DATE :

8/06/24

Finite State Machine (FSM)

- Finite state Machine is also known as Finite Automata. (FA)

1) Symbol – a, b, c, 0, 1, 2, 3
2) Alphabet – Σ Collection of symbols

   Eg:- {a, b}, {d, e, f, g}, {0, 1, 2}
3) String :- Sequence of symbols

   Eg:- a, b, 0, 1, aa, bb, ab, 01
4) Language – Set of strings.

   Eg:- s

   L1 = set of all strings of length 2

   = {00, 01, 10, 11}

Powers of 'Σ' $\Sigma = \{0, 1\}$

$\Sigma^0 \Rightarrow$ set of all strings of length 0 $\Sigma^0 = \{\ \}$

$\Sigma^1 \Rightarrow$ " length 1 $\Sigma^1 = \{0, 1\}$

$\Sigma^2 \Rightarrow$ " Length 2 $\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 \Rightarrow$ " length 3 $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$\Sigma^n \Rightarrow$ set of all strings of length n. $\Sigma^n = ?$

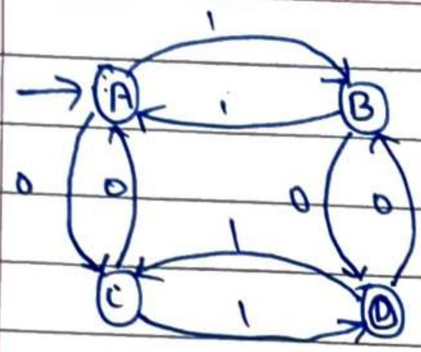
- Cardinality :- Number of Elements in a set.

$\hookrightarrow \boxed{\Sigma^n = 2^n}$

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots \cup \Sigma^n$$

Finite state Machine
(Finite Automata)

- FA with Output
  - Moore Machine
  - Mealy Machine
- FA without output
  - DFA
  - NFA
  - ε-NFA



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

$Q \Rightarrow$ set of all states. $\{A, B, C, D\}$

Σ = inputs $\{0, 1\}$

q_0 = Start | Intial state. $\{A\}$

F = set of Final states $\{D\}$

δ = transistion Function From

$Q \times \Sigma \rightarrow Q$

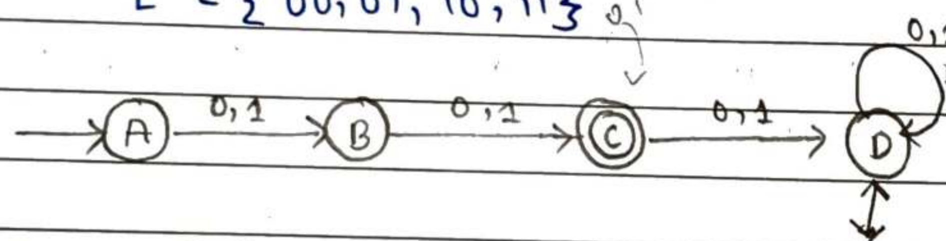
10/06/24

- Deterministic Finite Automata (DFA)

→ Construct a DFA that accepts set of all strings over (0, 1) of length 2.

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

$L = \{00, 01, 10, 11\}$



Dead state / Trap State.

Eg:- 00

$A \xrightarrow{0} B \xrightarrow{0} C$ - Final state.

Eg:- 10

$A \xrightarrow{1} B \xrightarrow{0} C$ - Final state

Eg:- 001

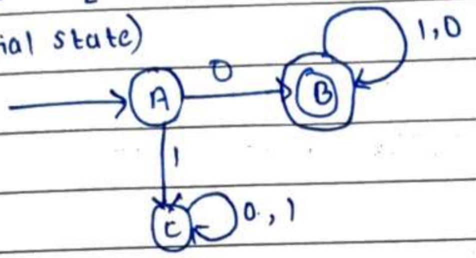
$A \xrightarrow{0} B \xrightarrow{0} C \xrightarrow{1} D$ - Not a Final state.

Q. Construct a DFA that contains

→ L1 = Set of all strings that start with '0'

L1 = { 0, 1, 00, 01, 100 01, 000, 001, 00, 100, 0000, 0100, ... }

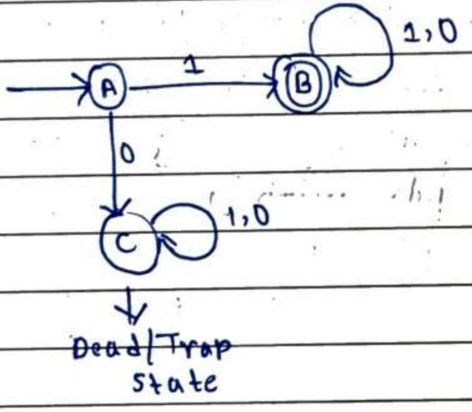
(Intial state)



Q.) Design a DFA that accepts a language over a input 0,1 and contains all the strings that ~~accepts~~ ~~starts with~~ one

→ L1 = Set of all strings over input 0,1

L = { 00, 01, 10, 11 }

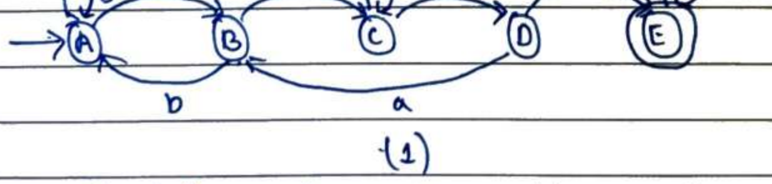


Q.) Construct a DFA that accepts any ~~integer~~ strings over {a,b} that does not contain the string aabb in it

→ Σ = {a,b}

① Try to Design a simpler problem.

Let us construct a DFA that accepts all strings over {a,b} that contains the string aabb in it.



(1)

L = {a,b}, aaabb, aabb, aabb, aa, bb, aab, abb, ...}

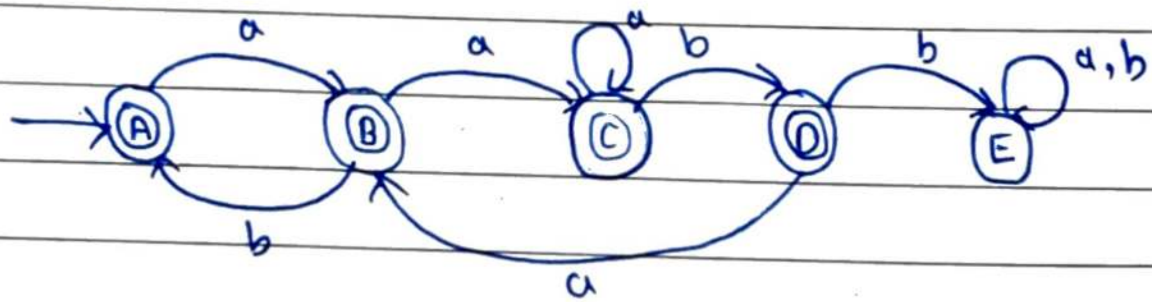
L_1 = { babaabb } L_2 = { aabaabb }

(2) Now, to construct a DFA that accepts any string over {a,b} that doesn't contain aabb, we will complement the previous stage (1)

NOTE:- FLIP THE STATE

- Make the Final state into a non-final state.
- Make the Non-final state into Final state.

Thus,



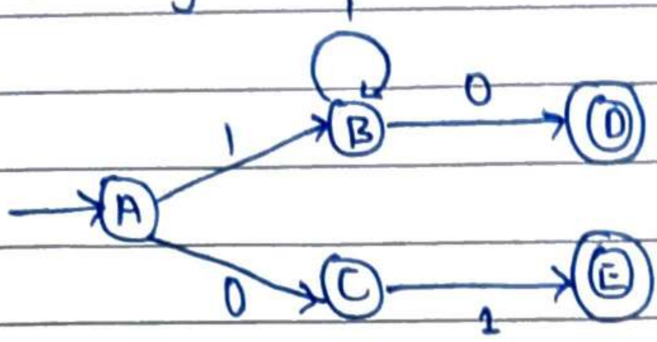
Assignment:-

1. Design a DFA over the alphabet {0,1} that contains all the string ~~110~~ that contains the substring 110
2. Design a DFA over the alphabet {0,1} that accepts all the strings that ends with 11
3. Design a DFA that accepts the following string
   (1) 1010
   (2) 000111
   (3) Accepts odd no. of zeroes.
   (4) Accepts Even no. of zero's

Deterministic Finite Automata (DFA)

Q. How to figure out what a DFA recognizes.

→



10

1 1 1 1 1 0

A B D

01 ✓

one binary digit '1'

L = { Accepts the string 01 or a string of atleast one '1' followed by a zero }

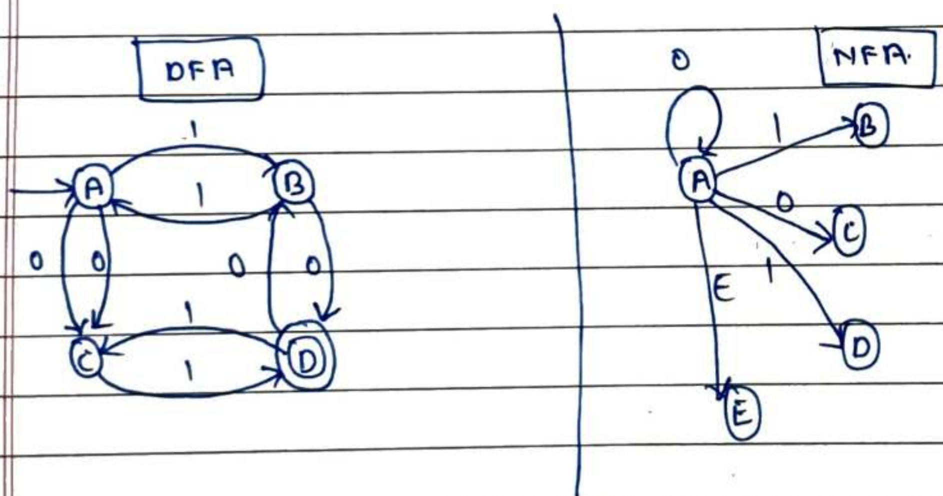
Non-Deterministic Finite Automata :-

⇒ DETERMINISM

- >> In DFA, given the current state we know what the next state will be.
- >> It has only one unique next state.
- >> It has no choices or randomness
- >> It is simple and easy to Design.
- >>

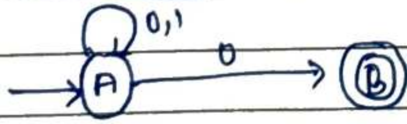
NON DETERMINISM :-

- >> In NFA, given the current state there could be Multiple next states.
- >> The Next state may be chosen at random.
- >> All the next states may be chosen in parallel.



→ NFA is more powerful than DFA, as in NFA parallel Operations are possible.

Example :- NFA



L = { set of all strings that End with zero }

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

$Q \rightarrow$ Set of all states — $\{A, B\}$

$\Sigma \rightarrow$ Inputs. — $\{0, 1\}$

$q_0 \rightarrow$ Start state | Intial state. — A

$F \rightarrow$ Set of Final states. — B

$\boxed{\delta = Q \times \Sigma \rightarrow 2^Q}$ — ?

$A \times 0 \rightarrow A$

$B \times 0 \rightarrow B$

$A \times 1 \rightarrow A$

$B \times 0 \rightarrow \emptyset$

$B \times 1 \rightarrow \emptyset$

All possible states :-

$\rightarrow$ A, B, AB, BA ......

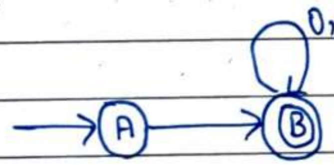
A' = AB, AC, BC, AB, ABC, ϕ.

$\boxed{2^3 = 8}$

Q.] Design a NFA that will accept the following Language where, L = { set of all string that start with zero }

ANS. SOLUTION :-

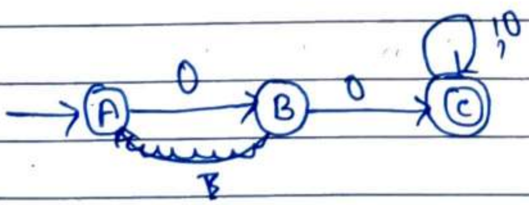
L = { 0, 01, 00, 010, 001, 011, ... }



Q.] Design a NFA that will accept the following Language where, L = { set of all string that start with 00 }

ANS. SOLUTION :-

L = { 00, 001, 0010, 0011, 00101 ..... }



Q. Construct a NFA that accepts sets of all strings over $\{0,1\}$ of length 2.

ANS. **Solution:-** $\Sigma = \{0,1\}$

$L = \{00, 01, 10, 11\}$

$$\to A \xrightarrow{0,1} B \xrightarrow{0,1} C$$

Eg:- ① 00

$\to A \xrightarrow{0} B \xrightarrow{0} C$ string Accepted.

② 10

$\to A \xrightarrow{1} B \xrightarrow{0} C$ string Accepted.

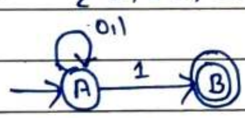
③ 110

$\to A \xrightarrow{1} B \xrightarrow{1} C \xrightarrow{0} \phi$ String Rejected

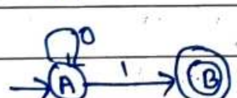
Q. Design NFA for the Following:-

(1.) $L1 = \{$ Set of all string that ends with '1' $\}$

$\Rightarrow$ $L1 = \{1, 01, 11, 001, 111, 0001, 0101, 0111, \ldots\}$



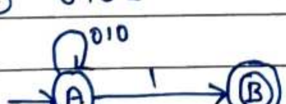
eg. ① 01



string Accepted

$A \xrightarrow{0} A \xrightarrow{1} B$

② 0101



string Accepted.

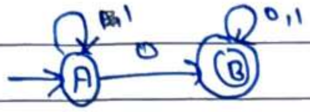
③ 10

$\to A \xrightarrow{1} B \xrightarrow{0} \phi$ string Rejected.

PAGE NO.
DATE :

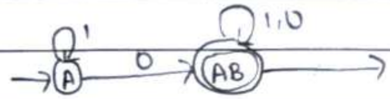
(2.) L2 = { set of all string that contain '0' }

L2 = { 0, 10, 110, 001, 010, 100, 0000, ... }



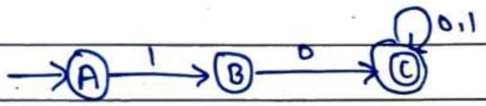
	0	1
A	A, B	A
B	B	B

	0	1
A	AB	A
AB	AB	AB



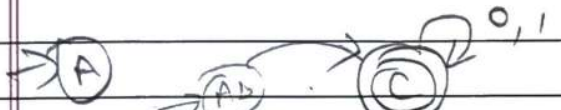
(3.) L3 = { set of all strings that start with '10' }

→ L3 = { 10, 101, 1010, 1011, 10101, 10111, ... }



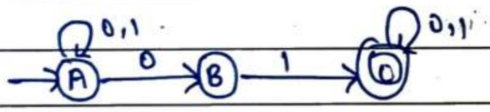
	0	1
→A	∅	B
B	C	∅
C	C	C

	0	1
A	∅	B
B	C	∅
C	C	C
AB	C	B



(4.) L4 = { set of all string that contain 01 }

→ L4 = { 01, 101, 001, 1001, 1101, ... }

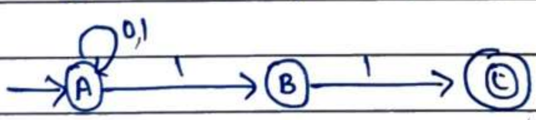


	0	1
→A	A, B	A
B	∅	C
C	C	C

	0	1
→A	AB	A
AB	AB	AC
AC	ABC	AC
ABC	ABC	AC

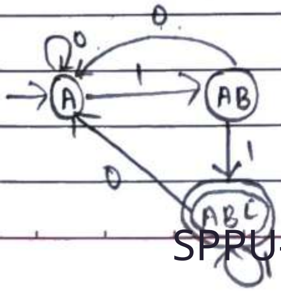
(5.) L5 = { set of all strings that ends with '11' }

→ L5 = { 11, 011, 111, 0111, 0011, 1011, 1111, 10011, ... }



	0	1
→A	A	A, B
B	∅	C
C	∅	∅

	0	1
→A	A	AB
AB	A	ABC
ABC	A	ABC



ASSIGNMENT 2:-

For the previous 5 Questions, Design a DFA.

———— X ———— X ———— X

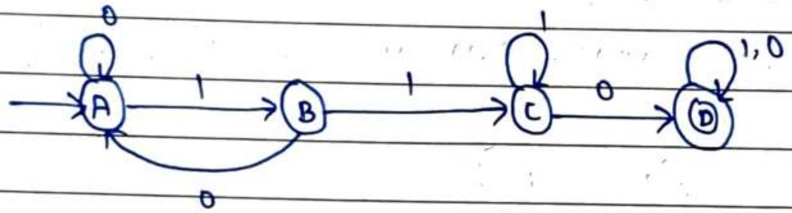
21/06/2024

ASSIGNMENT : NO: 1 : (ONE):-

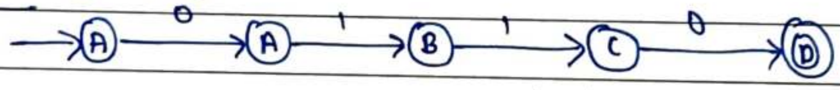
Q1.] Design a DFA over the alphabet {0,1} that contains all the strings that contain the subtring 110.

ANS. SOLUTION:-

Here, L = { 110, 1110, 0110, 1101, 1100, 11011, ... }



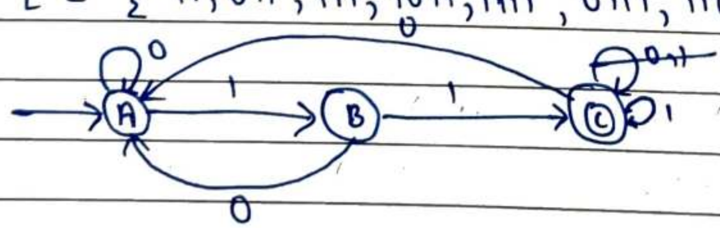
For Eg:- ① 0110



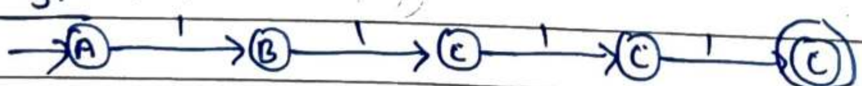
Q2.] Design a DFA over the Alphabet {0,1} that accepts all the strings that ends with 11

ANS. SOLUTION:-

L = { 11, 011, 111, 1011, 1111, 0111, 11011, ... }



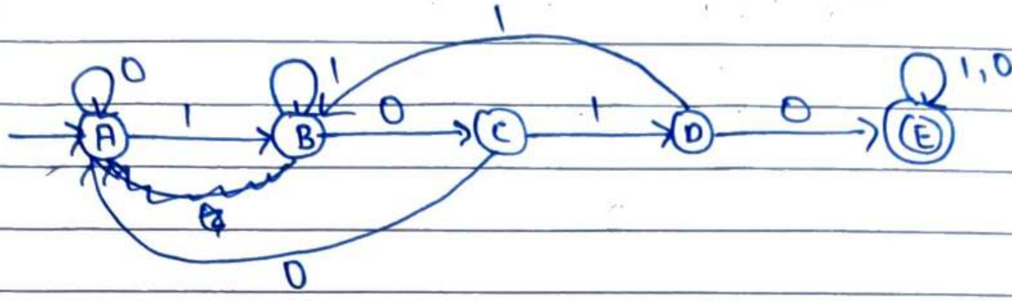
For Eg:- 1111



Final state

Q3. Design a DFA that accepts the following strings.

① 1010



L = { 01010 , 111010 , 111010110 , 000101011 , ... }

For Eg:- 111010

$\rightarrow A \xrightarrow{1} B \xrightarrow{1} B \xrightarrow{1} B \xrightarrow{0} C \xrightarrow{1} D \xrightarrow{0} E$

Final state

String Accepted.

For Eg:- 111011

$\rightarrow A \xrightarrow{1} B \xrightarrow{1} B \xrightarrow{0} C \xrightarrow{1} D \xrightarrow{1} B$

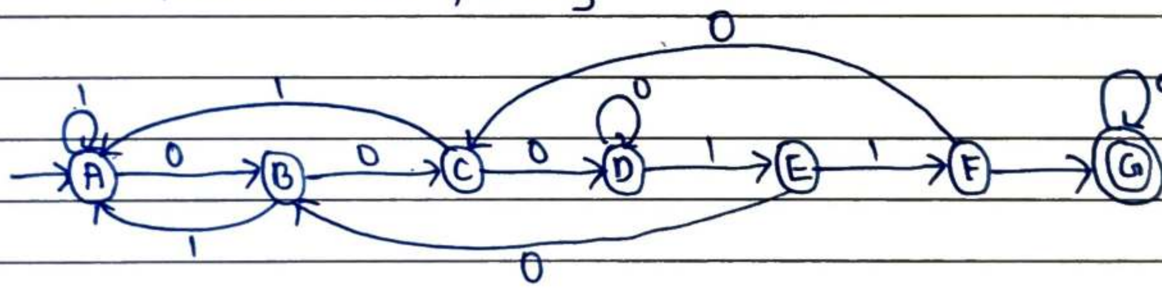
String Declined

Not Final state

② 0001117

L = { 111 000111 , 0110001118 , 011001100111 , 0001111110 , 1000111001 , ... }

⟹



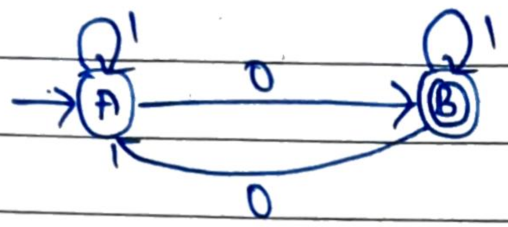
For Eg:- 111000111

$\rightarrow A \xrightarrow{1} A \xrightarrow{1} A \xrightarrow{1} A \xrightarrow{0} B \xrightarrow{0} C \xrightarrow{0} D \xrightarrow{1} E \xrightarrow{1} F \xrightarrow{1} G$

DATE :

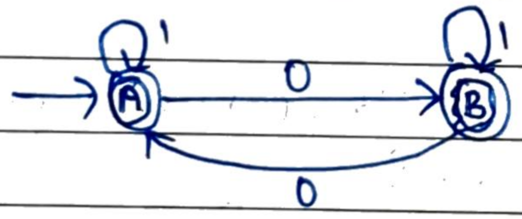
(3.) DFA that Accepts odd No. of zeroes.

→ L = { 0, 0001, 011, 101, 100000, 1000001, ... }

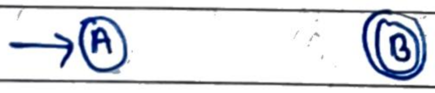


(4.) DFA that Accepts Even Number of zeroes.

→ L = { 100, 10000, 100001, 1001, 001, 1000000, ... }



OR



NFA - DFA Conversion :-

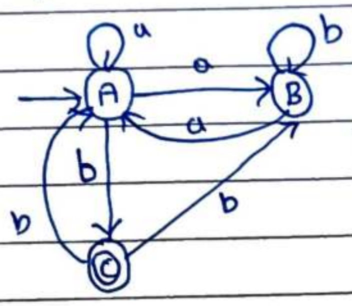
Q] Find the Equivalent DFA for the NFA given by $M = [\{A, B, C\}, (a, b), \delta, A, \{C\}]$ where δ is given by

→

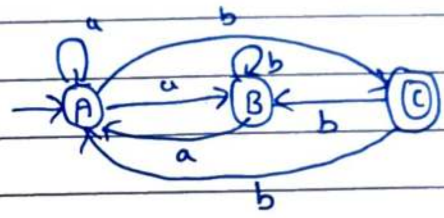
	a	b
→A	A,B	C
B	A	B
(C)	-	A,B

ANS. SOLUTION:-

NFA ⇒



OR



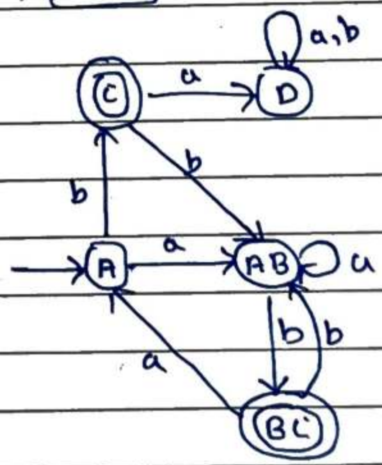
Now,

	a	b
→A	AB	C
AB	AB	BC
(BC)	A	AB
(C)	D	AB
D	D	D

The Equivalent DFA for the above NFA is:-

$M = [\{A, AB, BC, C, D\}, (a, b), \delta \to Q \times \Sigma = 2^Q, A, \{BC, C\}]$

Thus, $\boxed{\text{DFA}}$:-



NOTE:-

All the states that contain the First Final state i.e. in NFA should be taken as Final state in DFA.

For Eg:- BC and C are Final states in DFA as C is Final state in NFA.

Subset Construction Method.

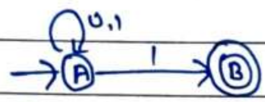
PAGE NO.
DATE :

Q.) CONVERT NFA TO DFA

→ L = { set of all strings over (0,1) that ends with '1' }

ANS. Solution:- Subset Construction Method.

$\Sigma = \{0,1\}$ L = { 1, 01, 11, 101, 001, 111, 1101, .. }



- This is a NFA.

Now

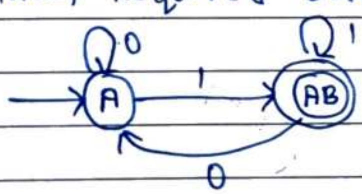
	0	1
A	{A}	{A,B}
B	$\emptyset$	$\emptyset$

Now,

	0	1
→ A	A	AB
AB	A	AB

Here, $\boxed{AB}$ → single state.

Thus, Required DFA is.



Q.) Given Below is the NFA For a language.

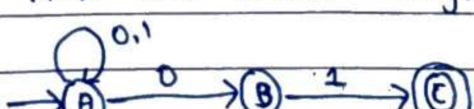
→ L = { set of all strings over (0,1) that ends with '01' }

Construct its Equivalent DFA.

ANS. SOLUTION:- $\Sigma = \{0,1\}$

① L = { 01, 101, 001, 0001, 0101, 1101, 00001, 10101, 11101, 00101, ... }

The NFA for above language is:-



Now, state Transistion Table:-

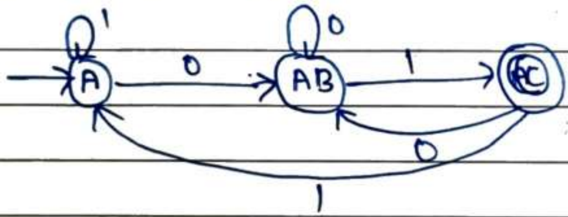
	0	1
A	A, B	A
B	∅	C
C	∅	∅

Now, For the DFA there are no multiple states

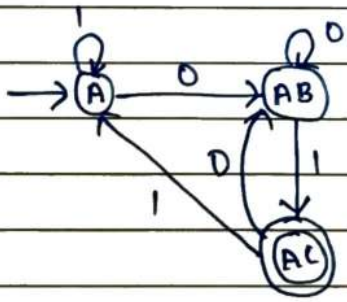
∴

	0	1
→A	AB	A
AB	AB	AC
AC	AB	A

Thus, The Required DFA is:-



OR

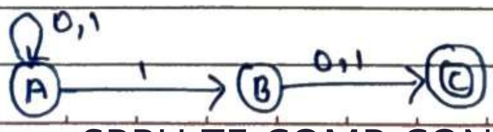


Q. Design an NFA for a language that accepts all strings over {0,1} in which the second last symbol is always '1'. Then convert it to its Equivalent DFA.

ANS. SOLUTION:- $\Sigma = \{0,1\}$

L = { 10, 11, 010, 111, 110, 1010, 1110, 1101010, 111110, ...}

The NFA for above language is



For Eg:- 1010

$\rightarrow A \xrightarrow{1} A \xrightarrow{0} A \xrightarrow{1} B \xrightarrow{0} C$

Final state.

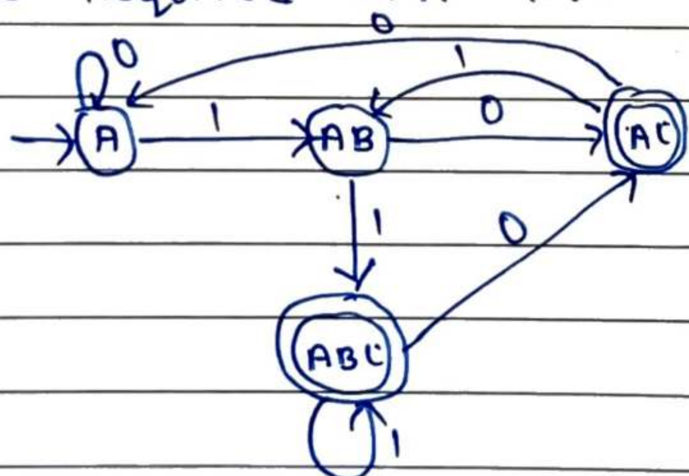
State Transistion Table for NFA:-

	0	1
→A	A	{A,B}
B	C	C
C	∅	∅

Now, For DFA;

	0	1
→A	A	AB
AB	AC	ABC
AC	A	AB
ABC	AC	ABC

The Required DFA is:-



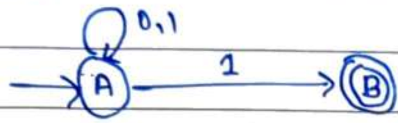
29/07/24 MON.

ASSIGNMENT : NO : 2 : (TWO) :-

Convert the Given NFA to DFA.

(1.) L = { Set of all strings that end with '1' }

⇒ L = { 1, 01, 11, 001, 011, 101, 111, 1011, 1001, 0001, 11111, ... }



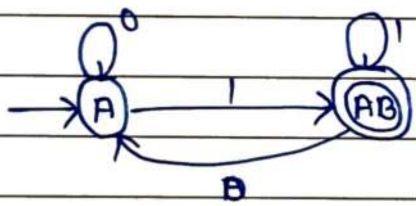
This is a NFA.

Now, State Tranisistion Table:-

	0	1
→ A	A	{A, B}
(B)	∅	∅

Now, for DFA

	0	1
→ A	A	AB
(AB)	A	AB

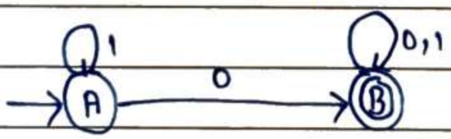


This is the required DFA.

(2.) L = { Set of all string that contain '0' }

→ L = { 0, 10, 01, 001, 100, 110, 101, 1001, 0001, 1000, ... }

The NFA for above Language is :-



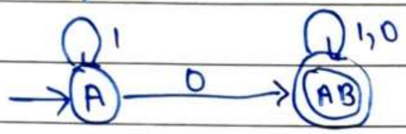
State Transistion Table

	0	1
A	A, B	A
B	B	B

Now, By Using subset construction Method:-

	0	1
→A	AB	A
AB	AB	AB

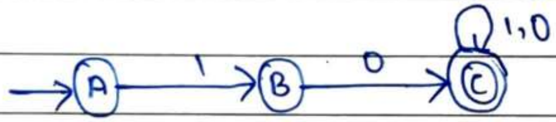
Thus, Required DFA is :-



(3.) L3 = { set of all string that start with '10' }

→ L3 = { 10, 101, 1011, 1010, 10110, 101111, 101010, ... }

The NFA for above language is:-



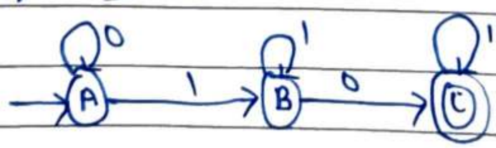
State transisistion Table.

	0	1
→A	∅	B
B	C	∅
C	∅	C

Now, By Using Subset construction Method;

	0	1
→A	∅	B
B	C	∅
C	C	C

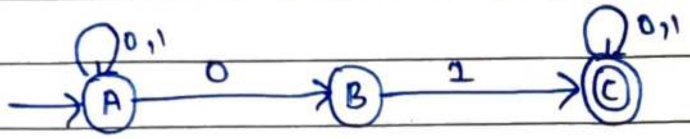
The Required DFA is



(4.) L4 = { set of string that contain 01 }

→ L4 = { 01, 101, 001, 1001, 1101, 10111, 1001101, ... }

The required NFA is:-



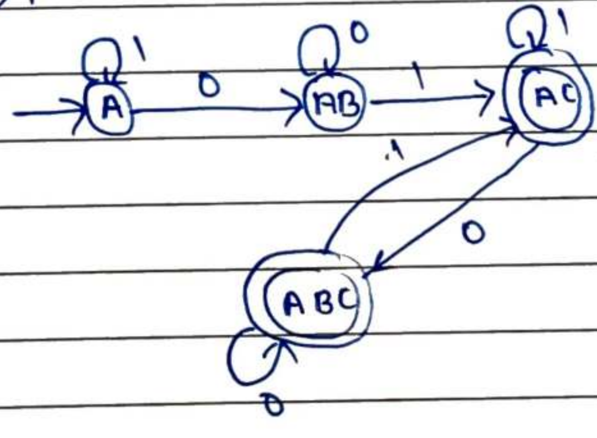
State Transition Table:-

	0	1
→A	A, B	A
B	ϕ	C
C	C	C

Now, By using subset construction method:-

	0	1
→A	AB	A
AB	AB	AC
AC	ABC	AC
ABC	ABC	AC

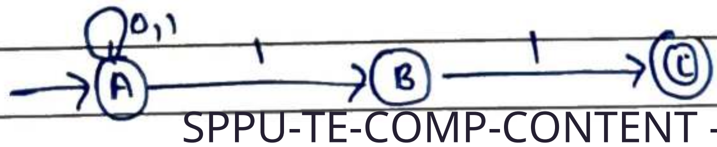
→ (consider combined state)



(5.) L5 = { set of all strings that end with '11' }

→ L5 = { 11, 11011, 011, 111, 1111, 1011, 10011, 0011, ... }

The NFA is:-



- State Transition Table:-

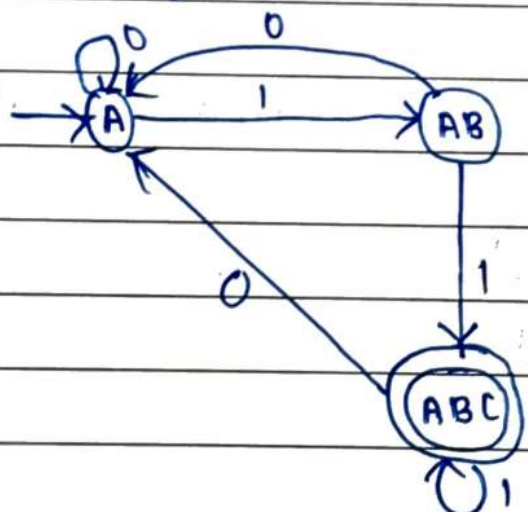
	0	1
→A	A	A, B
B	∅	C
C	∅	∅

Now, By Using Subset Construction Method:-

∴

	0	1
→A	A	AB
AB	A	ABC
C	A	ABC

- The Required DFA is:-

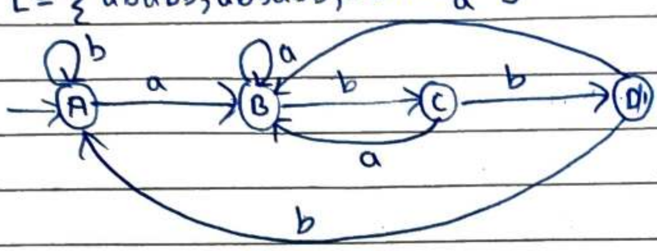


MOORE MACHINE :-

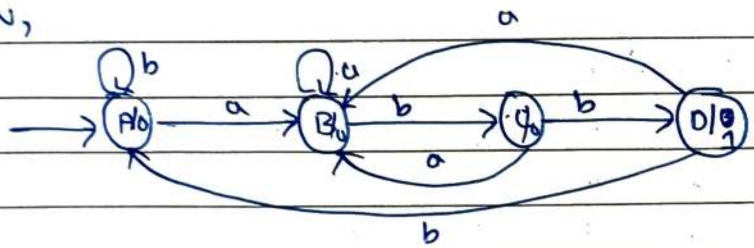
Q.] Construct a Moore machine that count the occurrences of the sequence 'abb' in any input string over {a,b}

ANS. SOLUTION :- $\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

DFA :- L = { ababb, abbabb, abbbabb }



Now,



- When the substring abb is encountered '1' will be printed by Moore machine and in all other cases the machine will give output '0'.

① • The Moore machine, the output is associated with state

OR

In the Moore machine, the output is a Function of current state

② • In the Mealy Machine, the output is a function of current state and current ~~out~~ Input

OR

In the Mealy Machine, the output is associated current state or ~~tran~~ current Transistion.

③ In Moore Machine, if input bits are 'n' then output bits are n+1

In Mealy Machine, if input bits are 'n' then output bits are 'n'

Tracing:-

case 1 : abb

	a	b	b
A	B	C	D
0	0	0	1

case 2 : ababb

	a	b	a	b	b
A	B	C	B	C	D
0	0	0	0	0	1

case 3 : abbabb

	a	b	b	a	b	b
A	B	C	D	B	C	D
0	0	0	1	0	0	1

case 4 :- abbbabb

	a	b	b	b	a	b	b
A	B	C	D	A	B	C	D
0	0	0	1	0	0	0	1

Q.) Construct the Moore machine for the following input alphabet $\Sigma = \{a, b\}$ and the output alphabet is $\Delta = \{0, 1\}$
(i.) aabab (ii.) abbb (iii.) ababb.

states	a	b	Outputs
$\rightarrow q_0$	q_1	q_2	0
q_1	q_2	q_3	0
q_2	q_3	q_4	1
q_3	q_4	q_4	0
q_4	q_0	q_0	0

Lets consider input string (i) aabab

1.)

	a	a	b	a	b
q_0	q_1	q_2	q_4	q_0	q_2
0	0	1	0	0	1

2.) abbb

	a	b	b	b
q_0	q_1	q_3	q_4	q_0
0	0	0	0	0

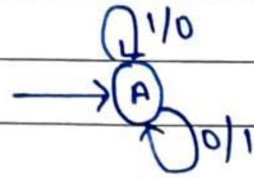
3.) ababb

	a	b	a	b	b
q_0	q_1	q_3	q_4	q_0	q_2
0	0	0	0	0	1

MEALY MACHINE :-

Q.) Construct a Mealy Machine that produces the 1's complement of any binary input string.

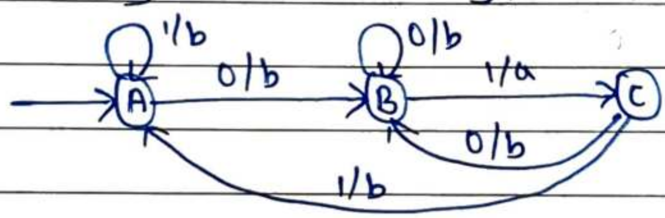
⇒



Here, 0/1
↓ ↓
Input Output.

Q.] Construct a Mealy Machine that prints 'a' whenever the sequence '01' is encountered in any input binary string.

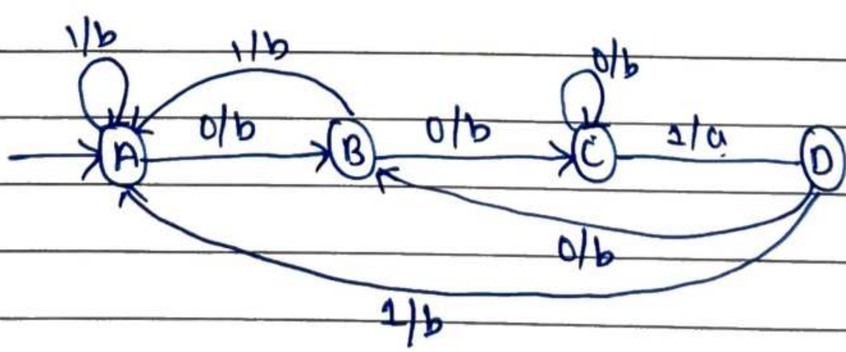
⇒ $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$



001 - bba.

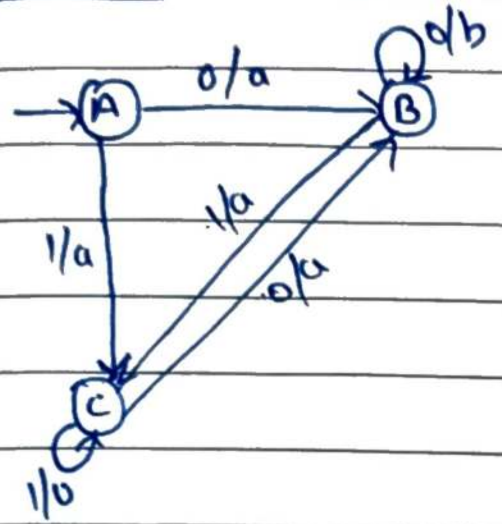
Q.] Constant - '001' O/P : 'a'

⇒ $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$



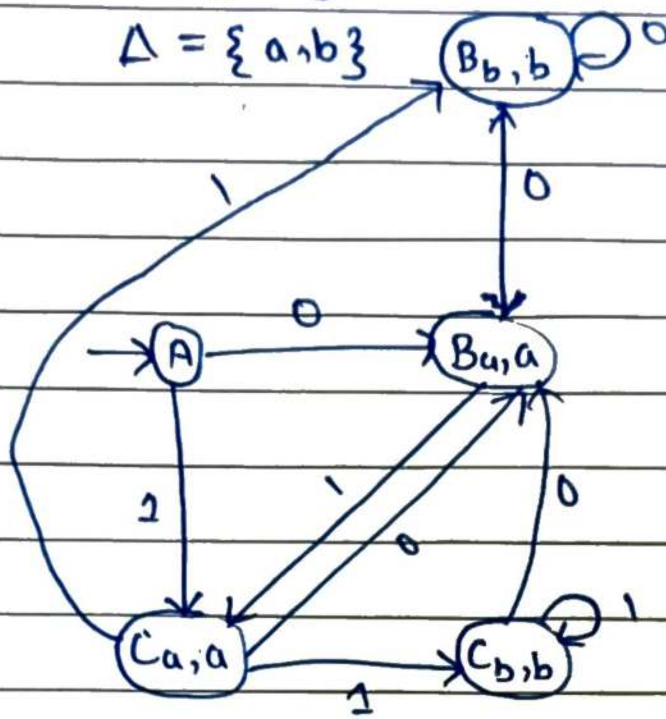
Ex:- 0010, 1011, 011

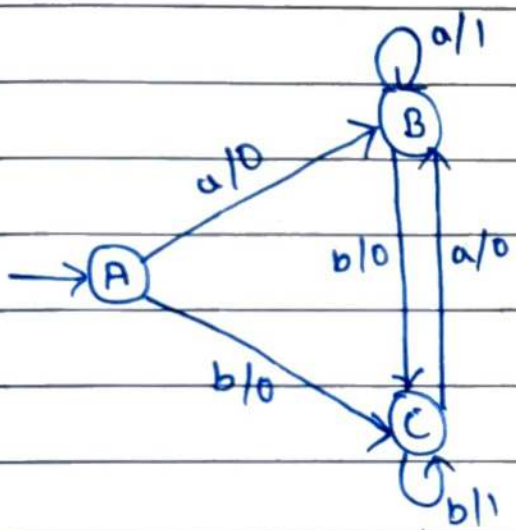
Convert : Mealy Machine to Moore Machine.



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

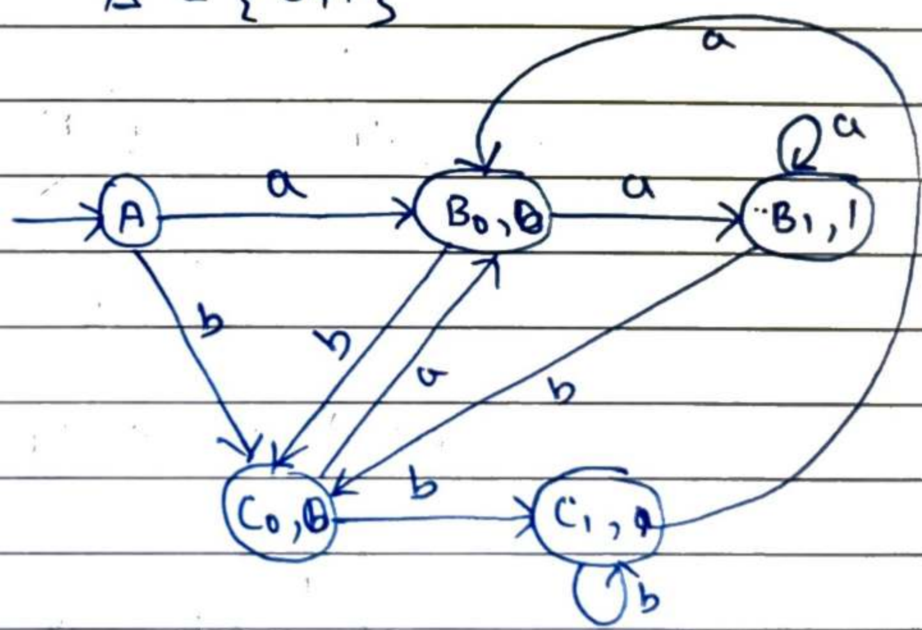
$\Delta = \{a,b\}$





$\Sigma = \{a,b\}$

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



DATE: 08/08/24

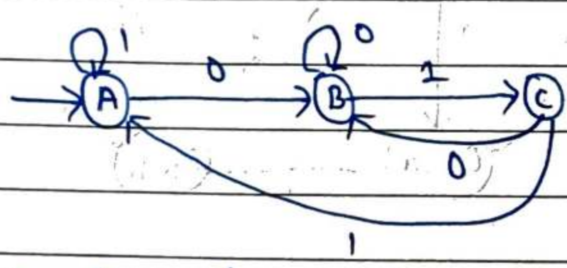
Conversion of Moore Machine to Mealy Machine.

→ Construct a Moore machine that prints 'a' whenever the sequence '01' is encountered in any input binary string and then convert it to its Equivalent mealy machine.

ANS.

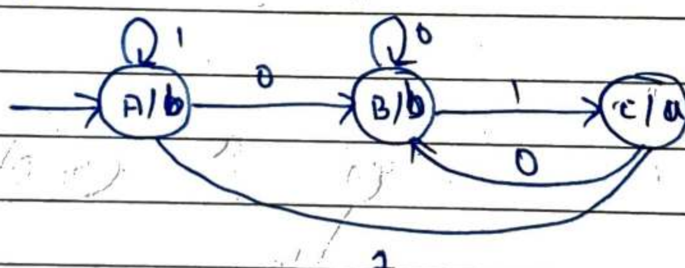
$\Sigma = \{0, 1\}$

$\Delta = \{a, b\}$



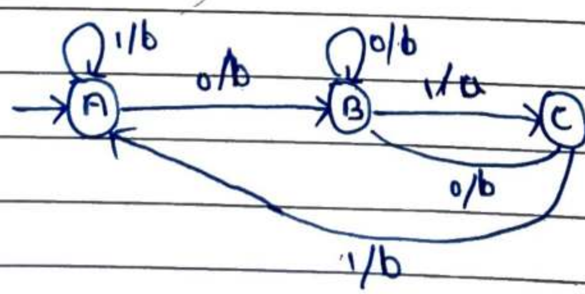
This is the DFA

Now, Moore Machine.



state	0	1	Output
→A	B	A	b
B	B	C	b
C	B	A	a

Now, Equivalent Mealy Machine ~~as~~



state	0	1
→A	B, b	A, b
B	B, b	C, a
C	B, b	A, b

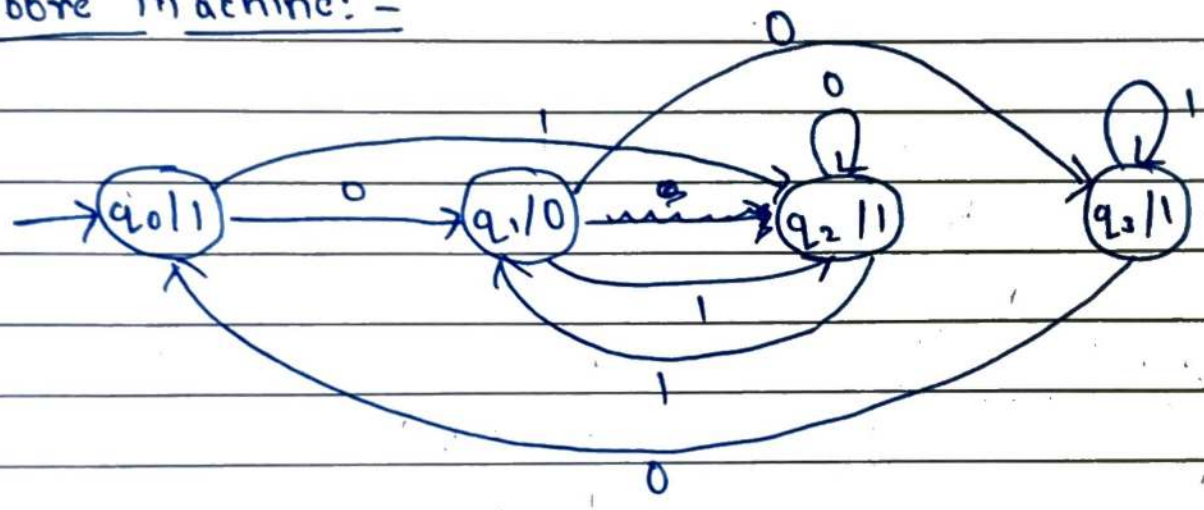
Q. Convert the given Moore Machine to its Equivalent Mealy Machine.

ANS. SOLUTION:- $\Sigma = \{0, 1\}$ $\Delta = \{0, 1\}$

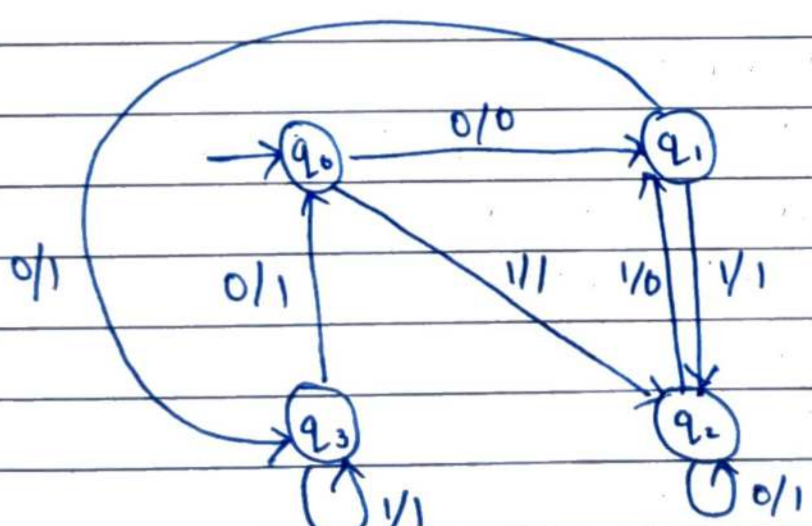
state	0	1	Output
$\rightarrow q_0$	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_1	1
q_3	q_0	q_3	1

state	0	1
$\rightarrow q_0$	$q_1, 0$	$q_2, 1$
q_1	$q_3, 1$	$q_2, 1$
q_2	$q_2, 1$	$q_1, 0$
q_3	$q_0, 1$	$q_3, 1$

Moore Machine:-



Mealy Machine:-



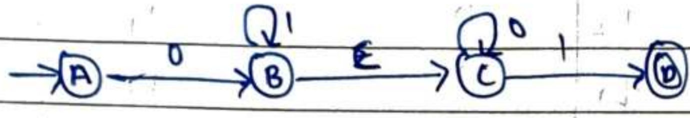
Epsilon (ε) - NFA

NOTE:- ε - NFA

↳ (Empty symbol) ⟶ (zero Length)

Def^n:- $\{Q, \Sigma, q_0, \delta, F\}$

where, $\delta : Q \times \Sigma \cup \epsilon \longrightarrow 2^Q$

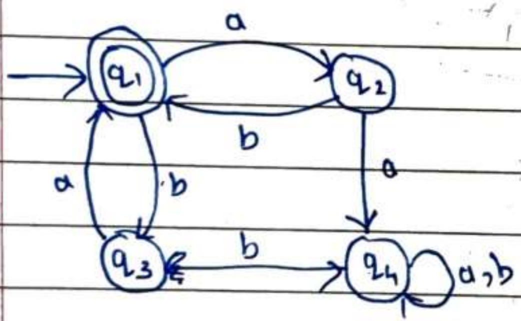
Eg:- 

Every state on ε goes to itself

IMP

Designing Regular Expressions:-

Q] Find the Regular Expression for the following DFA



$q_1 = \epsilon + q_2 b + q_3 a$ ——①

$q_2 = q_1 a$ ——②

$q_3 = q_1 b$ ——③

$q_4 = q_2 a + q_3 b + q_4 a + q_4 b$ ——④

Now, $q_1 = \epsilon + q_2 b + q_3 a$

Putting values of q_2 and q_3 from Eqn's ② and ③

$q_1 = \epsilon + (q_1 a) b + (q_1 b) a$

$q_1 = \epsilon + q_1 ab + q_1 ba$

$q_1 = \epsilon + q_1 (ab + ba)$

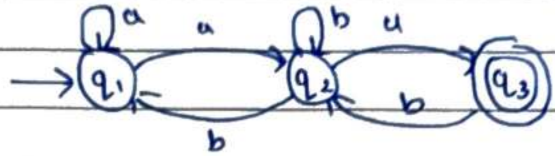
We know, $R = Q + RP$

where $R = QP^*$ .... (Arden's Theorem)

$\boxed{q_1 = \epsilon (ab + ba)^*}$

↳ (Regular Expression)

Q. Find the Regular Expression for the Following NFA



$q_3 = q_2 a \longrightarrow ①$

$q_2 = q_1 a + q_2 b + q_3 b \longrightarrow ②$

$q_1 = \epsilon + q_1 a + q_2 b \longrightarrow ③$

Now, $q_3 = q_2 a$

Now, substituting value of q_2 in above Eqn.

① $q_3 = (q_1 a + q_2 b + q_3 b) a$

$q_3 = q_1 aa + q_2 ba + q_3 ba \longrightarrow ④$

② $q_2 = q_1 a + q_2 b + q_3 b$

Putting value of q_3 from ①

$q_2 = q_1 a + q_2 b + (q_2 a) b$

$\therefore \underbrace{q_2}_{R} = \underbrace{q_1 a}_{Q} + \underbrace{q_2}_{R} \underbrace{(b + ab)}_{P}$

$\boxed{q_2 = (q_1 a)(b + ab)^*} \longrightarrow ⑤$

③ $q_1 = \epsilon + q_1 a + q_2 b$

Putting value of q_2 from ⑤

$q_1 = \epsilon + q_1 a + ((q_1 a)(b + ab)^*) b$

$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \underbrace{(a + a(b + ab)^*) b}_{P}$

$q_1 = \epsilon (a + (a(b + ab)^*) b)^*$

$q_1 = (a + a(b + ab)^* b)^* \longrightarrow ⑥$

Final state q_3

$q_3 = q_2 a$

$= q_1 a (b + ab)^* a$

Putting value of q_2 From ⑤

$$\boxed{q_3 = (a + a(b+ab)^* b)^* \, a(b+ab)^* a}$$

$\Rightarrow R$