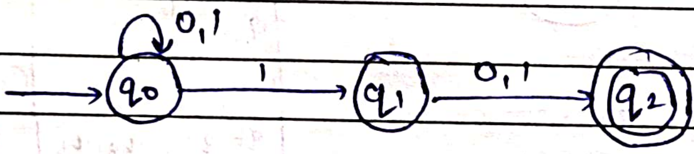
DFA



	0	1
q_0	$q_0 q_1$	q_0
$q_0 q_1$	$q_0 q_1$	$q_0 q_2$
* $q_0 q_2$	$q_0 q_1$	q_0

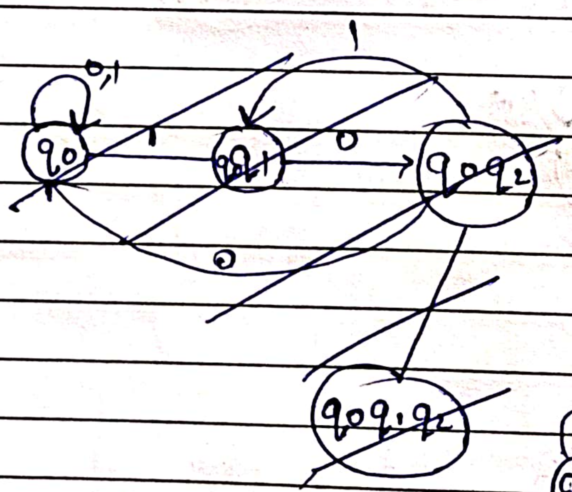
2 Q:- Design a NFA for the language that accepts all strings over (0,1) in which the Second ~~class~~ last symbol is always 1, Then convert it into its equivalent DFA.

NFA



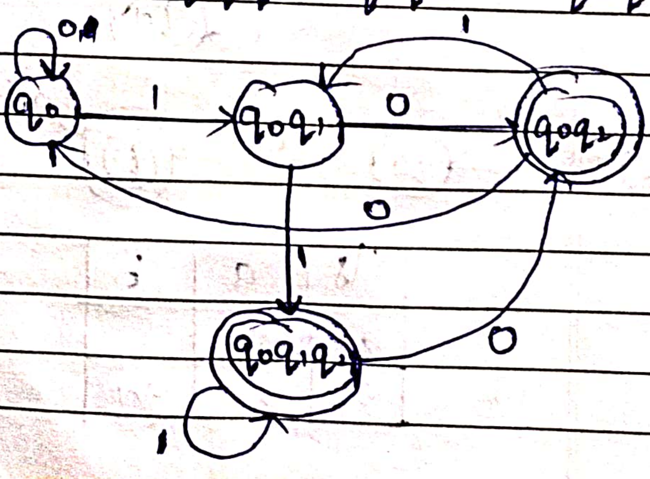
	0	1
q_0	q_0	$q_0 q_1$
q_1	q_2	q_2
* q_2	ϕ	ϕ

DFA



	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	$q_0 q_1$
* $q_0 q_1 q_2$	$q_0 q_2$	$q_0 q_1 q_2$

—!—

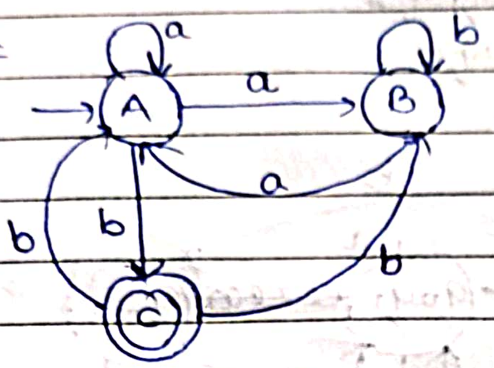


NFA

Q:3

	a	b
→ A	A,B	C
B	A	B
* C	–	A,B

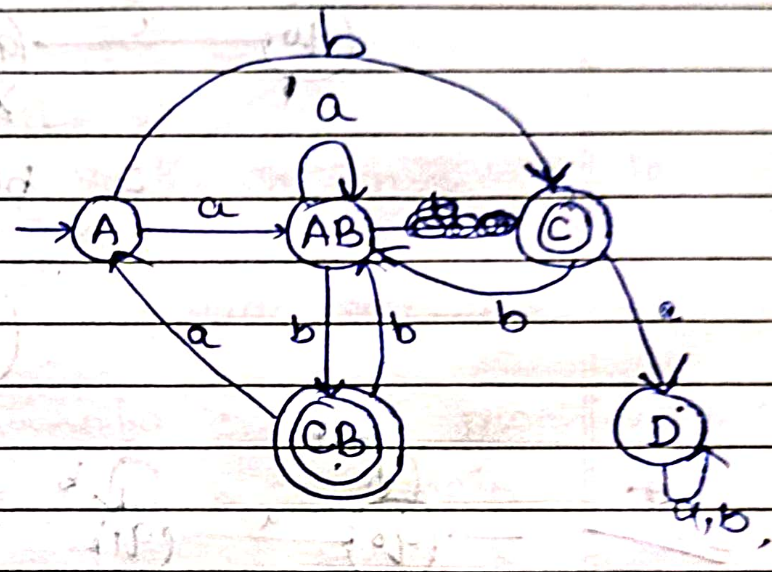
NFA =



DFA :–

	a	b
→ A	A,B	C
A,B	AB	CB
CB	A	AB
C	D	AB
D	D	D

dead state

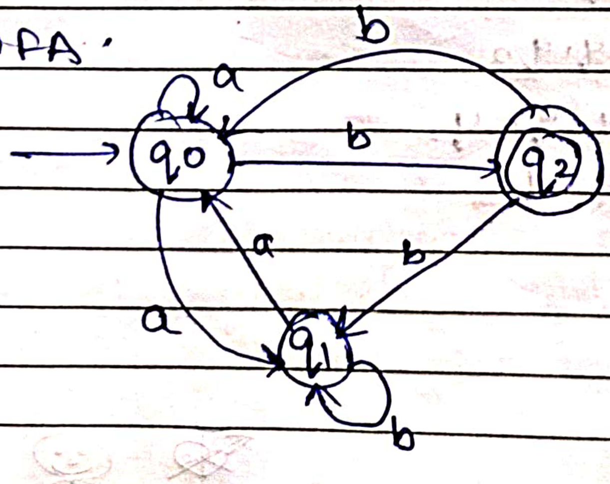


Q3:- Find the equivalent DFA for the NFA given by M = [{A,B,C}, {a,b}, δ, A, {C}] where δ is given by

transition function

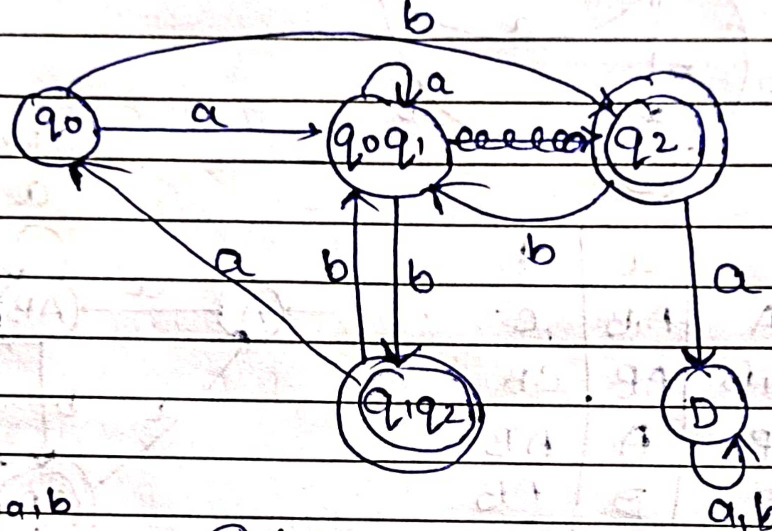
	a	b
→ A	AB	C
B	A	B
* C	–	AB.

Q4:- NFA.

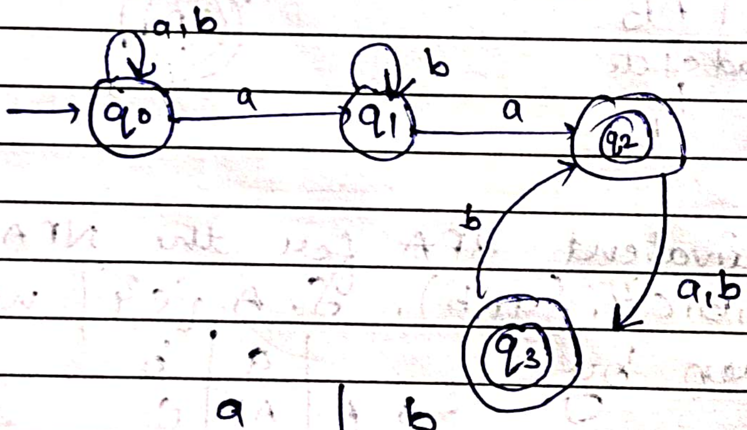


	a	b
q0	q0q1	q2
q1	q0	q1
* q2	ϕ	q0q1

	a	b
q_0	q_0q_1	q_2
q_0q_1	q_0q_1	q_1q_2
* q_0q_2	q_0	q_0q_1
~~$q_0q_1q_2$~~	~~q_0q_1~~	~~$q_0q_1q_2$~~
* q_2	D	q_0q_1
D	D	D

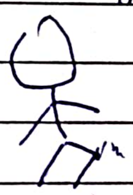


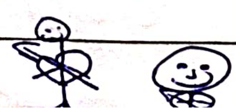
2.

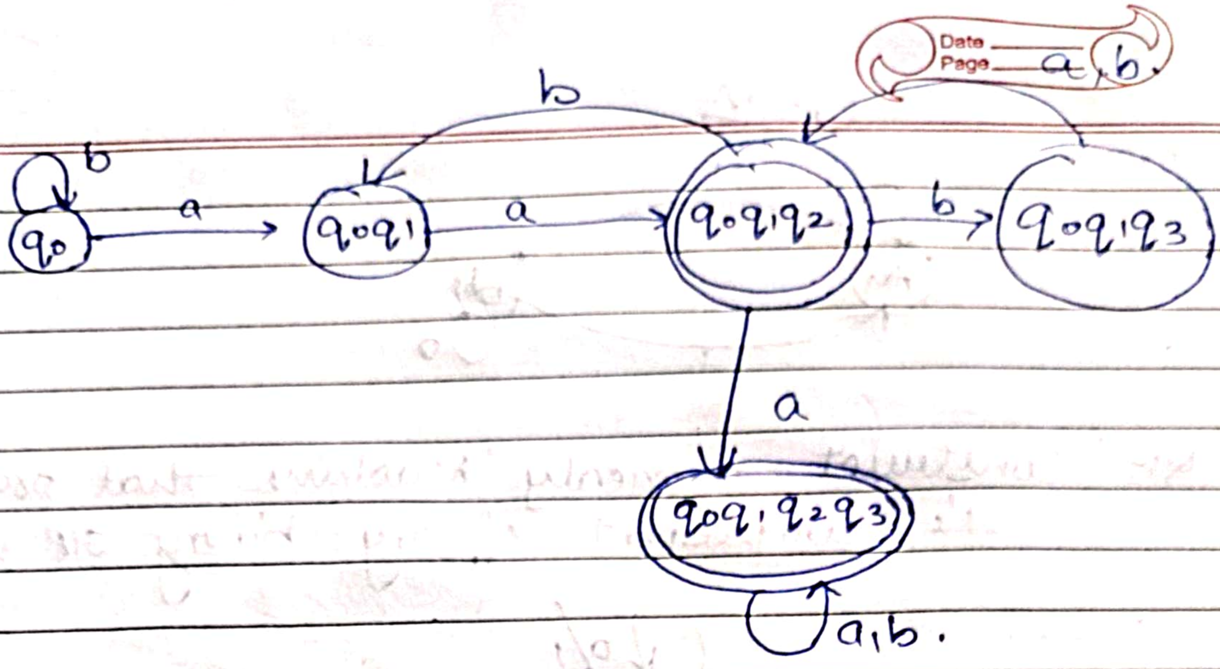


	a	b
q_0	q_0q_1	q_0
q_1	q_2	q_1
* q_2	q_3	q_3
q_3	ϕ	q_1

	a	b
q_0	q_0q_1	q_0
q_0q_1	$q_0q_1q_2$	q_0q_1
* $q_0q_1q_2$	$q_0q_1q_2q_3$	$q_0q_1q_3$
* $q_0q_1q_2q_3$	$q_0q_1q_2q_3$	$q_0q_1q_3q_2$
* $q_0q_1q_3$	$q_0q_1q_2$	$q_0q_1q_2$







finite automata with outputs

There is no final state in these.

1. Mealy Machine: It is a finite automata in which o/p is associated with each transition. In Mealy Machine every transition for a particular I/P symbol has a fixed o/p.

Mathematically, Mealy Machine is a six tuple machine and is defined as

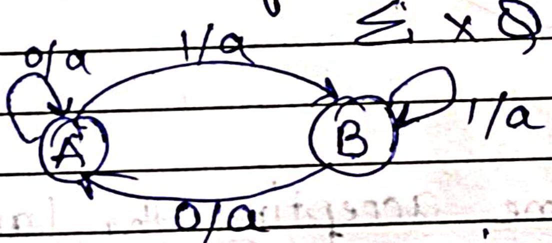
$$M_0 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

Q → States, Σ → I/P, Δ → o/p, δ → Transition, q_0 → Initial state

λ = o/p function

$$\Sigma \times Q \longrightarrow \Delta$$

Ex:-

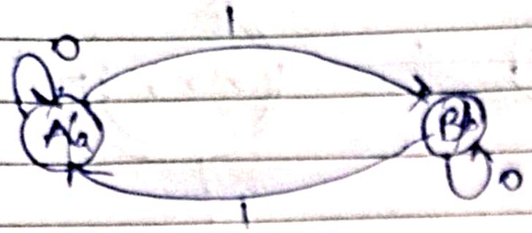


String :- 0110
aa ba

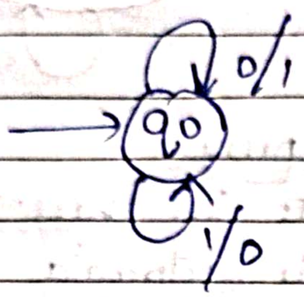
2. Moore Machine:- It is a finite automata in which the o/p is associated with each state. In moore Machine every state of this finite machine has a fixed o/p.

Mathematically, it is a six tuple machine & its is defined as $M_0 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$.

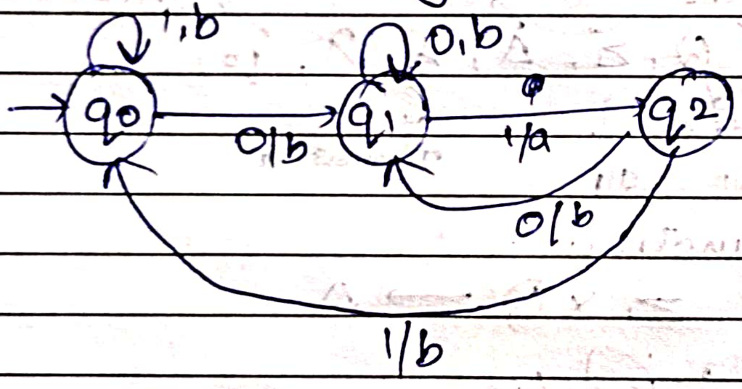
Q → Δ



Q:-1 Construct a mealy Machine that produces 1's complement of any binary I/P string.



Q2:- Construct a mealy machine that prints a whenever the sequence 01 is encountered in any I/P binary string.

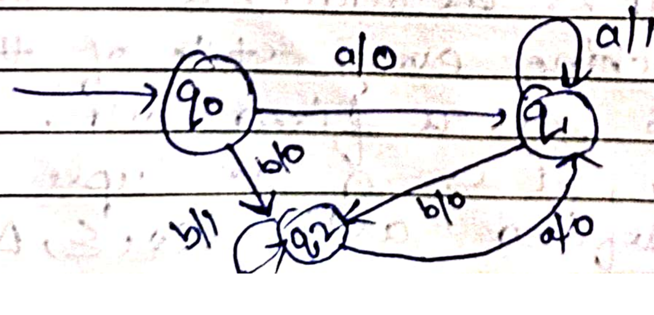


0 1 1 0 1
b a b b a

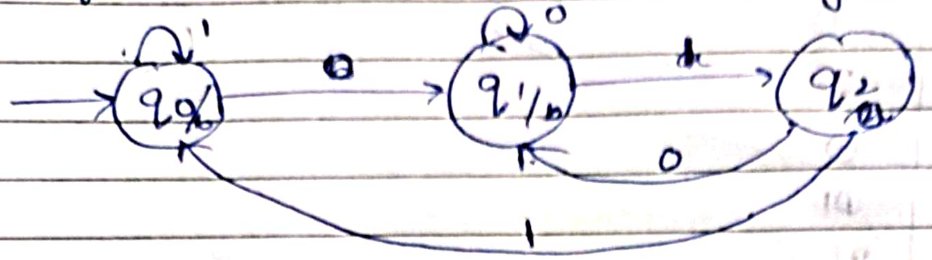
3. Design a mealy machine accepting the language consisting of strings from Σ^* where $\Sigma = \{a, b\}$ & string which end with either aa or bb.

Ans:

i/p: a b a a
o/p: 0 0 0 1



Q:- Construct a moore machine that prints a whenever the sequence 01 is encountered in any I/P binary string.

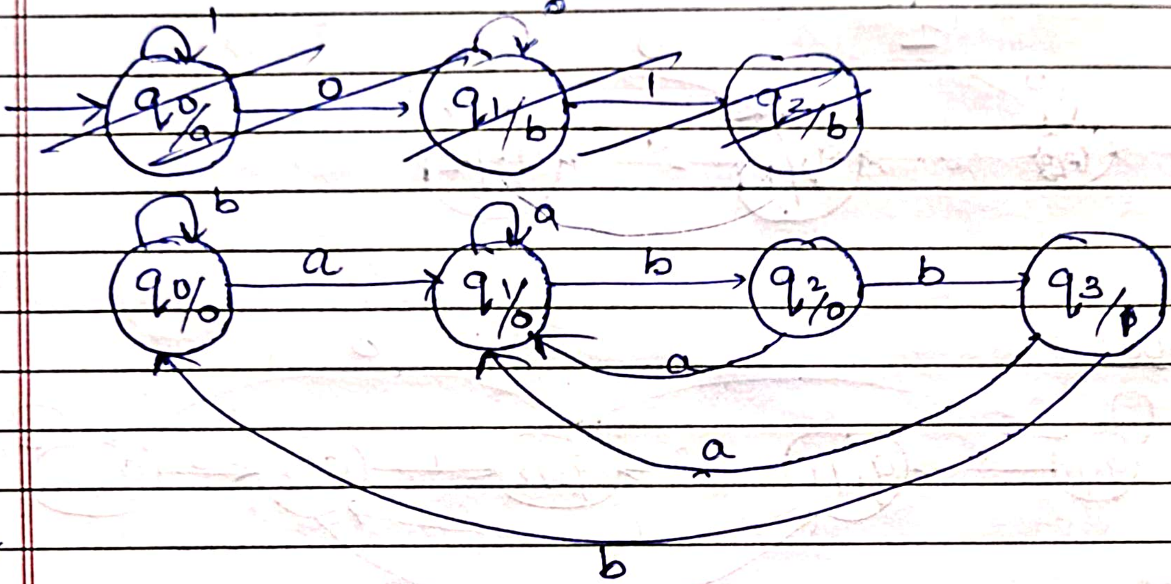


01101

babba

Q.- Construct a moore machine that counts the occurrence of the sequence abb in any I/P string over a, b.

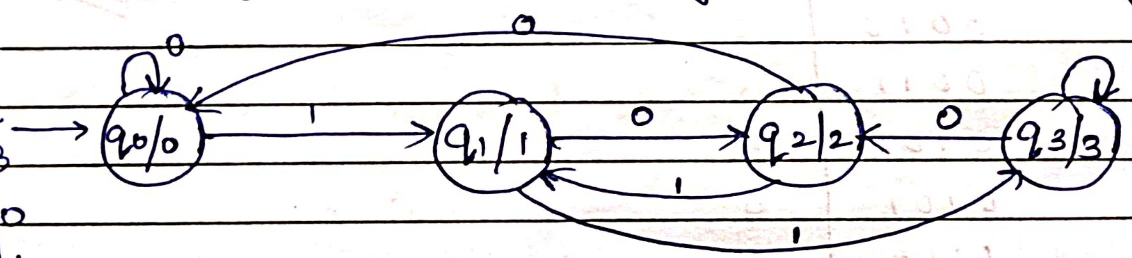
a,bb,
1



imp

Q:- Construct a moore machine which determine residue of mode 4 of each binary string treated as binary integer.

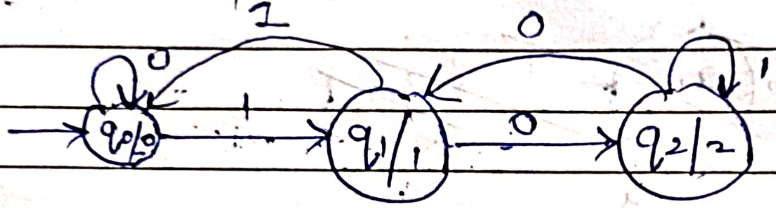
0000	0
0001	1
0010	2
0011	3
0100	0
0101	1
0110	2
0111	3
1000	0
1001	1



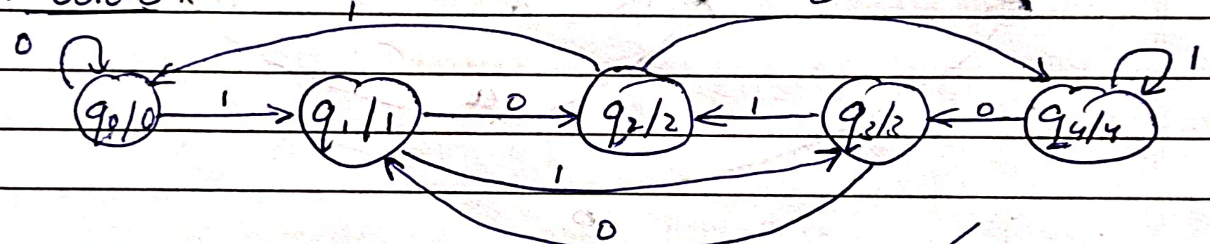
Q:- Contruct a moore machine which determine residue of mode 3 of each binary string treated as binary integer.

mode3.

0000	0
0001	1
0010	2
0011	0
0100	1
0101	2
0110	0
0111	1
1000	2



Mode 5:

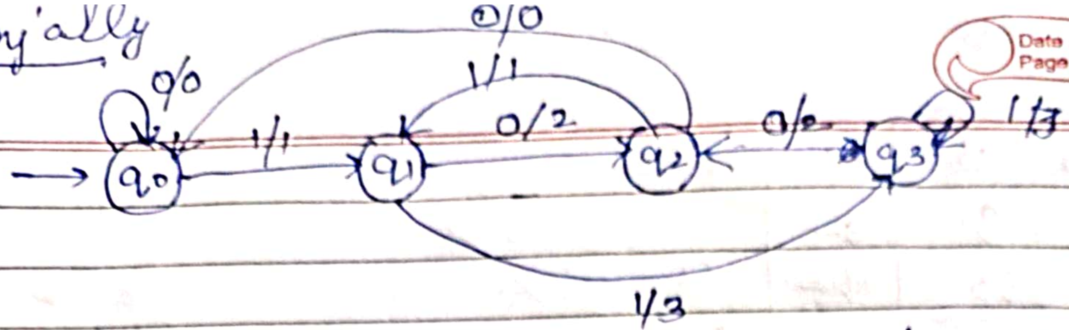


0000	0
0001	1
0010	2
0011	3
0100	4
0101	0
0110	1
0111	2
1000	3
1001	4

~~bad handwriting~~

Q:- Construct a mealy machine which calculates residue of mode4 for each binary string treated as binary integer.

Graphically

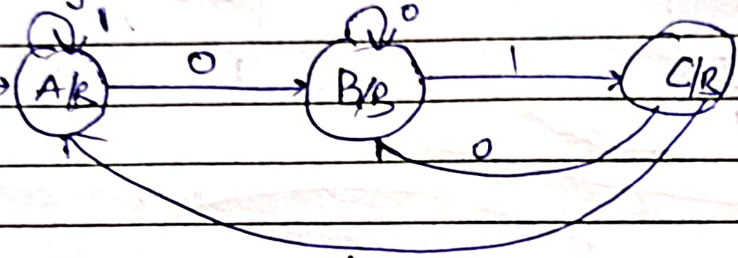


v good

Q:- Conversion of moore machine to mealy machine

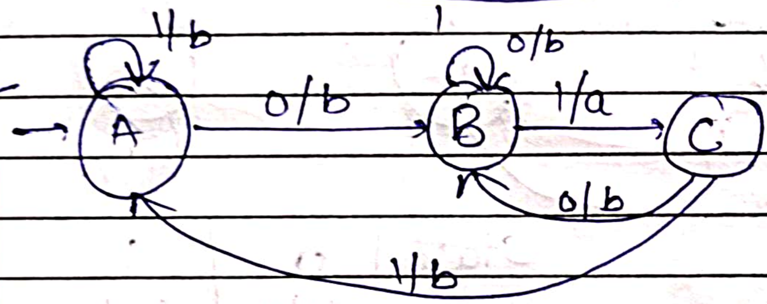
1. Construct a moore machine that print 'a' whenever seq. 01 is encountered at any I/P binary string and than convert it into its equivalent mealy machine

Moore



State	0	1	O/P
A	B	A	b
B	B	C	b
C	B	A	a

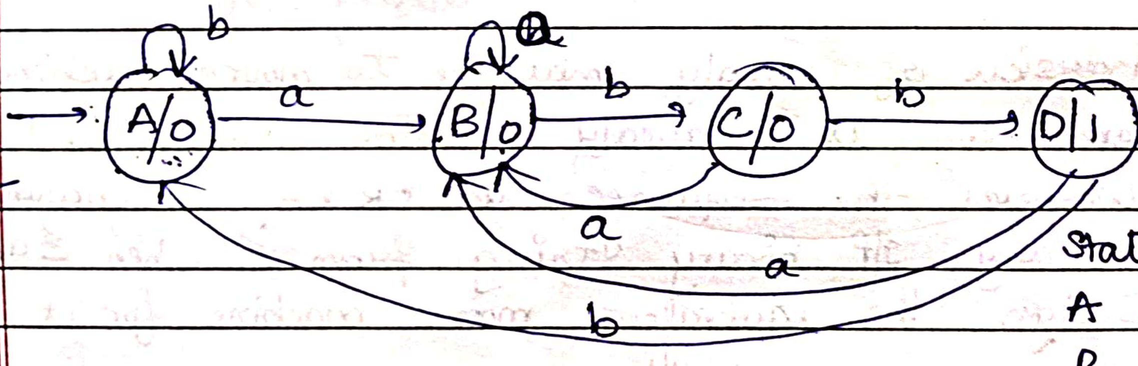
Mealy



State	0	1
A	B, b	A, b
B	B, b	C, a
C	B, b	A, b

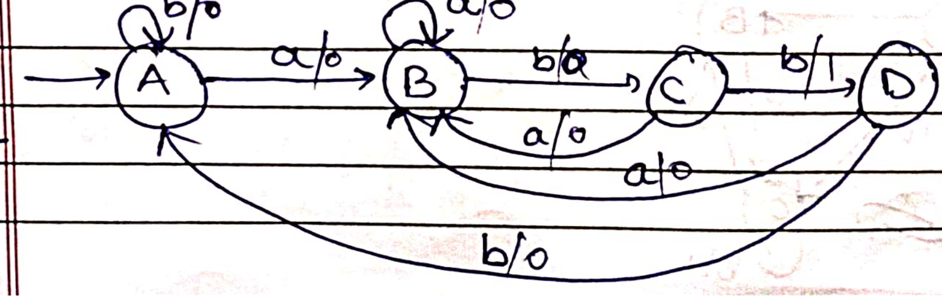
2. The given moore machine count the occurence of seq. abb in any I/P binary string over a, b. Convert to its mealy machine.

Moore



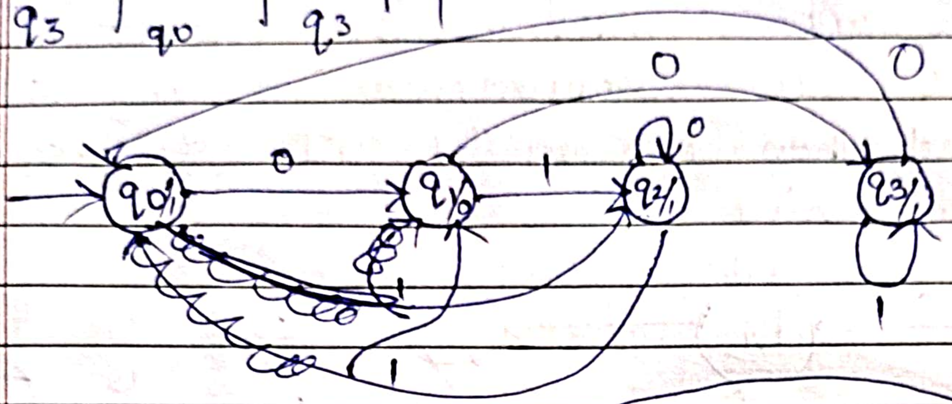
State	a	b	O/P
A	B	A	0
B	B	C	0
C	B	D	0
D	B	A	1

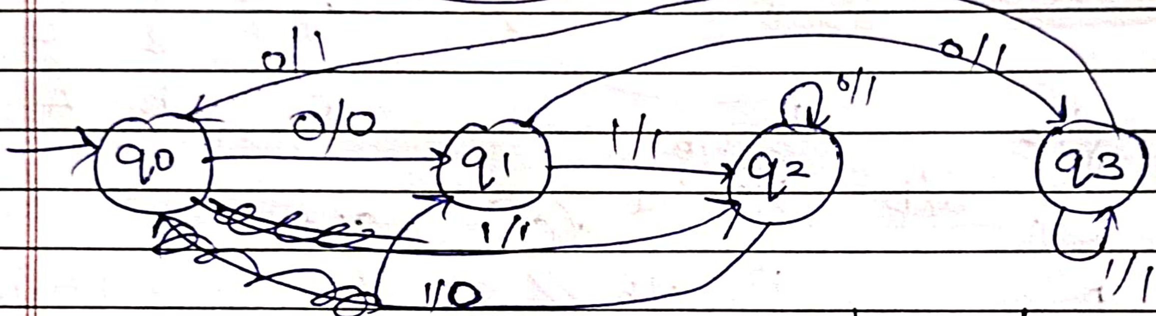
Mealy



A	B, 0	A, 0
B	B, 0	C, 0
C	B, 0	D, 1
D	B, 0	A, 0

State	0	1	O/P
→ q0	q1	q2	1
q1	q3	q2	0
q2	q2	q1	1
q3	q0	q3	1

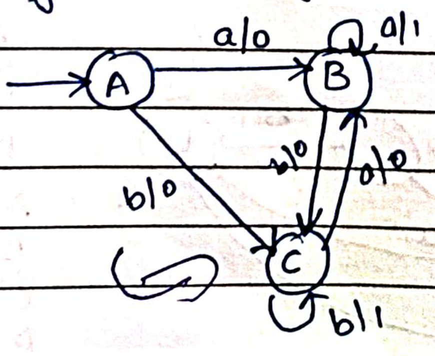


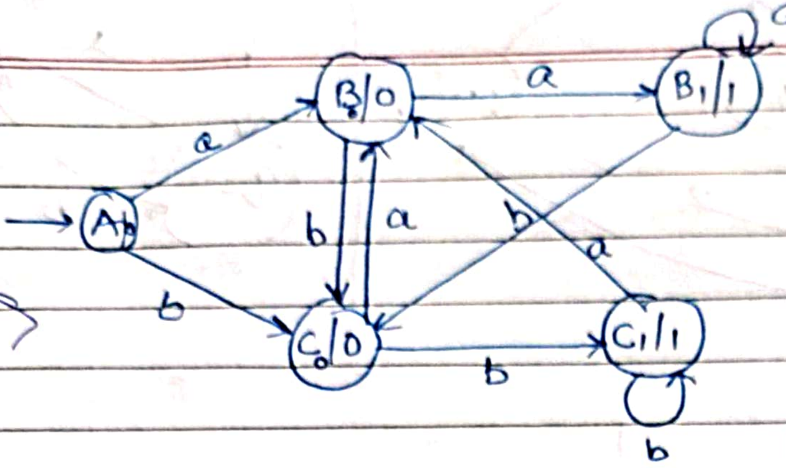


State	0	1
q0	q1/0	q2/1
q1	q3/1	q2/1
q2	q2/1	q1/0
q3	q0/1	q3/1

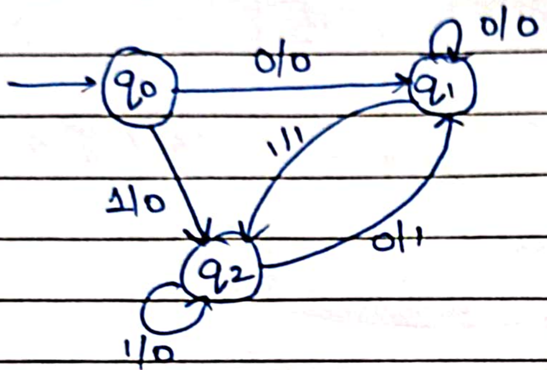
Q:- Conversion of mealy machine to moore machine

① Given below is a mealy machine that prints 1 whenever the sequence aa OR bb is encountered in any I/P binary string from Σ^* when Σ a, b. Design the equivalent moore machine for it.

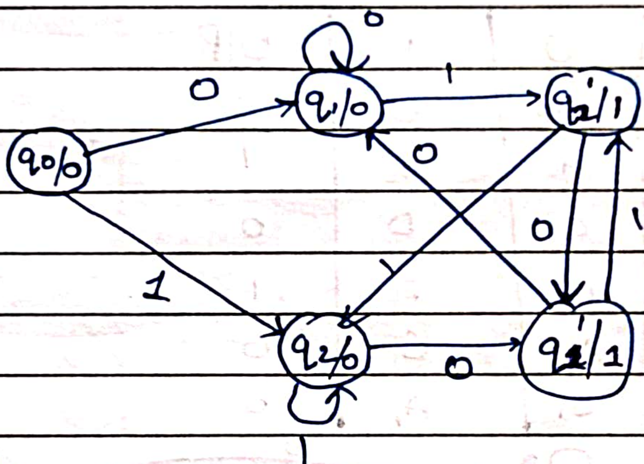




Q:- 2



Moore Machine :→

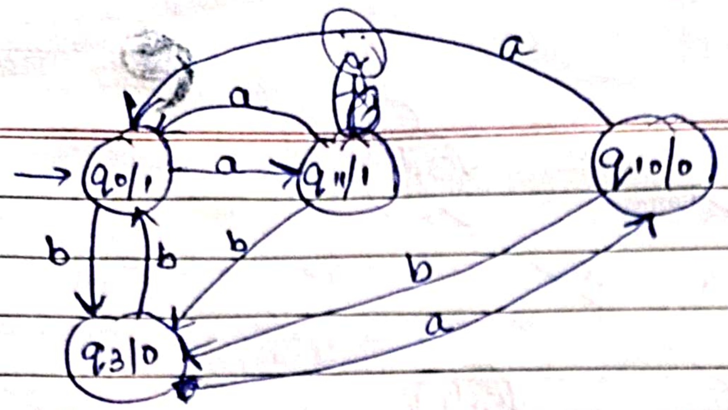


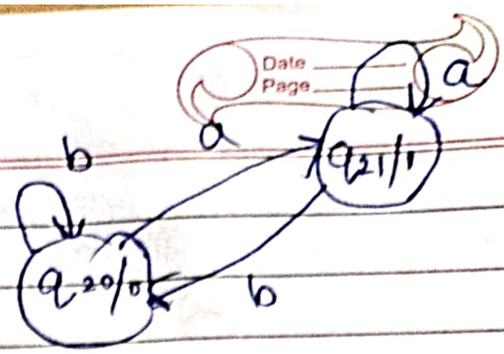
Q:-3

State	a	b
q_0	q_3,0	q_1,1
q_1	q_0,1	q_3,0
q_2	q_2,1	q_2,1
q_3	q_1,0	q_0,1

State	a	b	o/p
q_0	q_3	q_{11}	1
q_{10}	q_0	q_3	0
q_{11}	q_0	q_3	1
q_{20}	q_{21}	q_{20}	0
q_{21}	q_{21}	q_{20}	1
q_3	q_{10}	q_0	0

When we convert moore machine to mealy machine the no. of states remain same. But when we convert mealy to moore, the no. of state increase. The max. no. of states can be $(x \times y)$ where 'x' is states and 'y' is o/p's.

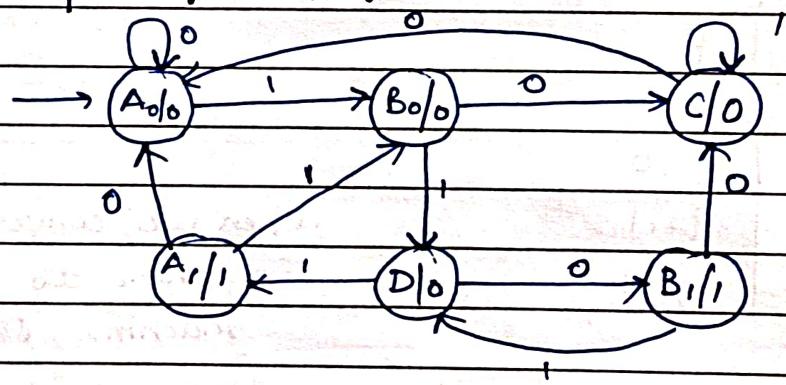




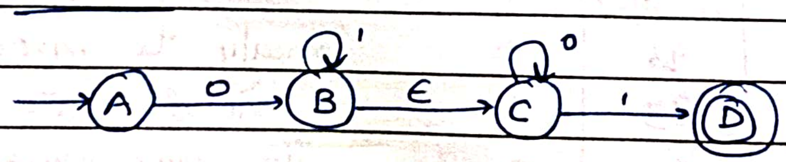
Q:-4

State	0	1
A	A,0	B,0
B	C,0	D,0
C	A,0	C,0
D	B,1	A,1

State	0	1	O/P
A_0	A_0	B_0	0
A_1	A_0	B_0	1
B_0	C	D	0
B_1	C	D	1
C	A_0	C	0
D	B_1	A_1	0

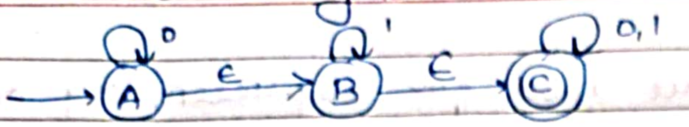


∈ NFA:-



*Every state on ∈ goes to itself.

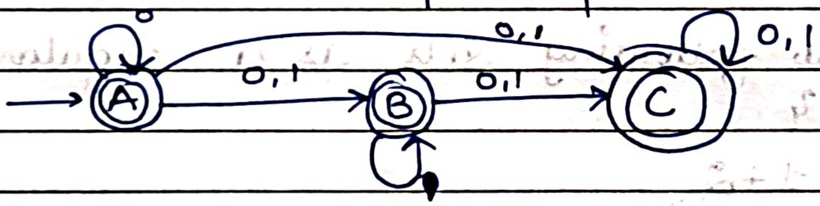
ϵ^* = ∈ closure. ⟶ All the states that can be reached from a particular state only by seen the ε symbol is known as ε*.

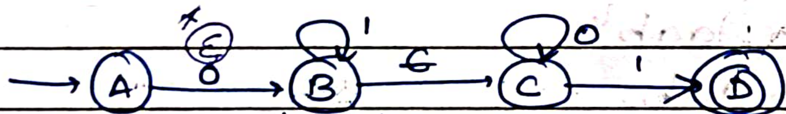
1. 

State	ε*	0	ε*
• A	A	A	A,B,C
	B	ϕ	—
	C	C	C
• B	B	ϕ	—
	C	C	C
• C	C	C	C

State	ε*	1	ε*
• A	A	ϕ	—
	B	B	B,C
	C	C	C
• B	B	B	B,C
	C	C	C
• C	C	C	C

State	0	1
→ A	A,B,C.	B,C
B	C	B,C
C	C	C

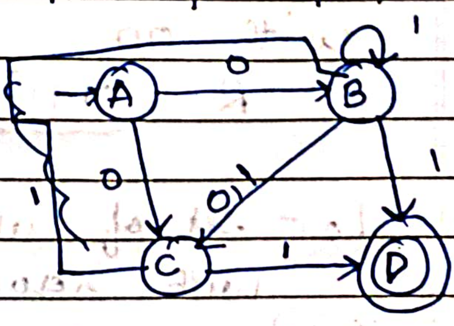


2. 

State	ε*	0	ε*
• A	A	B	B,C
• B	B	ϕ	—
	C	C	C
• C	C	C	C
• D	D	ϕ	ϕ

State	ε*	1	ε*
• A	A	ϕ	—
• B	B	B	B,C
	C	D	D
• C	C	D	D
• D	D	ϕ	ϕ

State	0	1
→ A	B,C	ϕ
→ B	C	B,C,D
→ C	C	D
→ D	ϕ	ϕ



x.   Tanveer
Geetoryally

3. Regular expression:-
Regular expression are used for representing certain set of strings in algebraic fashion.

Algebric fashion:-
1. Any terminal symbol i.e. symbols that belongs to Σ including Λ and ϕ are re
2. The union of 2 regular expression is also a regular expression.
3. The concatenation of 2 regular expression is also a regular expression.
4. The iteration or closure of a regular expression is also a regular expression.
5. The regular expression over sigma are precisely these obtain recursively by the application of above rules onces or several time.

Q:→ Describe the following sets as a regular expression.
(a) $\{0, 1, 2\}$
$R = 0 + 1 + 2$
(b) $\{abb, a, b, aab\}$
$R = abb + a + b + aab$.
(c) $\{\Lambda, 0, 00, 000, ...\}$
$R = 0^*$
(d) $\{1, 11, 111, 1111, ...\}$
$R = 1^+$ → + means null nahi hai
BKi sari string hai

Q:- L_1 = Set of all strings of 0's and 1's ending with 00.
$R = \underbrace{(0+1)^*}\, 00$

Q:- L_2 = set of all strings of 0's and 1's starting with zero and ending with 1.
0 —— 1

$R = 0(0+1)^* 1$

Q:- $L_3 = \{\Lambda, 11, 1111, 111111 \ldots\}$

$R = (11)^*$

Q:- Define L which uses characters a and b such that all words begin with 'a'.

$R = a(a+b)^*$

Q:- Define L of all strings which uses characters a and b that have atleast 2 letters and beginning & ending letter is 'a'. and in b/w there can be any word using 'b'.

$R = a(b)^* a$

Q:- Write a regular expression for set of strings of a and b whose fifth symbol from last is a.

$R = (a+b)^* a (a+b)^4$

Q:- Write a regular expression for the language $L = \{a^{2n} b^{2m+1}, n \geq 0, m \geq 0\}$

$R = (aa)^* (bb)^* b$

Q:- $L = \{w \in \{0,1\}^* : w$ has no pair of consecutive zeros

$R = (\Lambda + 0) \cdot (1 + 10)^*$

Q:- $L = \{a^n b^m : (n+m)$ is even$)$

$R = (aa)^* (bb)^* + (aa)^* a (bb)^* b$