16/1/24 Theory Of Automata & Compiler Design.

Page No.

Date

Module-1: Finite Automata.

★ **Chomsky Hierarchy:**

Grammar is divided into 4 types as follows:

I) Type 0 (Unrestricted Grammar) → superset of all types
- It is recognised by turing machine.
- These language is also known as recursively Enumerable language.
- Type 0 grammar include all formal grammar.

II) Type 1 (Context sensitive Grammar)
- Recognised by ~~Single~~ Linear bound Automata.

III) Type-2 (Context Free Grammar)
- Recognised by pushdown Automata.

IV) Type-3 (Regular Grammar)
- Recognised by Finite Automata.
- Most Restricted form of grammar

★ **Finite Automata (FA)**
- Applications:

① Used in text Editors.
② Used for recognising pattern using regular expreⁿ.
③ Used for designing of the combinaⁿ & sequential circuits using Mealy moore machine.
④ Can parse text to extract info. & structure data.
↳ Resolve.

* A ~~First~~ Finite Automata consists of:
$\{Q, \Sigma, q_0, \delta, F\}$. ⇒ 5 Tuple Definitions.

Q : Finite set of states.
Σ : Finite set of I/P symbols.
q_0 : Initial state.
δ : Finite set of Transitions.
F : Finite set of Final ~~set~~ States.

$\delta : Q \times \underbrace{\Sigma}_{(0,1)} \longrightarrow Q \cup \underbrace{\phi}_{\text{No. of transition.}}$

Finite Automata.

- Finite Automata W/ O/P
  - Mealy Machine
  - Moore Machine
- Finite Automata W/O O/P.
  - Deterministic Finite Automata (DFA). — Either 0 or 1 transin
  - Non-Deterministic Finite Automata (NDFA) or (NFA)
    - with Epsilon moves (Null move)
    - W/O Epsilon moves

Minimization of DFA.

Page No.
Date

- Conversion takes place from NFA to DFA.

W/ ϵ $\rightarrow$ W/O ϵ $\rightarrow$ W/O Non-DFA $\rightarrow$ DFA.
= Minimisatn of FA.

FA $\rightarrow$ RG (Regular Grammar).

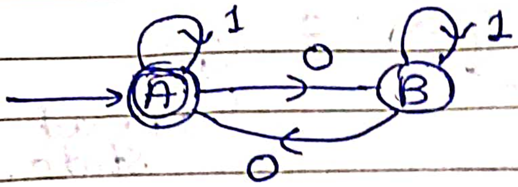
Q) Automata for even length:
$L = \{0_\epsilon 0, 11, 0_\epsilon 1, 0011, 1110, 0100, \phi.\}$

$\rightarrow$ ϕ is minimal state.
If it is ϕ, we can go w/o state.
$\therefore$ Init

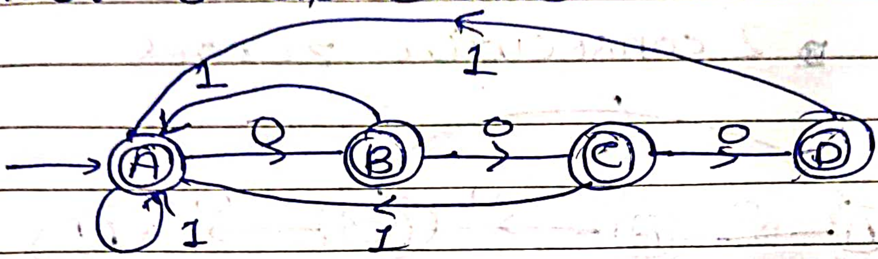
4/11

Q) Even no. of zeroes & any no. of one's.

$L = \{\emptyset, 1, 11, 00, 001, 010, 1010, 11100 \ldots\}$.



Q) Atmost 3 consecutive zeroes.



Tuple Definition: $(Q\ \Sigma\ \delta\ q_0\ F)$

$Q = \{A, B, C, D\}$

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

$q_0 = A$

$F = \{A, B, C, D\}$

$\delta = Q \times \Sigma \rightarrow Q$

Transition Table ⇒

Q \ Σ	0	1
A	B	A
B	C	A
C	D	A
D	$\emptyset$	A

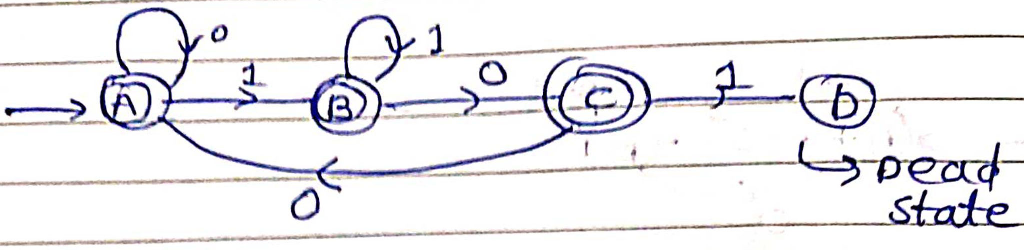
Simulation: I/P = 10100

$\delta(A, 10100) \rightarrow \delta(A, 0100) \rightarrow \delta(B, 100) \rightarrow \delta(A, 00)$

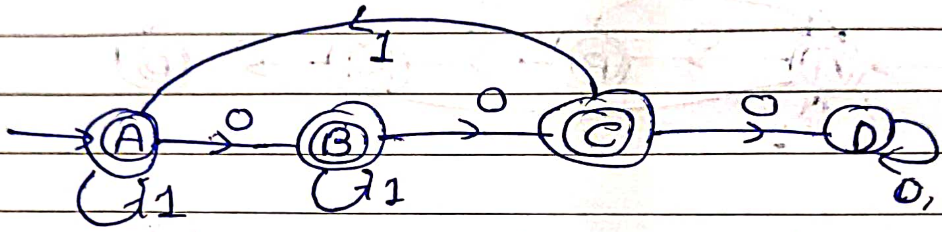
$\downarrow$

$C \leftarrow \delta(B, 0)$

Q) Automata not having 101 as a substring

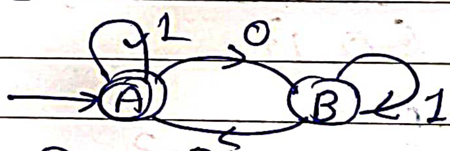


Q) At most 2 consecutive zeroes

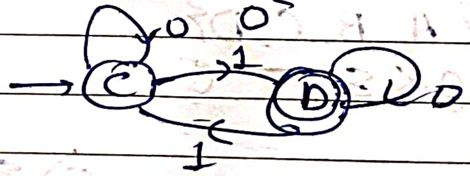


Q) Even 0's & odd 1's

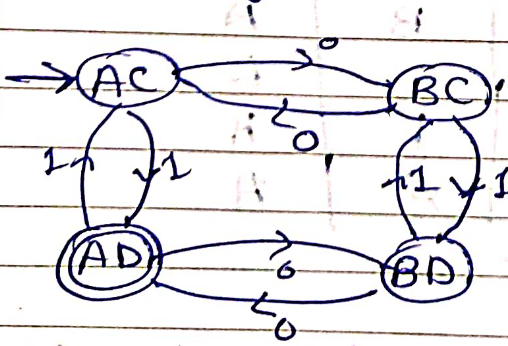
→ For Even 0's ⇒



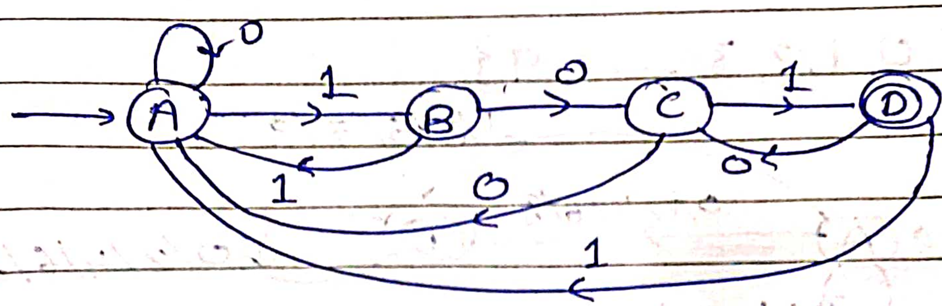
For odd 1s ⇒



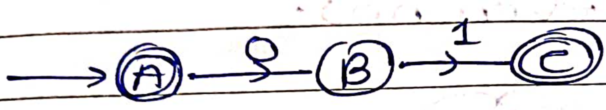
⇓

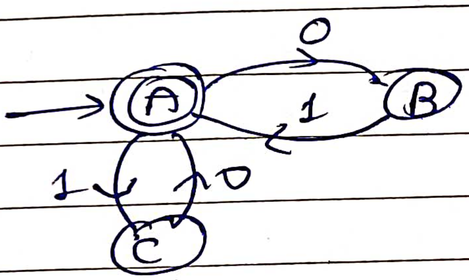


Q) Ending w/ 101

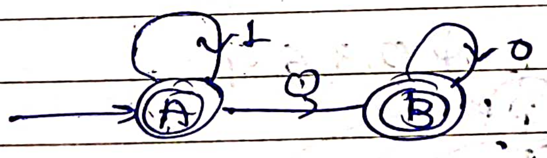


Q) Equal 0's & 1's

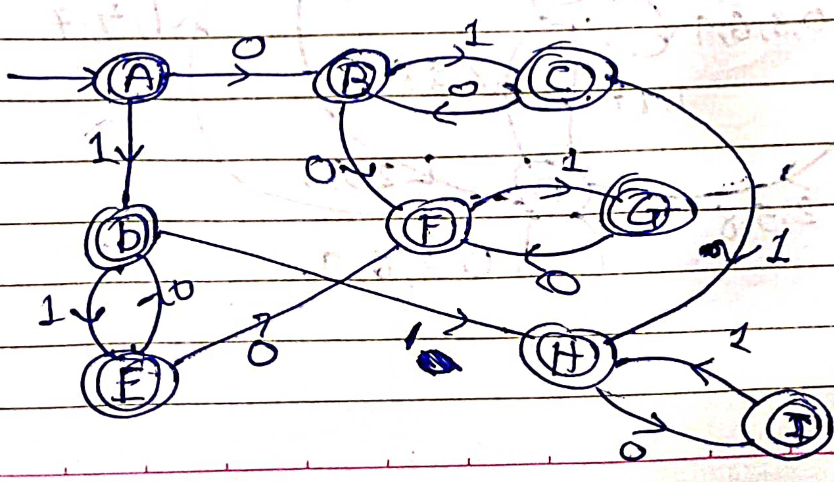




Q) No 0 is followed by 1. → eg (01 ✗)

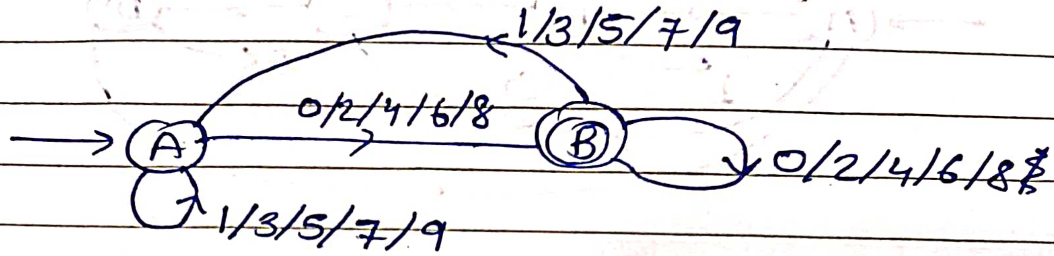


*Q) Atmost one pair of 0's & one pair of 1's.

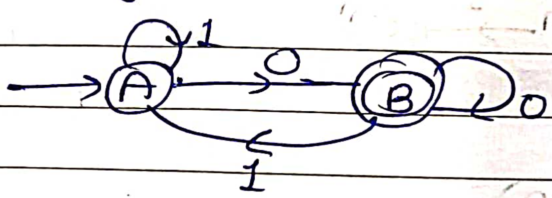


Q) Even Integers.

$\Sigma = \{0, 1, 2, 3, \ldots, 9\}$



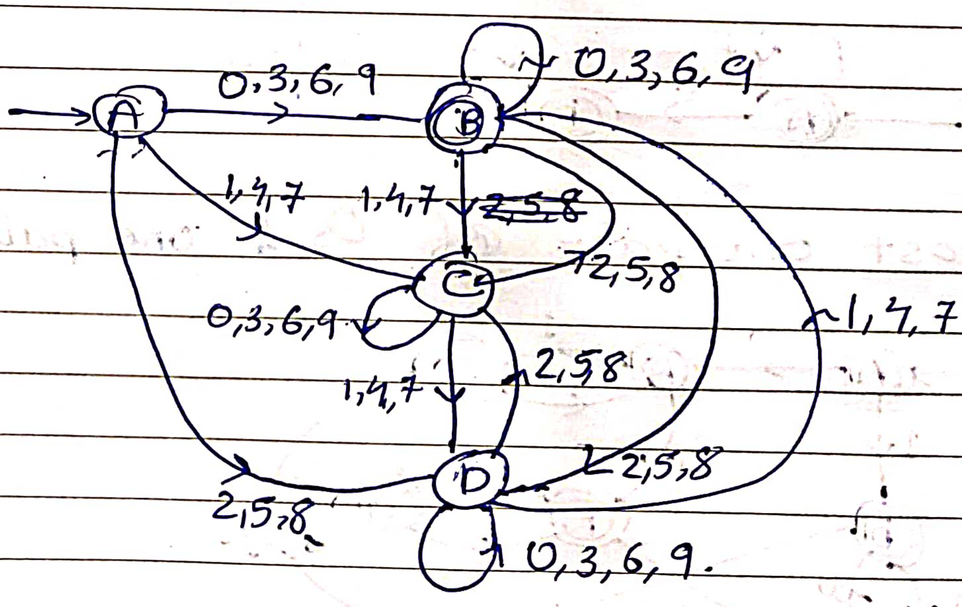
For Binary Nos., even nos.



Q) Integers Divisible by 3

$\Sigma = \{0, 1, 2, \ldots, 9\}$

- 1+4+7 div by 3
- 2+5+8 div by 3



30 → 0	40 → 1	50 → 2
31 → 1	41 → 2	51 → 0
32 → 2	42 → 0	52 → 1
33 → 0	43 → 1	53 → 2
⋮	⋮	⋮
39 → 0	49 → 1	59 → 2

For Binary Nos,

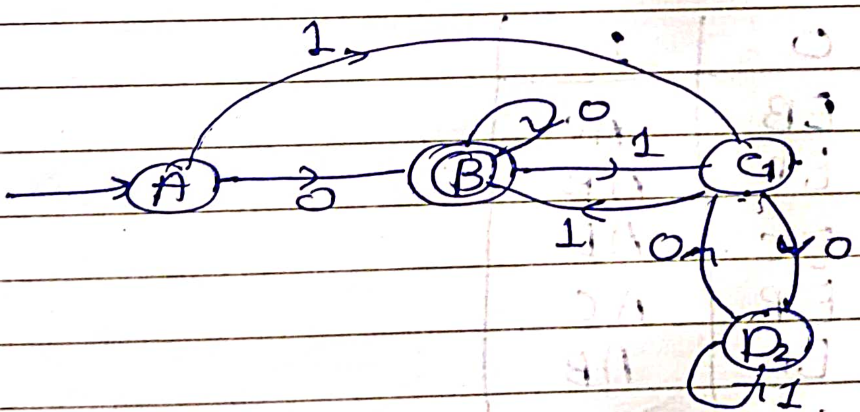
$\underbrace{11}_{3}0 = 6 \rightarrow$ Rem 0

111 = 7 → Rem 1.

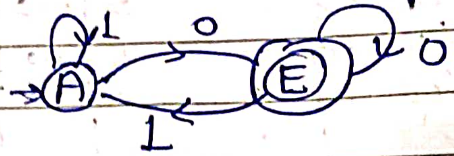
1110 = 14 → Rem 2

1111 = 15 → Rem 0

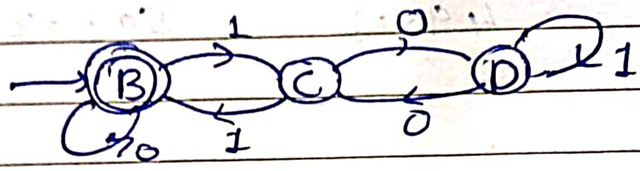
when ~~0 is~~ 0's appended at LSB, magnitude gets doubled, if 1 gets appended Magnitude ~~is~~ is double+1.



Q) Divisible either by 2 or 3.

Div. by 2 ⇒ 

$(Q_1, \Sigma_1, \delta_1, q_{01}, f_1)$

Div. by 3 ⇒ 

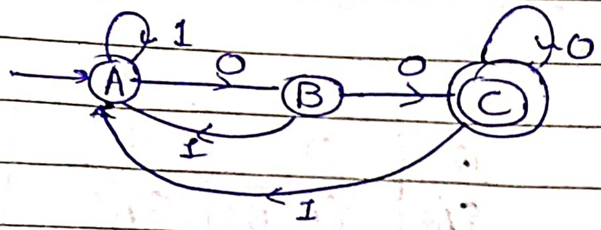
$(Q_2, \Sigma_2, \delta_2, q_{02}, f_2)$

$Q = \{AB, AC, AD, EB, EC, ED\}$
$\Sigma = \{0, 1\}$
$q_0 = AB$
$F = \{AB, EB, EC, ED\}$ ⇒ Any pair w/ either of the final states will be final.

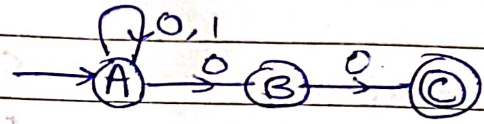
Q \ Σ	0	1
AB	EB	AC
AC	ED	AB
AD	EC	AD
EB	EB	AC
EC	ED	AB
ED	EC	AD

NFA/NDFA – one input symbol can have multiple paths.

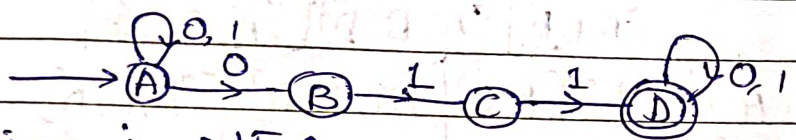
DFA Ending w/ 00.



NFA Ending w/ 00. $\rightarrow$



Eg:



★ Simulation in NFA:

I/P = 01100.

$\bar{\delta}(A,01100) \longrightarrow \bar{\delta}(A,1100) \longrightarrow \bar{\delta}(A,100), \bar{\delta}(A,00), \longrightarrow \delta(B,0) \longrightarrow \times$

$\delta(A,0) \rightarrow A, B$

$\bar{\delta}(A,01100) \longrightarrow \bar{\delta}(B,1100) \longrightarrow \bar{\delta}(C,100) \longrightarrow \bar{\delta}(D,00), \longrightarrow \delta(D,0) \longrightarrow D$

Transition Table:

$Q \backslash \Sigma$	0	1
A	{A, B}	{A}
B	$\emptyset$	{C}
C	$\emptyset$	{D}
D	{D}	{D}

$\bar{\delta} \Rightarrow$ Represented for sequence of simulation

$\delta \Rightarrow$ The last transition will have just δ.

{ } $\Rightarrow$ To maintain the uniformity, we put brackets to all values.

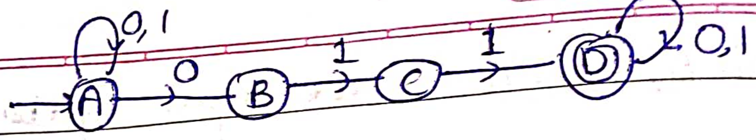
$$\delta : Q \times \Sigma \longrightarrow 2^{Q}$$

2^{Q} $\rightarrow$ power set

★NFA is not possible for divisibility or numerical types.

Page No.

Date



★ NFA ⟶ DFA

NFA: F1 $(Q, \Sigma, \delta, q_0, F) \Longrightarrow$ DFA: F2 $(Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$

$Q_1 \subset 2^Q$

$\Sigma_1 = \Sigma$

$q_{01} = q_0$

F_1

$\delta_1 = Q \times \Sigma \to 2^Q$.

$Q \backslash \Sigma$	0	1
{A}	{A, B}	{A}
{A,B}	{A, B}	{A, C}
{A, C}	{A,B}	{A, D}
{A,D}	{A, B, D}	{A, D}
{A,B,D}	{A, B, D}	{A, C, D}
{A, C, D}	{A,B,D}	{A, D}

$\delta_1(\{A,B\}, 0) = \delta(A,0) \cup \delta(B,0)$

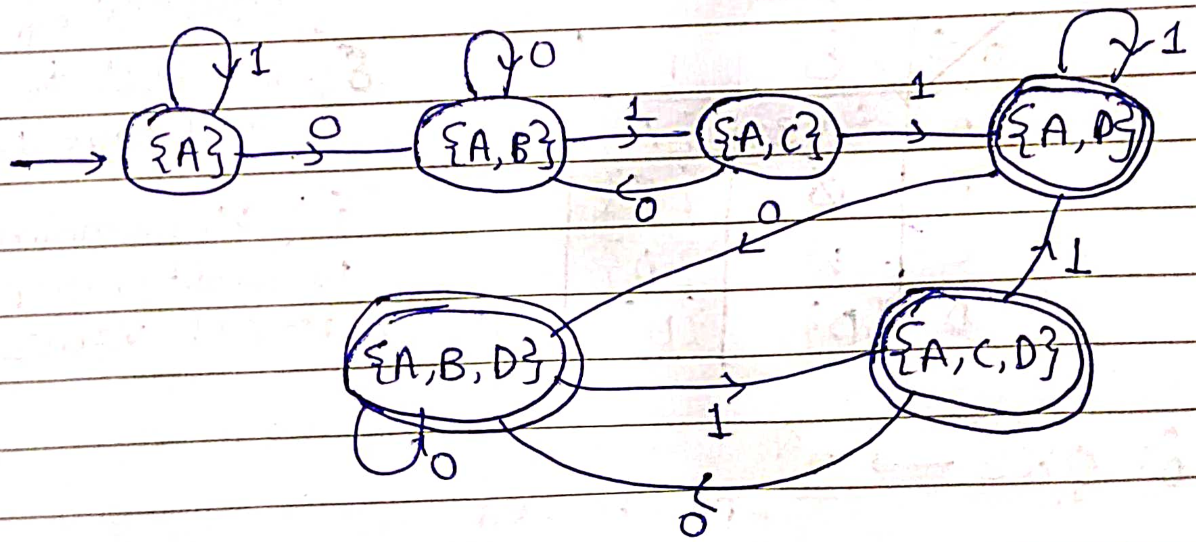
$= \{A, B\} \cup \emptyset$

$= \{A, B\}$.

$\therefore Q_1 = \{ \{A\}, \{A,B\}, \{A,C\}, \{A,D\}, \{A,B,D\}, \{A,C,D\} \}$.

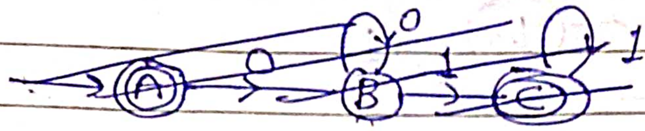
$F_1 = \{ \{A,D\}, \{A,B,D\}, \{A,C,D\} \}$

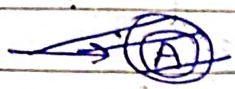
↳ Any set that has the final state of NFA will be Final state in DFA.



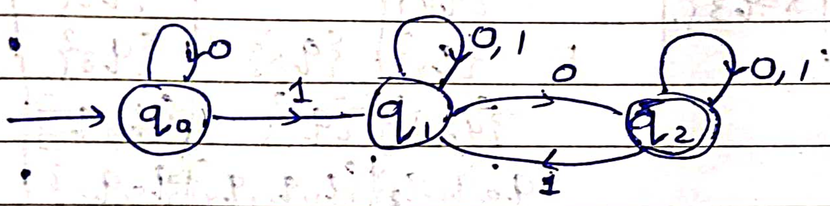
Q) Construct Automata for n no. of 0's followed by n no. of 1's.

→





Q) Convert NFA to DFA.

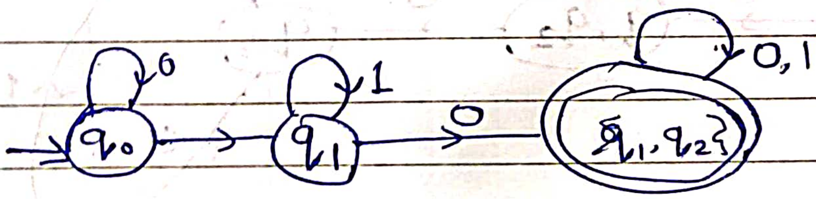


→

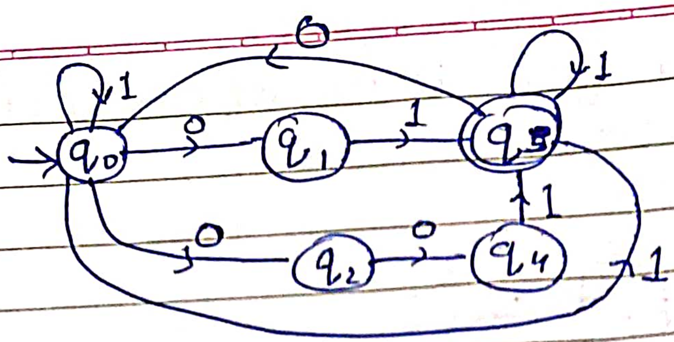
$Q \backslash \Sigma$	0	1
q_0	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_1, q_2\}$

→

$Q \backslash \Sigma$	0	1
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_1, q_2\}$	$\{q_1\}$
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_2\}$

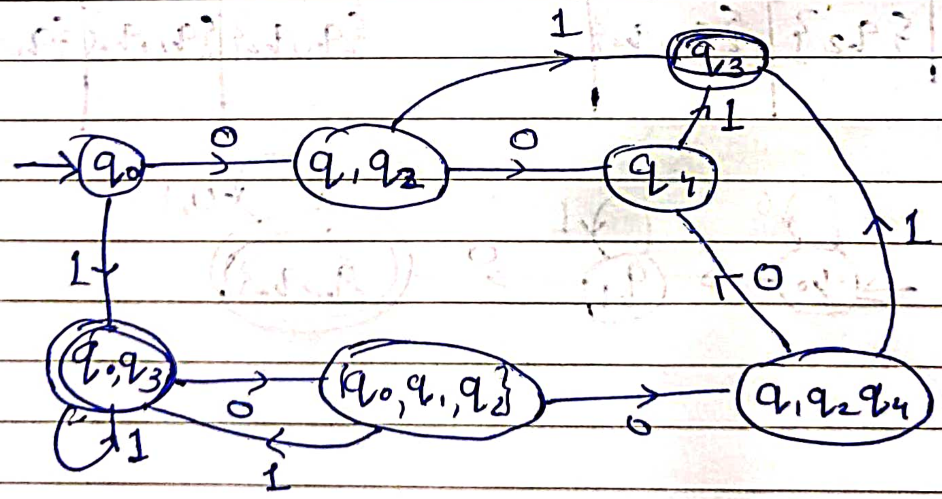


Q)



$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_0, q_3\}$
q_1	ϕ –	$\{q_3\}$
q_2	$\{q_4\}$	–
q_3	$\{q_0\}$	$\{q_3\}$
q_4	ϕ –	$\{q_3\}$

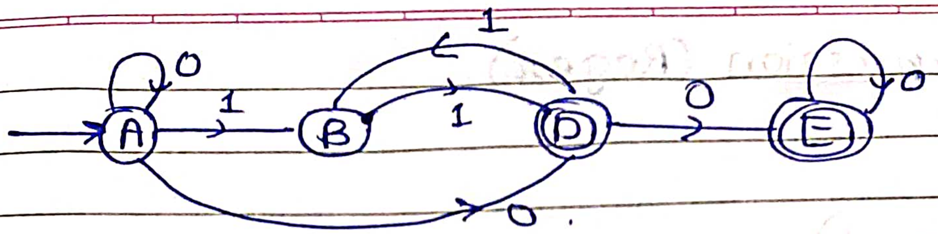
	$Q \backslash \Sigma$	0	1
	~~$\{q_1, q_2\}$~~	~~$\{q_4\}$~~	ϕ
$\Rightarrow$	$\{q_0\}$	$\{q_1, q_2\}$	$\{q_0, q_3\}$
	$\{q_1, q_2\}$	$\{q_4\}$	$\{q_3\}$
F	$\{q_0, q_3\}$	$\{q_1, q_2\}$	$\{q_0, q_3\}$
	$\{q_4\}$	–	$\{q_3\}$
F	$\{q_3\}$	$\{q_0\}$	$\{q_3\}$
	~~$\{q_0, q_1\}$~~	~~$\{q_1\}$~~	~~$\{q_0, q_3\}$~~
	$\{q_0, q_1, q_2\}$	$\{q_1, q_2, q_4\}$	$\{q_0, q_3\}$
	$\{q_1, q_2, q_4\}$	$\{q_4\}$	$\{q_3\}$
	~~$\{q_0, q_4\}$~~	~~$\{q_1\}$~~	~~$\{q_0, q_3\}$~~
	~~$\{q_1\}$~~	–	$\{q_3\}$



These are Types of NFA W/O Epsilon. (ε)

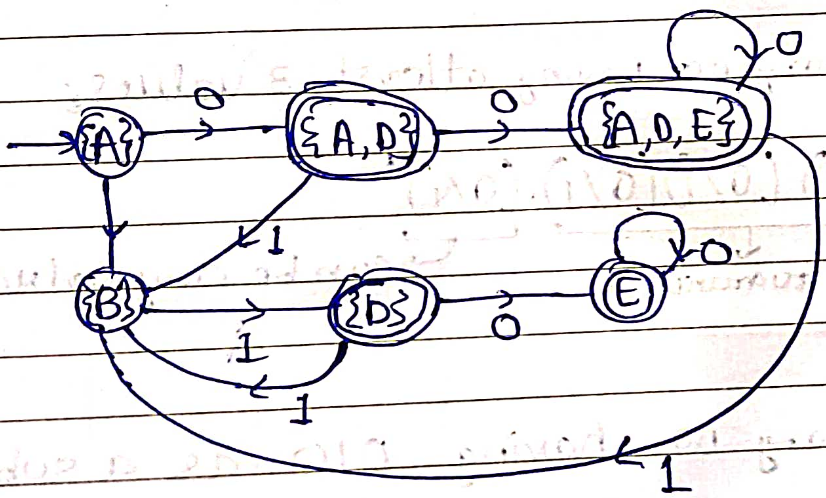
Page No.
Date

Q)



Q \ Σ	0	1
→ A	{A, D}	{B}
B	{} –	{D}
D	{E}	{B}
E	{E}	–

Q \ Σ	0	1
→ {A}	{A, D}	{B}
F{A, D}	{A, D, E}	{B}
{B}	{} –	{D}
F{A, D, E}	{A, D, E}	{B}
F{D}	{E}	{B}
F{E}	{E}	–



* **Regular Expression (Regex)**

Operators:

① OR (/ or +)

② Concatenation. (.)

③ Closure → For repetition.

Closure: Positive(+) / Kleen (*)

$(R_1)^* \rightarrow R_1^n \mid n \geq 0$

$(R_1)^+ \rightarrow R_1^n \mid n \geq 1$

$R_1^+ = R_1 . R_1^*$ → R_1^* can be repeated any no. of times & can be null as well.

Q) Regex for Binary no. having atleast 3 values:

→ $\underbrace{(0/1)(0/1)(0/1)}_{\text{Guaranteed}} . \underbrace{(0/1)^*}_{\text{can be any value}}$

Q) Regex for Binary no. having 010 as a substring.

→ $(0/1)^*\,010\,(0/1)^*$

Q) Regex where no 0's will be followed by 1 or no ~~on~~ 1's will be followed by 0.

→ $0^*/1^*$

~~1^*011~~
1^*010

Q) Regex never ending w/ 11.

→ $(0/1)^*(01/10/00) / (0/1/\epsilon)$

Q) Regex for Bin. lang. having no 2 cons. 0's & no 2 cons. 1's.

→ ~~(0^*1^*)~~ $(0.1)^* / (1.0)^* / (0/1/\epsilon) / (0.1)^*0 / (10)^*1$

OR

$(01)^*(\epsilon/0) / (10)^*(\epsilon/1)$

Q) Regex for generating integers:

→ $(\epsilon/+/-)(0/1/2/3/4/5/6/7/8/9)(0/1/2/3/4/5/6/7/8/9)^*$

or

$(\epsilon/-)\underbrace{(0/1/2/3/4/5/6/7/8/9)^+}$ → Null can't be the outcome.

Q) Regex for Bin. lang. having atmost 3 cosecutive 0's.

~~$(\epsilon/0/1)^* / (0/1)(0/1)$~~

$1^*/\epsilon/(0/1)/(0/1)(0/1)(0/1)/1^*$

~~$\epsilon/(0/1)^*/(0/1)1$~~

Atleast 3 cons 0's:

$(0/1)^*000(0/1)^*$

Q) Even no. of 0's followed by odd no. of 1's.

$(00)^*1(11)^*$

$(00)^*(11)^*1$

No two cons. 0's.

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

Q) Regex for binary no. where the no. will always be odd.

$$(01)^*1^*$$

Q) $r = \cancel{\epsilon+a+}\,a(a+b)^*$

$s = aa^* \cdot b$

$t = a^*b$

Language of r = a followed by any $comb^n$ of a & b.

Language of s = one or more a followed by b.

Language of t = any no. of a followed by b.

s ⇒ subset of t, $L(s) \subset L(t)$

$L(s) \subset L(r)$

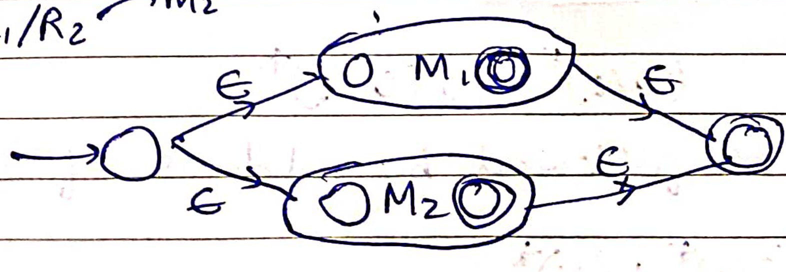
* <u>NFA w/ ε moves:</u>

Regex Representa^n ⇌ F.A. Representa^n

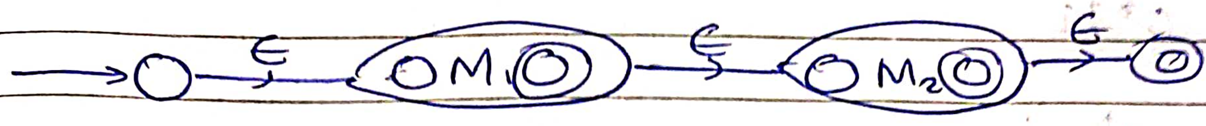
Operators in Regex: OR, concatena^n, closure.

• <u>Implementa^n of OR operator in NFA w/ ε moves:</u>

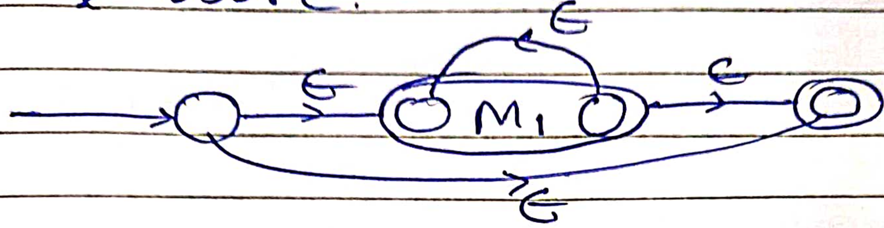
OR ⇒ $M_1 \leftarrow R_1/R_2 \rightarrow M_2$



$R_1 \cdot R_2 \Rightarrow$ Concatenation.



$R_1^* \Rightarrow$ Closure.



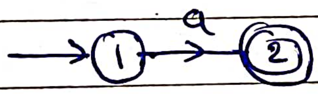
Q) NFA w/ ε for $a^*(b/bb)$.

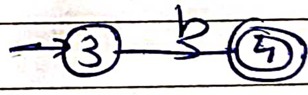
→ $R = a^*(b/bb)$

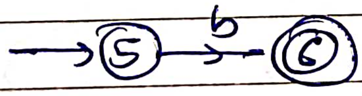
$R_1 = a^*$, $R_2 = b$, $R_3 = bb$.

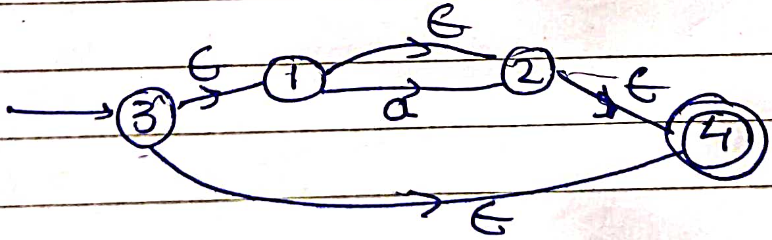
$\therefore R = R_1(R_2/R_3)$, $R_3 = R_5 \cdot R_6$.

$R_1 = R_4^*$, $R_4 = a$

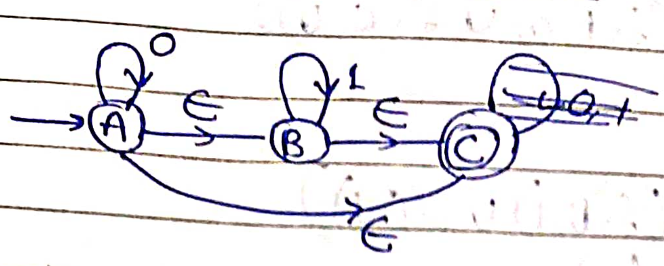
$R_4 =$ 

$R_5 =$ 

& $R_6 =$ 

$R_1 \Rightarrow$ 

Q) How to convert NFA w/ ϵ to NFA w/o ϵ.



Q \ Σ	0	1	ϵ
A	{A}	ϕ	{B, C}
B	ϕ	{B}	{C}
C	ϕ	ϕ	ϕ

* Epsilon Closure :

1) Epsilon closure of state, includes the state itself.
2) It also includes all the states that are directly or indirectly connected to that state through the ϵ.

$\therefore$ ϵ-closure(A) = {A, B, C}
ϵ-closure (B) = {B, C}
ϵ closure (C) = {C}

NFA w/ ϵ $\longrightarrow$ NFA w/o ϵ

$M_1\ (Q, \Sigma, \delta, q_0, F)$ $\quad$ $(Q, \Sigma, \underline{\delta_1}, q_0, F)$

$\delta_1(A,0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(A), 0))$
$= \epsilon\text{-closure}(\delta(A,B,C), 0))$
$= \epsilon\text{-closure}(\delta(A,0) \cup \delta(B,0) \cup \delta(C,0))$
$= \epsilon\text{-closure}(A \cup \phi \cup \phi)$
$= \epsilon\text{-closure}(A) = \{A, B, C\}$

$\delta_1(A,1) = \epsilon\text{-closure}(\delta(A,1) \cup \delta(B,1) \cup \delta(C,1))$
$= \epsilon\text{-closure}(\phi \cup B \cup \phi)$
$= \{\epsilon\text{-closure}(B) = \{B, C\}$

$\delta_1(B,0) = \cancel{\epsilon}\ \epsilon\text{-closure}(\cancel{\delta(A,0)} \cup (\delta(B,C),0))$
$= \epsilon\text{-closure}(\delta(B,0) \cup \delta(C,0))$
$= \cancel{\epsilon}\ \phi$

$\delta_1(B,1) = \epsilon\text{-closure}(\delta(B,1) \cup \delta(C,1))$
$= \epsilon\text{-closure}(B)$
$= \{B,C\}.$

$\delta_1(C,0) = \epsilon\text{-closure}(\delta(C,0))$
$= \epsilon\text{-closure}(C)$
$= \phi$

$\delta_1(C,1) = \phi.$

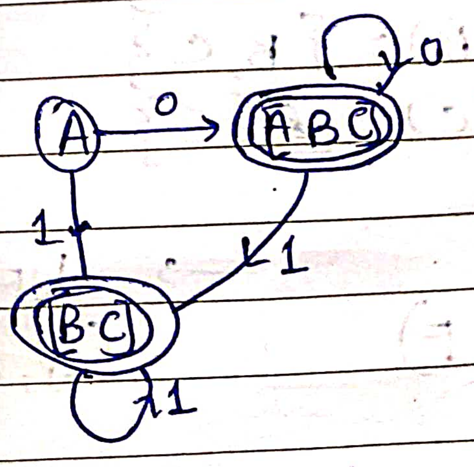
Q \ Σ	0	1
A	{A,B,C}	{B,C}
B	φ	{B,C}
C	φ	φ

NFA w/o ε

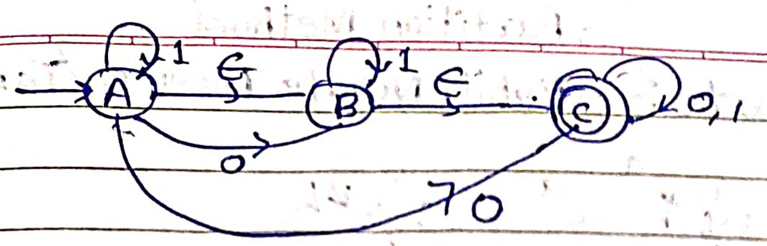
to DFA ⇒

Q \ Σ	0	1
A	{A,B,C}	{B,C}
{A,B,C}	{A,B,C}	{B,C}.
{B,C}	φ	{B,C}.

DFA.



Page No.
Date

Q) 

NFA w Epsilon:

$Q \backslash \Sigma$	0	1	ϵ
A	{B, C}	{A}	{B}
B	ϕ	{B}	{C}
C	{C}	{C}	0

ϵ-closure (A) = {A, B, C}
ϵ-closure (B) = {B, C}
ϵ-closure (C) = {C}

$\delta_1(A, 0) = \epsilon\text{-}c(\delta(A,0) \cup \delta(B,0) \cup \delta(C,0))$
$= \epsilon\text{-closure}((B,C) \cup \phi)$
$= \{B, C\}$

$= \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(A), 1))$
$\delta((A,B,C), 1)$
$\delta(A,1) \cup \delta(B,1) \cup (C,1)$
$\epsilon\text{-closure}\ \{A, B, C\}$
$\{A,B,C\} \cup \{B,C\} \cup \{C\}$

$\delta_1(A, 1) = \epsilon\text{-}c(\delta(A,1) \cup \delta(B,1))$
$= \epsilon\text{-}c(A \cup B)$
$= \{A, B, C\}$.

$\delta_1(B, 0) = \epsilon\text{-}c(\delta(B,0) \cup \delta(C,0))$
$= \epsilon\text{-}c(\phi \cup C)$
$= \epsilon\text{-}c(C) = \{C\}$

$\delta_1(B, 1) = \epsilon\text{-}c(\delta(B,1) \cup \delta(C,1))$
$= \epsilon\text{-}c(B \cup C)$
$= \{B, C\}$

$\delta_1(C, 0) = \epsilon\text{-}c(\delta(C,0))$
$= \epsilon\text{-}c(C) = \{C\}$

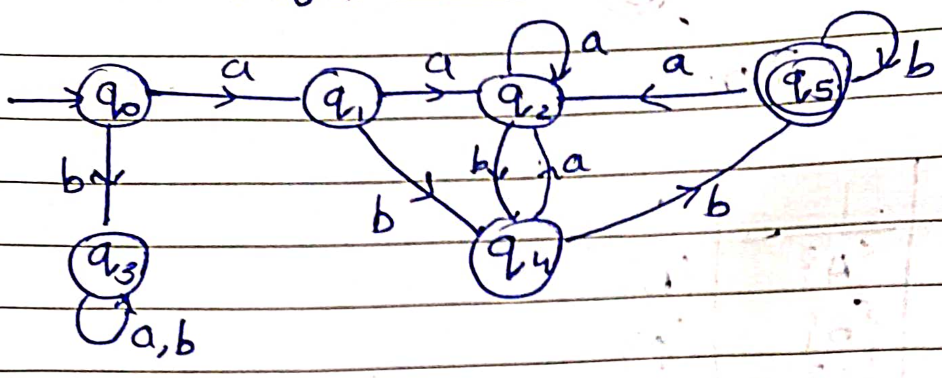
$\delta_1(C, 1) = \epsilon\text{-}c(\delta(C,1)) = \epsilon\text{-}c(C) = \{C\}$.

$Q \backslash \Sigma$	0	1
A	{B, C}	{A, B, C}
B	{C}	{B, C}
C	{C}	{C}

Page No.
Date

2 Methods:
→ Partition Method
→ Myhill Nerode Theorem / Tabular Method.

* Minimization of Automata



I) Partition Method

Partitions: (Non-final, final.)

denoted by $\pi_0 = \{ \overline{q_0 q_1 q_2 q_3 q_4}, \overline{q_5} \}$ → ~~The groups in~~ the partitions are not equivalent w/ each other.

q_3 → Dead.

Steps for minimization:

① Remove all Dead states.
② Apply method

States belonging to same partition, may or may not be equivalent.

~~π_0~~

$\pi_0 = \{ \overline{q_0 q_1 q_2 q_4}, \overline{q_5} \}$

↓

~~$\pi_1 = \{$~~ Take pairs & check equivalence.

$q_0 \xrightarrow{a} q_1$ $q_0 \xrightarrow{b} -$
$q_1 \xrightarrow{a} q_2$ $q_1 \xrightarrow{b} q_4$ × ∴ Not Equivalent.

same partition criteria

$q_0 \xrightarrow{a} q_1$ $q_0 \xrightarrow{b} \cancel{q_4}$
$q_2 \xrightarrow{a} q_2$ $q_2 \xrightarrow{b} q_4$ × ∴ Not Equivalent.

$q_1 \xrightarrow{a} q_2$ $q_1 \xrightarrow{b} q_4$
$q_2 \xrightarrow{a} q_2$ $q_2 \xrightarrow{b} q_4$ ~~∴ Equivalent~~ ∴ It maybe Equivalent

↳ check whether ~~the el~~ both elements in the pair belong to same partition.

~~$\pi_1 = \{ q_0$~~

$q_0 \xrightarrow{a} q_1$ $\quad$ $q_0 \xrightarrow{b} -$ $\times$

$q_4 \xrightarrow{a} q_2$ $\quad$ $q_4 \xrightarrow{b} q_5$

$\therefore q_0 \not\equiv q_4.$

We can check equivalence of q_4 w/ either q_1 or q_2 as they are equivalent for now.

(choose either q_1, q_4 or q_2, q_4 as $q_1 \cong q_2$ equivalent)

$\therefore q_2 \xrightarrow{a} q_2$ $\quad$ $q_2 \xrightarrow{b} q_4$ $\times$

$q_4 \xrightarrow{a} q_2$ $\quad$ $q_4 \xrightarrow{b} q_5$

Nonfinal & final state can't form a pair.

$\therefore q_2 \not\equiv q_4.$

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

$\pi_2 = \{ \overline{q_0}, \overline{q_1 q_2}, \overline{q_4}, \overline{q_5} \}$ $\because q_1, q_2$ have pairs of same ~~elements~~ states.

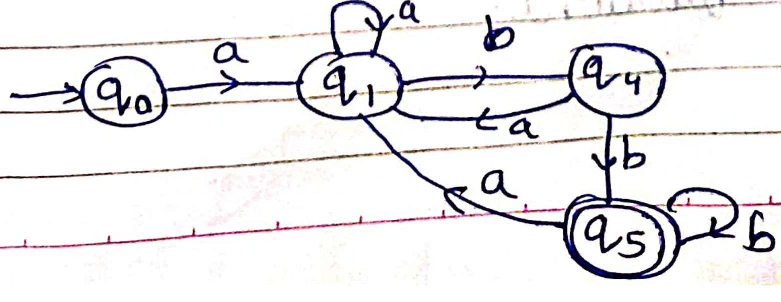
$\therefore q_1 \equiv q_2.$

$\pi_1 = \pi_2.$

$\because q_1 \equiv q_2$, we can eliminate either of these states in the minimized Automata.

Q \ Σ	a	b
q_0	q_1	ϕ
q_1	q_1	q_4
q_4	q_1	q_5
q_5	q_1	q_5

⇒ Minimized Automata after applying Partition method.



Page No.

Date

II) Tabular Method (Box Method):

The table represents all possible combinaⁿ of the pair of the states.

X ⇒ Pair can never be equivalent.

Pair of same state will always be uncrossed.

From 2nd to nth state

q_1	X			
q_2	X	=		
q_4	X	X	X	
q_5	X	X	X	X
	q_0	q_1	q_2	q_4

From start state to $(n-1)^{th}$ state.

∵ Final can never be ≡ to non-final states.

final & nonfinal pairs get crossed ie $q_5q_0, q_5q_1, q_5q_2, q_5q_4$

$q_0 \xrightarrow{a} q_1$ $\quad$ $q_0 \xrightarrow{b} \phi$ X

$q_1 \xrightarrow{a} q_2$ $\quad$ $q_1 \xrightarrow{b} q_4$

↳ check if the box is crossed in the table.

$q_0 \xrightarrow{b} \phi$ X $\quad$ $q_0 \xrightarrow{b} \phi$ X

$q_2 \xrightarrow{b} q_4$ $\quad$ $q_4 \xrightarrow{b} q_5$

$q_1 \xrightarrow{a} q_2$ $\quad$ $q_1 \xrightarrow{b} q_4$ may be

$q_2 \xrightarrow{a} q_2$ $\quad$ $q_2 \xrightarrow{b} q_4$

$q_1 \xrightarrow{a} q_2$ $\quad$ $q_1 \rightarrow q_4$ X

$q_4 \xrightarrow{a} q_2$ $\quad$ $q_4 \rightarrow q_5$

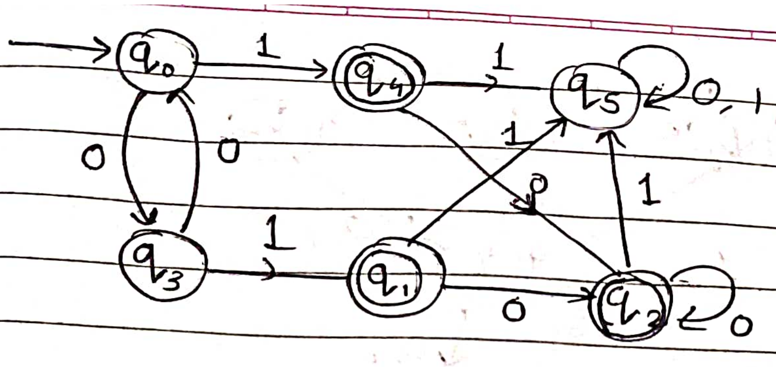
$q_2 \xrightarrow{a} q_2$ $\quad$ $q_2 \xrightarrow{b} q_4$ X

$q_4 \xrightarrow{a} q_2$ $\quad$ $q_4 \xrightarrow{b} q_5$

Now check again for the uncrossed box.

Since the pairs are of same state, they will be equivalent.

Q)



$Q \backslash \Sigma$	0	1
q_0	q_3	q_4
q_1	q_2	ϕ
q_2	q_2	ϕ
q_3	q_0	q_1
q_4	q_2	ϕ.

→ ① $q_5 \Rightarrow$ Dead state. ✗

② $\pi_0 = \{ \overline{q_0 q_3} , \overline{q_1 q_2 q_4} \}$

$q_0 \xrightarrow{0} q_3$ ✓ $\quad q_0 \xrightarrow{1} q_4$ ✓
$q_3 \xrightarrow{0} q_0$ $\quad q_3 \xrightarrow{1} q_1$

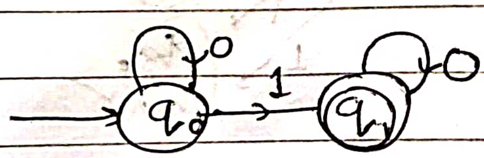
∴ They may be equivalent.

$q_1 \xrightarrow{0} q_2$ ✓ $\quad q_1 \xrightarrow{0} q_2$ ✓
$q_4 \xrightarrow{0} q_2$ $\quad q_2 \xrightarrow{0} q_2$
$q_1 \xrightarrow{1} \phi$ ✓ $\quad q_1 \xrightarrow{1} \phi$ ✓
$q_4 \xrightarrow{1} \phi$ $\quad q_2 \xrightarrow{1} \phi$

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

$\therefore \pi_0 = \pi_1 \Rightarrow q_0 \equiv q_3$ & $q_1 \equiv q_2 \equiv q_4$.

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_1	ϕ

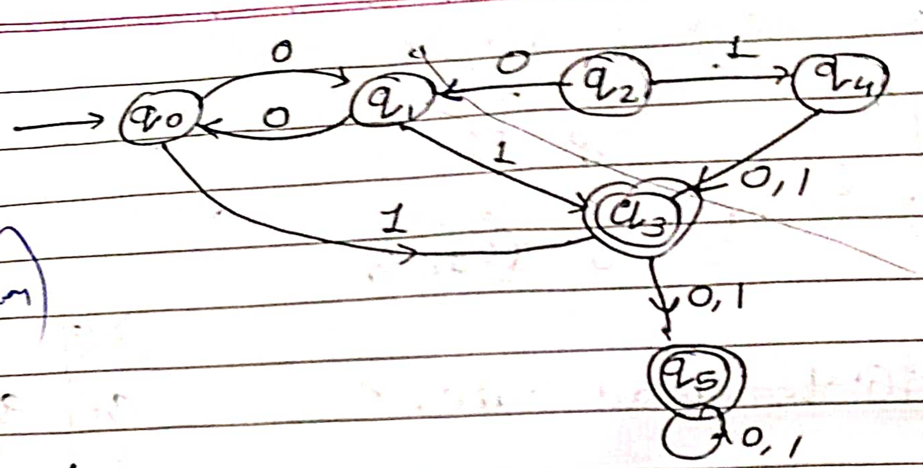


$q_2 \Rightarrow$ Unreachable ... to reach to that state ... initial state.

Page No.

Date

Q)



(if q_2 eliminate then q_2 & q_4 will go away)

→ $q_2 \Rightarrow$ Unreachable.

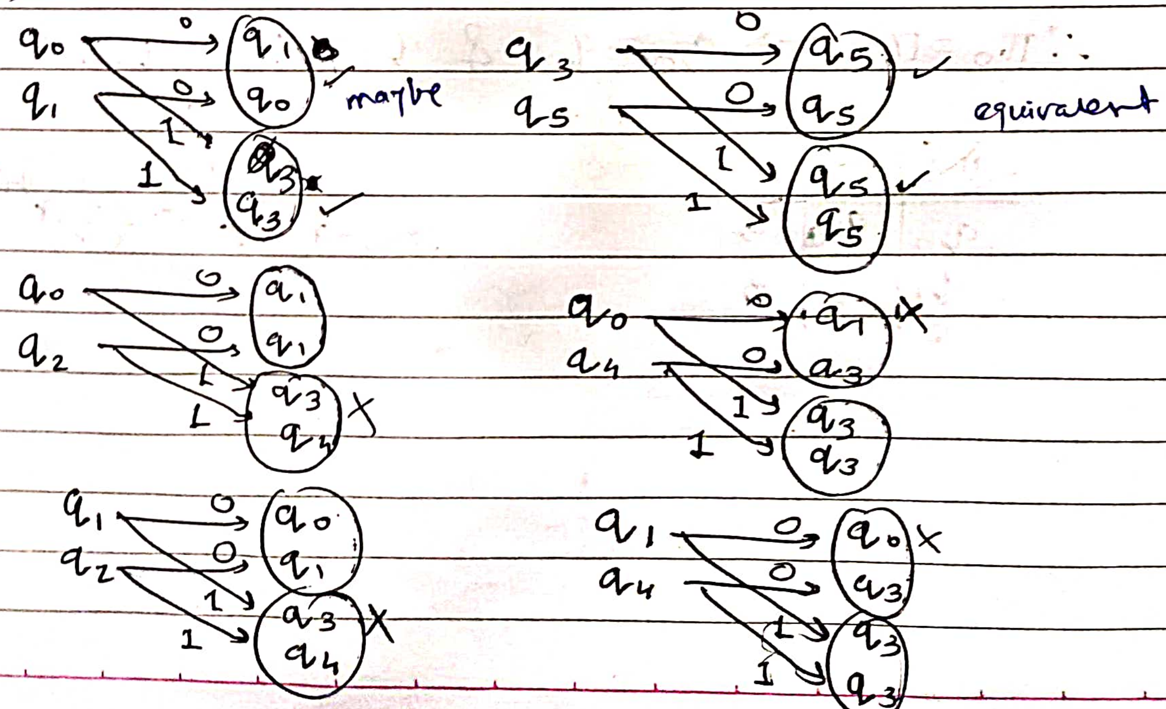
$\therefore q_4$ becomes unreachable.

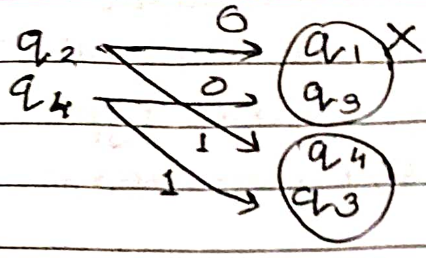
* Solving w/o eliminating.

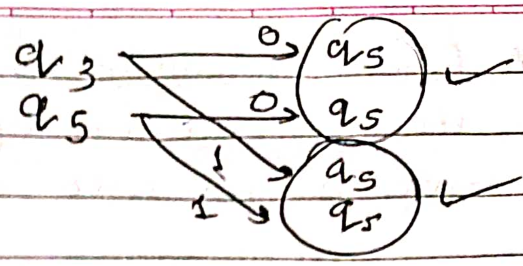
$Q \backslash \Sigma$	0	1
q_0	q_1	q_3
q_1	q_0	q_3
q_2	q_1	q_4
q_3	q_5	q_5
q_4	q_3	q_3
q_5	q_5	q_5

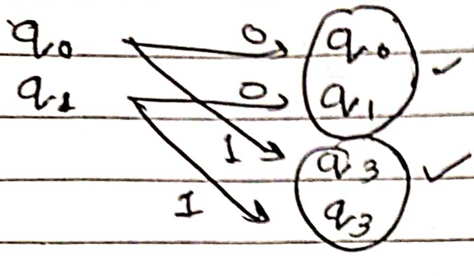
	q_0	q_1	q_2	q_3	q_4
q_1	≡				
q_2	X	X			
q_3	X	X	X		
q_4	X	X	X	X	
q_5	X	X	X	≡	X

(final & non-final states get crossed ie. q_5q_0, q_5q_1, q_5q_2, q_5q_4, q_3q_0, q_3q_1, q_3q_2 & only q_5q_3 work)



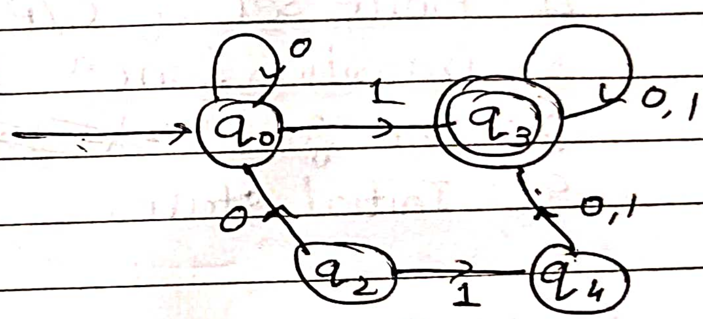






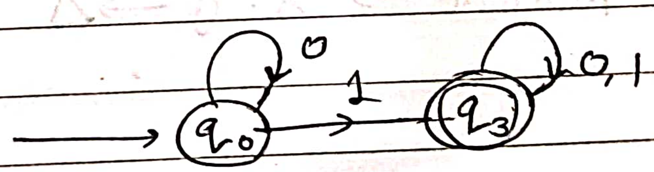
$\therefore\ q_0 \equiv q_1$ & $q_3 \equiv q_5$.

$Q \backslash \Sigma$	0	1
q_0	q_0	q_3
q_2	q_0	q_4
q_3	q_3	q_3
q_4	q_3	q_3



Here q_2, q_4 are unreachable.

$\therefore$



Page No. O/P is associated w/ the transi^n

Date

Mealy Machine → O/P is associated w/ the transi^n

Moore Machine → O/P is associated w/ state

★ Finite Automata w/ O/P → Mealy Machine, Moore Machine

There is no need of Final state in this type of Automata as we are generating O/P over here.

The tuple Definition is given by:

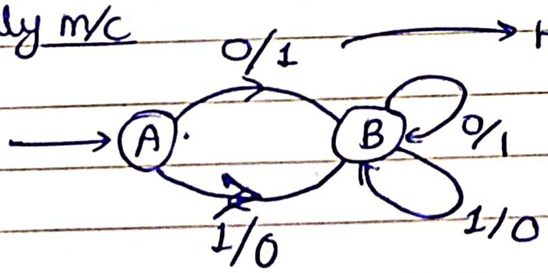
$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

Q: Finite set of states.
Σ: Finite set of I/P symbols.
Δ: Finite set of O/P symbols.
δ: Transition func^n, $\delta: Q \times \Sigma \rightarrow Q$.
λ: O/P func^n; ~~$\lambda: Q \times \Sigma \rightarrow \Delta$~~
q_0: Initial state.

λ Func^n:
① Mealy Machine ⇒ $\lambda: Q \times \Sigma \xrightarrow{\delta \to \Delta} \Delta$ ⇒ O/P func^n wrt. state & I/P symbol.
② Moore Machine ⇒ $\lambda: Q \rightarrow \Delta$

Q) Machine ~~for~~ to generate 1's complement.

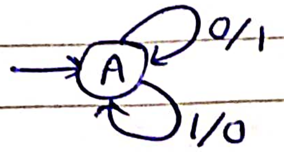
Mealy m/c



→ Here / doesn't denote "or", here it means that on the I/P of 0, we should get O/P as 1.

T.D ⇒ $Q = \{A, B\}$, $\Delta = \{0, 1\}$
$\Sigma = \{0, 1\}$, $q_0 = A$.

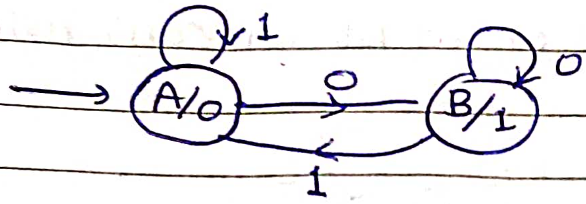
⇓



Q \ Σ	0: δ	0: λ	1: δ	1: λ
A	B	1	B	0
B	B	1	B	0

↳ Transi^n table & λ func^n combined.

Moore M/C → For n O/P symbols, the min. no. of states in m/c will be n.



$\delta \Rightarrow$

Q \ Σ	0	1
A	B	A
B	B	A

$\lambda \Rightarrow$

Q	Δ
A	0
B	1

Simulation (Mealy)

I/P = 0110

$\delta(A, 0110) \rightarrow \delta(B, 110) \rightarrow \delta(B, 10) \rightarrow \delta(B, 0) \rightarrow B$

O/P = $\lambda(A,0)$ $\lambda(B,1)$ $\lambda(B,1)$ $\lambda(B,0)$

1 0 0 1.

= 1001

Simulation (Moore)

I/P = 0110

$\delta(A, 0110) \rightarrow \delta(B, 110) \rightarrow \delta(A, 10) \rightarrow \delta(A, 0) \rightarrow B$

O/P = $\lambda(A)$ $\lambda(B)$ $\lambda(A)$ $\lambda(A)$ $\lambda(B)$

0 1 0 0 1

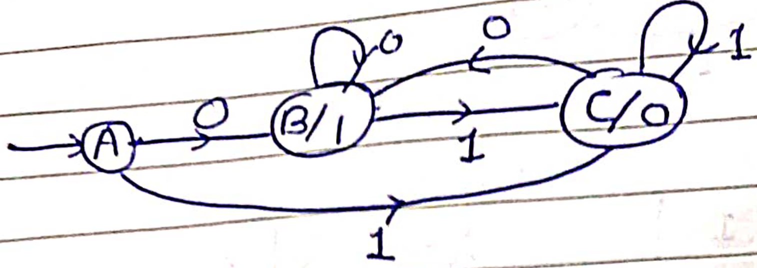
= 01001.

NOTE: In Moore m/c, we will get O/P of the length (n+1) for I/P (n).

0 1 0 . 32 6.4

~~If we want to~~

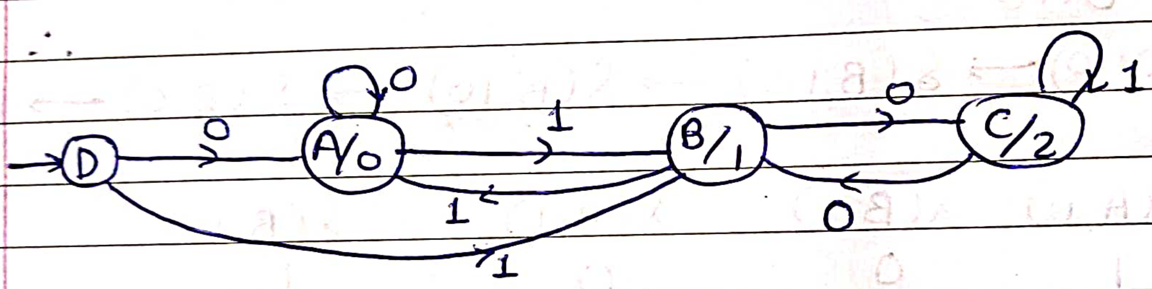
If we don't want the extra bit, we can construct the m/c as



So now here $\lambda(A)$ will be null (ϵ) & hence we will get same length of the O/P as the I/P.

Q) n mod 3 : Moore machine.

Here we will have 3 desired o/p : 0, 1, 2.



If the I/P is such that remainder is 0 Eg: 11, & we add '0' to its ~~Eg: 11~~ LSB, the number will be doubled

If the I/P has remainder 1, & we add 0 to it, the value of the resulting number is Double + 1.

$Q = \{A, B, C, D\}$
$\Sigma = \{0, 1\}$
$\Delta = \{0, 1, 2\}$
$q_0 = \{D\}$

$\delta \Rightarrow$

Q \ Σ	0	1
A	A	B
B	C	A
C	B	C
D	A	B

$\lambda \Rightarrow$

Q	Δ
A	0
B	1
C	2
D	ϵ

I/P = ~~1101~~
O/P = ~~1001~~

1 1 0 1
1 0 0 1

i.e for I/P 1, O/P is 1
I/P 11, O/P is 10
I/P 110, O/P is 100.
I/P 1101 O/P 1001

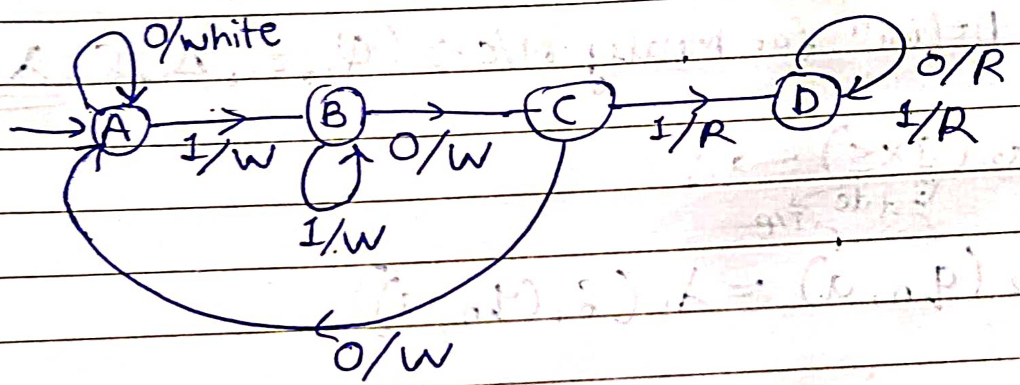
Page No.
Date

Q) Mealy M/c to generate E & O, If even no. then O/P 'E' & for odd no. O/P O.

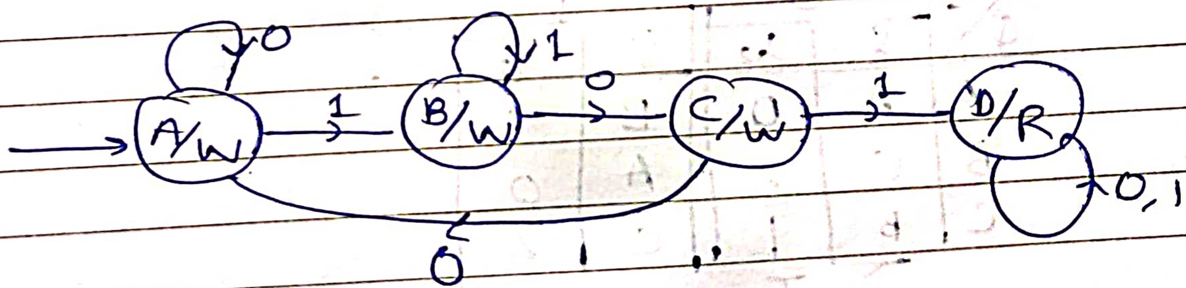
Q) Construct a mealy M/C on binary lang. where the O/P generated will be red, when the input contains the substring 101, otherwise O/P will be white.

→ $\Delta = \{Red, white\}$
$\Sigma = \{0, 1\}$.

Mealy M/C:



Moore M/C:

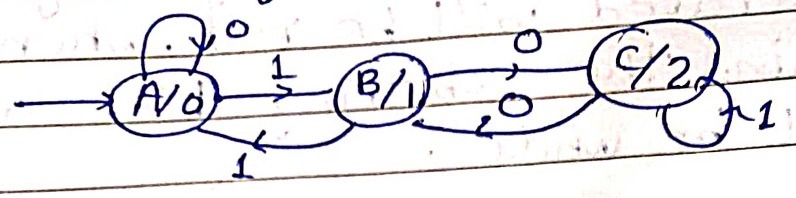


Tut: 5: ~~AH~~
RE → FA

Date

★ Converting Moore M/C to Mealy M/C.

Eg. ~~Moo~~ Moore m/c for n mod 3.



$Q \setminus \Sigma$	δ: 0	δ: 1	λ
A	A	B	0
B	C	A	1
C	B	C	2

Tuple Defini^n for Moore M/C ⇒ $(Q, \Sigma, \Delta, \delta, \lambda_1, q_0)$

⇓

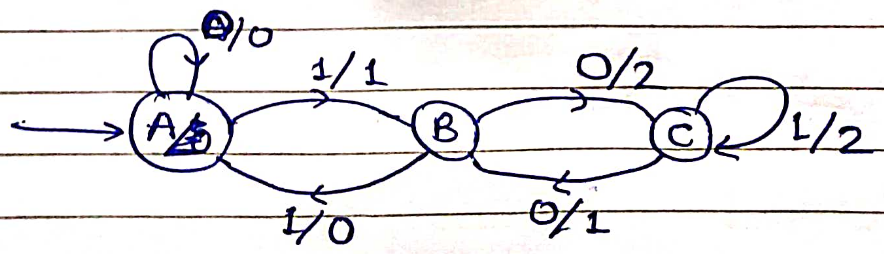
Tuple Defini^n for Mealy M/C ⇒ $(Q, \Sigma, \Delta, \delta, \underline{\underline{\lambda_2}}, q_0)$

$\lambda_2 (\underbrace{Q}_{\text{state}} \times \underbrace{\Sigma}_{I/P}) \rightarrow \Delta$

$\lambda_2 (q_n, a) = \lambda_1 (\delta, (q_n, a))$

$Q \setminus \Sigma$	0: δ	0: λ_2	1: δ	1: λ_2
A	A	0	B	1
B	C	2	A	0
C	B	1	C	2

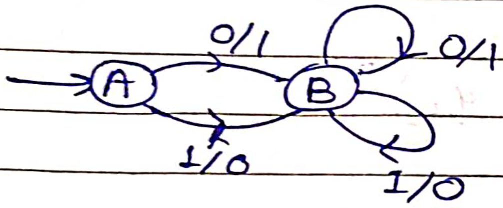
$\lambda(\delta)$



* Converting Mealy M/C to Moore M/C

Eg: Mealy M/C for 1's complement.

eg bracket & no comma inside



Q \ Σ	0: δ	0: λ	1: δ	1: λ
A	B	1	B	0
B	B	1	B	0

T.D. for Mealy Machine ⇒ $(Q, \Sigma\ \Delta\ \delta\ \lambda\ q_0)$

⇓

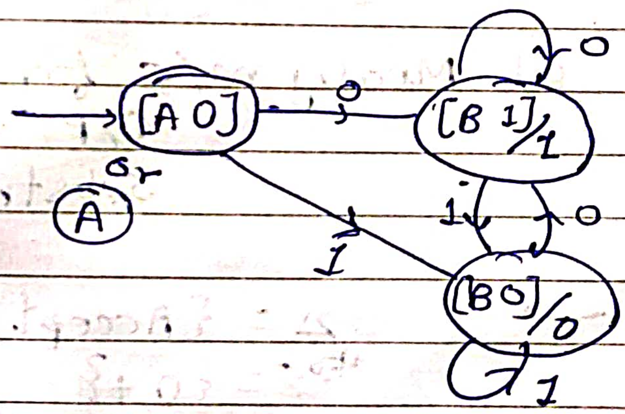
T.D. for Moore M/C ⇒ $(Q_1, \Sigma_1 \Delta_1 \delta_1 \lambda_1 q_{01})$

$Q_1 : [Q \times \Delta]$

$\therefore Q_1 = \{[A, 0], [A, 1], [B, 0], [B, 1]\}$

As $\lambda_1([Q, a]) = a$, $\delta_1([Q, a], 0) = [\delta(Q, 0)\ \lambda(Q, 0)]$

Q_1 \ Σ	δ_1: 0	δ_1: 1	λ_1
[A 0]	[B 1]	[B 0]	0
[A 1]	[B 1]	[B 0]	1
[B 0]	[B 1]	[B 0]	0
[B 1]	[B 1]	[B 0]	1



~~Since there is no output~~

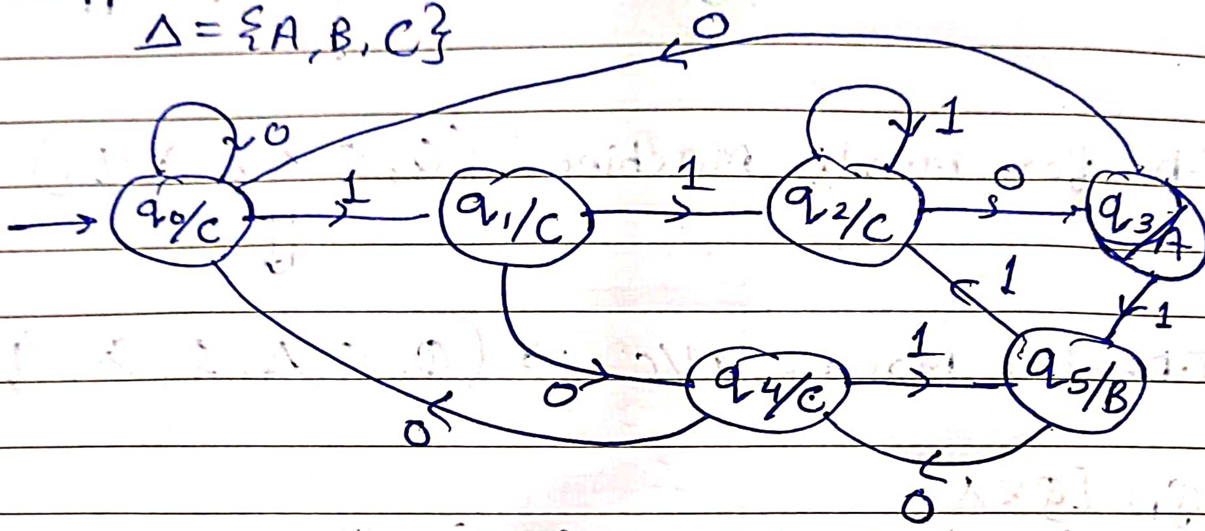
Since control will never go on [A 1], it is unreachable, There is no possibility of O/P at A.

∴ [A 0] will be initial state.

* We can also write [A0] as just A, because there is no combinaⁿ of A w/ anything.

Q) Generate ~~any~~ Moore m/c for:
O/P A when 110 appears.
B - 101
C - else.

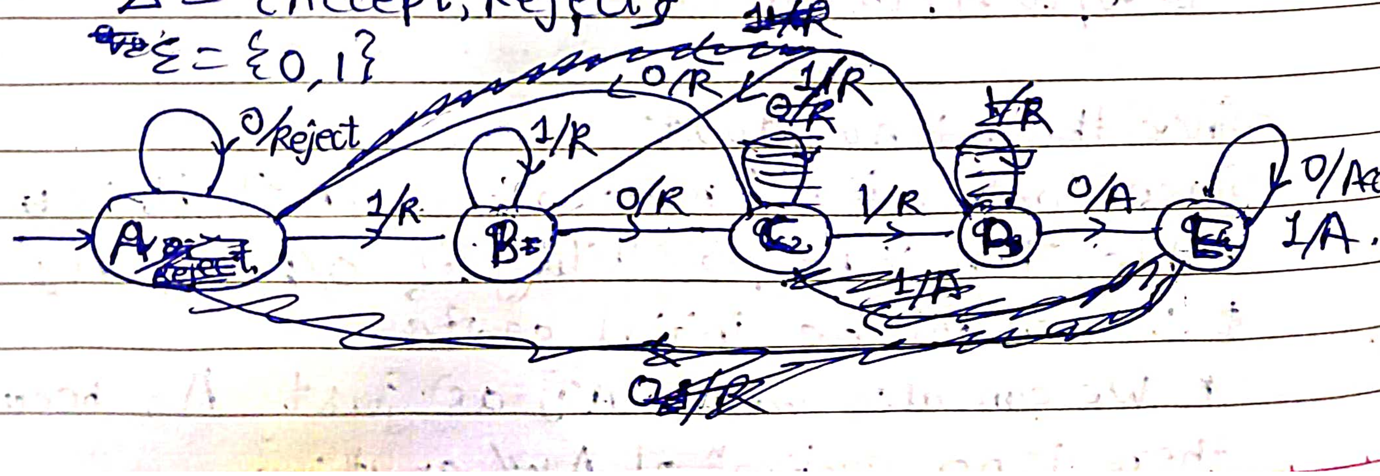
→ Moore M/C
T.D ⇒ (Q Σ Δ δ λ q_0)
$\Sigma = \{0,1\}$
$\Delta = \{A, B, C\}$



Mealy M/C: Just replace the O/P in the states to the transition.

Q) Mealy m/c for:
O/P "Accept" when I/P has substring 1010, else ~~"Not"~~ "Reject"

→ $\Delta = \{Accept, Reject\}$
$\Sigma = \{0,1\}$



converting to moore M/c:

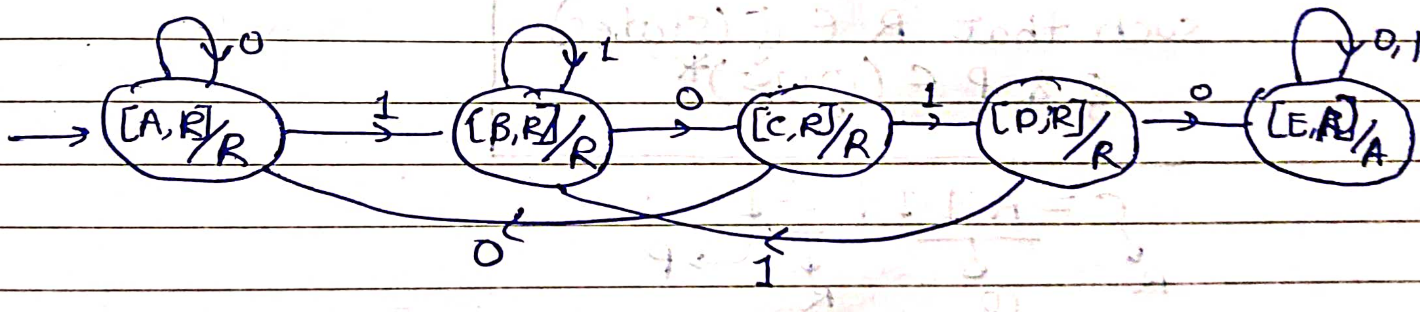
$Q_1 : [Q \times \Delta] = \{[A,A], [A,R], [B,A], [B,R], [C,A], [C,R], [D,A], [D,R], [E,A], [E,R]\}$. ⟶ Possible states.

Q \ Σ	δ: 0	δ: 1	λ
~~→[A,A]~~	[A,R]	[B,R]	A
→[A,R]	[A,R]	[B,R]	R
[B,R]	[C,R]	[B,R]	R
[C,R]	[A,R]	[D,R]	R
[D,R]	[E,A]	[B,R]	R
[E,A]	[E,A]	[E,A]	A

} Set of Actual States.

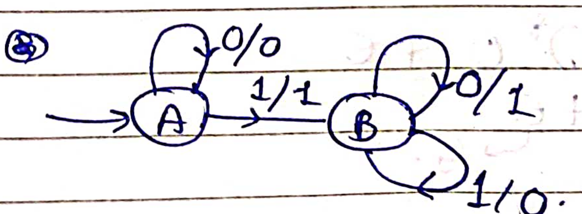
Since [A,A] is never coming in the transition table, [A,A] becomes unreachable, so we eliminate it.

Now the Initial state will be [A,R].



Q) Create a m/c to generate 2's complement of binary no.

Note : I/P is taken from Right → Left.

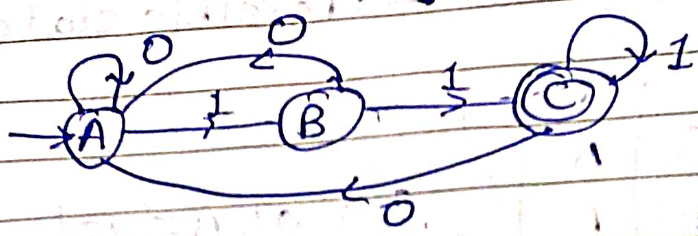


100100

011100

FA → RE

* Regex for a given Automata:



$A = A\cdot 0 + B\cdot 0 + C\cdot 0 + \epsilon$ — ① for initial state

$B = A\cdot 1$ — ②

$C = B\cdot 1 + C\cdot 1$ — ③

from ② & ③

$C = A\cdot 1\cdot 1 + C\cdot 1$ Find expression for final state

- **ARDEN'S THEOREM**

  $R = Q + RP \Rightarrow R = QP^*$

  such that $R \in Q$ (states)

  Q & $P \in (\Sigma \cup \epsilon)^*$

$C = \underbrace{A\cdot 1\cdot 1}_{Q} + \underbrace{C}_{R}\cdot \underbrace{1}_{P}$, $C \to R$

$\therefore C = A11(1)^*$ — ④

$A = A\cdot 0 + B\cdot 0 + C\cdot 0 + \epsilon$

$A = A\cdot 0 + A\cdot 1\cdot 0 + A11(1)^* 0 + \epsilon$

$\underbrace{A}_{R} = \underbrace{A}_{R}\underbrace{(0 + 10 + 111^* 0)}_{P} + \underbrace{\epsilon}_{Q}$

$\therefore$ By Arden's Theorem,

$A = \epsilon(0 + 10 + 111^* 0)^*$

Now, $\epsilon\cdot A = A\epsilon = A$.

$\therefore A = (0 + 10 + 111^* 0)^*$

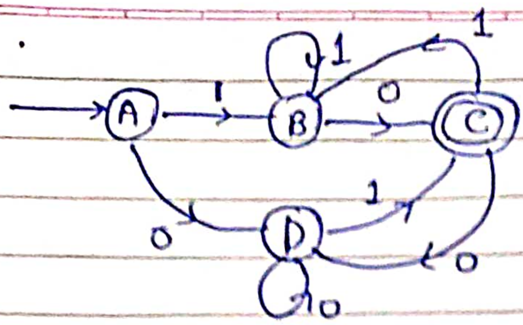
$\boxed{C = (0 + 10 + 111^* 0)^* 111^*}$

$R = Q + RP \Rightarrow R = QP^*$

$+ \Rightarrow$ OR operator (commutative)

$\cdot \Rightarrow$ Concatenation (Not commutative) $A \cdot B \neq B \cdot A$.

Q)



Incoming options (arrows)

$A = \epsilon$ — ①

$B = A \cdot 1 + B \cdot 1 + C \cdot 1 \cancel{= (A+B+C)1}$ — ②

$C = B \cdot 0 + D \cdot 1$ — ③

$D = A \cdot 0 + C \cdot 0 + D \cdot 0$ — ④

Replace A from ① in ②.

$$B = \epsilon \cdot 1 + B \cdot 1 + C \cdot 1.$$
$$= 1 + B \cdot 1 + C \cdot 1 \quad \text{— ⑤}$$

Replace A in ④

$$D = 0 + C \cdot 0 + D \cdot 0 \quad \text{— ⑥}$$

Applying Arden's Thm in ⑤

$$B = 1 + B \cdot 1 + C \cdot 1$$
$$B = \underbrace{1 + C \cdot 1}_{Q} + B \cdot \underbrace{1}_{P}$$

$\boxed{R = Q + RP}$ here $R = B$

$$\therefore B = (1 + C \cdot 1)1^*$$
$$= 11^* + C \cdot 11^* \quad \text{— ⑦}$$

Applying Arden's in ⑥.

$$D = \underbrace{0 + C \cdot 0}_{Q} + D \cdot \underbrace{0}_{P}$$

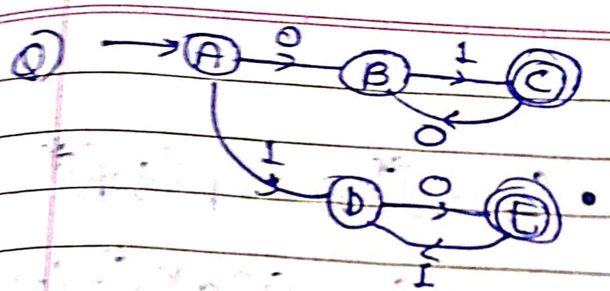
$$D = (0 + C \cdot 0)0^*.$$
$$= 00^* + C \cdot 00^* \quad \text{— ⑧}$$

Replace B & D from ⑦ & ⑧ in eq^n ③.

$$C = (11^* + C11^*) \cdot 0 + (00^* + C00^*) \cdot 1$$
$$= 11^*0 + C11^*0 + 00^*1 + C00^*1$$
$$C = \underbrace{11^*0 + 00^*1}_{Q} + C\underbrace{(11^*0 + 00^*1)}_{P}$$

$$\therefore C = (11^*0 + 00^*1)(11^*0 + 00^*1)^*$$

Q)



$A = \epsilon$ —(1)

$B = A \cdot 0 + C \cdot 0$ —(2)

$C = B \cdot 1$ —(3)

$D = A \cdot 1 + E \cdot 1$ —(4)

$E = D \cdot 0$ —(5)

~~Replace A from (1) w/ A in (2).~~

$\cancel{B = \epsilon \cdot 0 + C \cdot 0}$

$\cancel{= 0 + C \cdot 0}$ ~~—(6)~~

~~D ... (1) w/ A in (4)~~

$\cancel{D = \epsilon \cdot 1 + E \cdot 1}$

$\cancel{= 1 + E \cdot 1}$ ~~—(7)~~

~~Applying Arden's thm in (6).~~

$\cancel{B = 0 + C \cdot 0}$

Substitute (2) in (3).

$C = (A \cdot 0 + C \cdot 0) \cdot 1$ → Put $A = \epsilon$

$= A \cdot 01 + C \cdot 01$ —(6)

Substitute (4) in (5).

$E = (A \cdot 1 + E \cdot 1)\, 0$

$= A \cdot 10 + E \cdot 10$ —(7)

Applying Arden's thm in (6).

$C = \underbrace{A \cdot 01}_{Q} + C \cdot \underbrace{01}_{P}$

$C = (A \cdot 01)(0 \cdot 1)^*$ —(8)

Applying Arden's in (7).

$E = \underbrace{10}_{Q} + E \cdot \underbrace{10}_{P}$

$E = (10)(10)^*$ —(9)

Final language of automata: we'll combine the two regex.

$(01)(01)^* / (10)(10)^*$