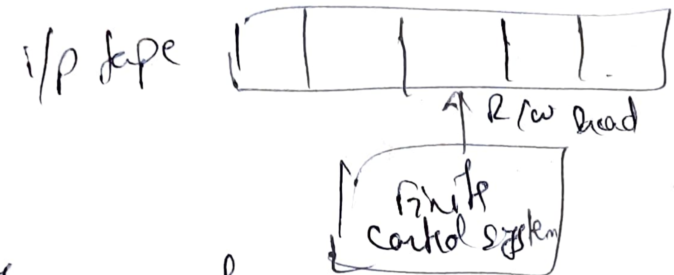
7/01/21

⇒ ATC :-

Module-1 :-

1) Theory Questions :- Def^n of alphabet, language & other basic def^n (concatenation, powers of alphabet)

2) FA structure :- (dig. & explain)



$\delta : q_0 \times \Sigma = q_0$

3) Theorem proof (NDFSM to DFSM)

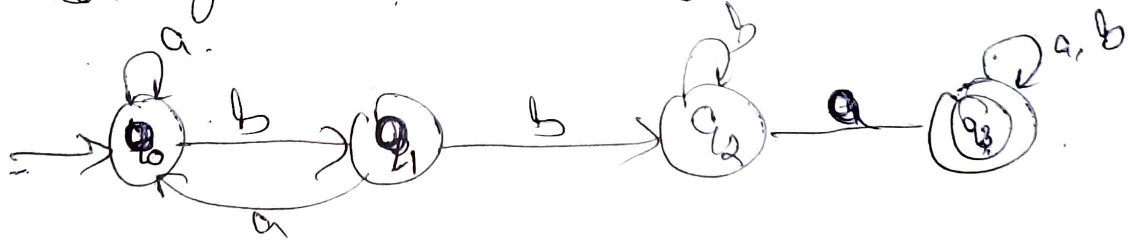
Problems :-

1) DFA :- pattern matching, counter design, divisibility

pattern matching :-

1) Design a DFA over $\Sigma = \{a, b\}$ which will accept the string containing bba as substring.

⇒ Transition diagram ⇒



abbababb ⇒ accepted.

$q_0\ q_1\ q_2\ q_3\ q_3\ q_3\ q_3\ q_3\ q_3$

Transition function ⇒

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

	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_2
q_2	q_3	q_2
$* q_3$	q_3	q_3

$\hat{\delta}$ rule :-

1) $\hat{\delta}(q, \epsilon) = \{q\}$
2) $\hat{\delta}(q, ab) = \delta(\hat{\delta}(q, a), b)$

if given show the acceptance of the string $w = abbab$

i) $\hat{\delta}(q_0, a) = \hat{\delta}(q_0, \epsilon a)$
$= \delta(\hat{\delta}(q_0, \epsilon), a)$
$= \delta(q_0, a)$
$= q_0$

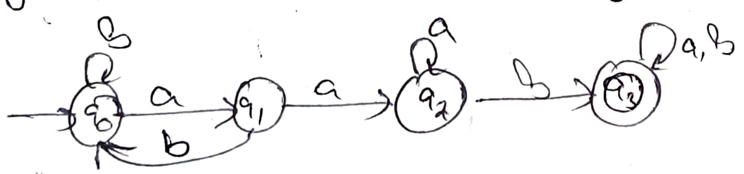
ii) $\hat{\delta}(q_0, ab) = \delta(\hat{\delta}(q_0, a), b)$
$= \delta(q_0, b) = q_1$

iii) $\hat{\delta}(q_0, abb) = \delta(\hat{\delta}(q_0, ab), b)$
$= \delta(q_1, b)$
$= q_2$

iv) $\hat{\delta}(q_0, abba)$
$= \delta(\hat{\delta}(q_0, abb), a)$
$= \delta(q_2, a) = q_3$

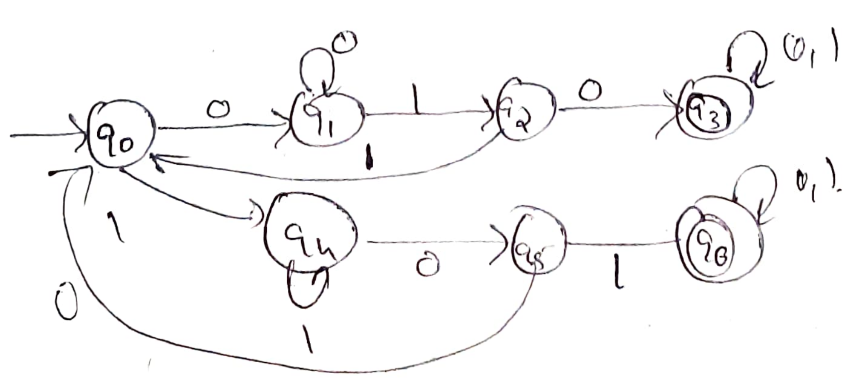
v) $\hat{\delta}(q_0, abbab) = \delta(\hat{\delta}(q_0, abba), b)$
$= \delta(q_3, b)$
$= q_3$ (Accepted)

2) Design a DFA for $L = \{x\,aab\,y \mid x, y \in \{a, b\}^*\}$

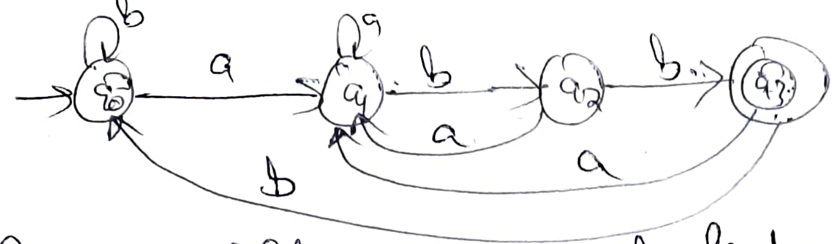


3) $L = \{w \mid w$ contains 010 or 101 as a substring$\}$

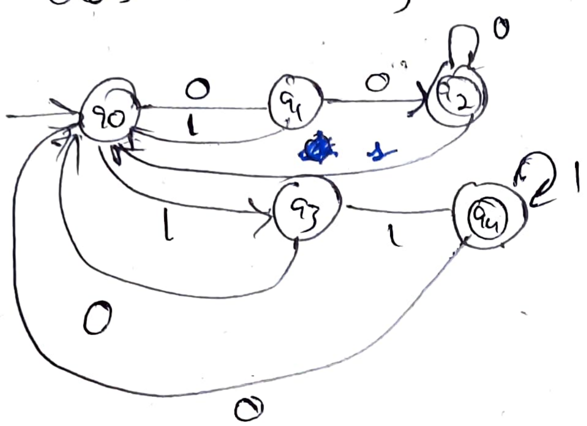
$\Rightarrow$



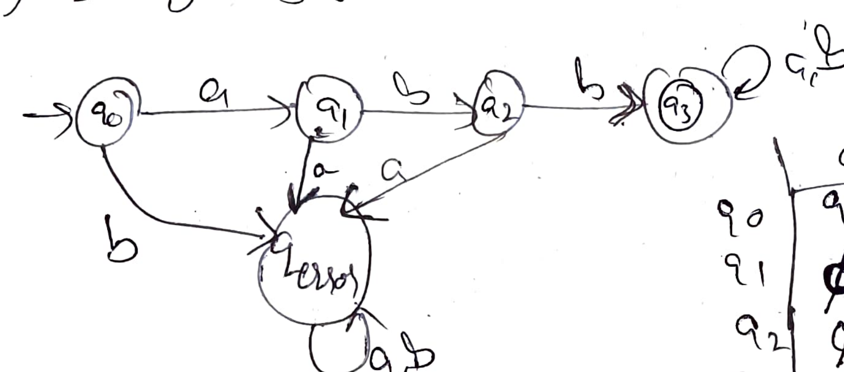
4) Design a DFA over $\Sigma = \{a, b\}$, where all strings end with abb



5) Design a DFA over $\Sigma = \{0,1\}$ for $L = \{w \mid w$ ends with 11 or ends with 00$\}$



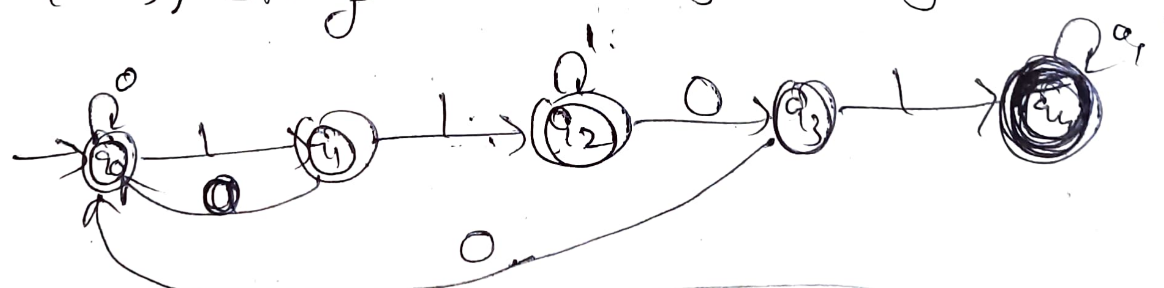
6) strings start with abb



	a	b
q_0	q_1	ϕ
q_1	ϕ	q_2
q_2	ϕ	q_3
q_3	q_3	q_3

7) $\Sigma = \{0,1\}$, strings not containing substring 1101.

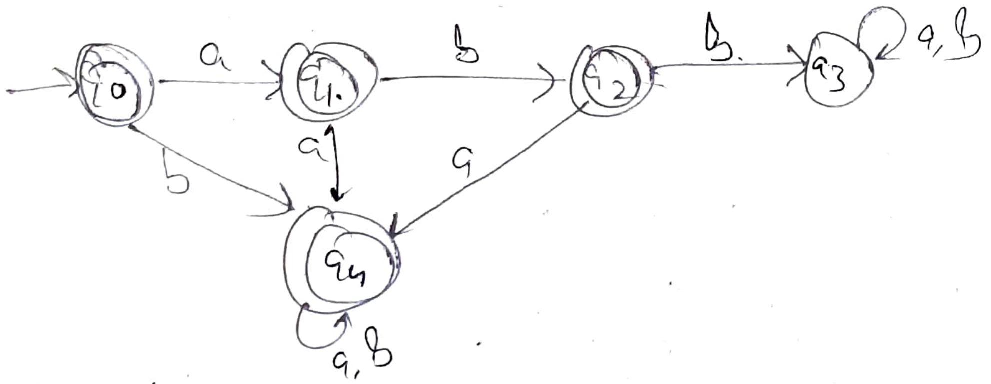
$\Rightarrow$



Note $\Rightarrow$ Change the non-final state to final.

Final state : q_0, q_1, q_2, q_3

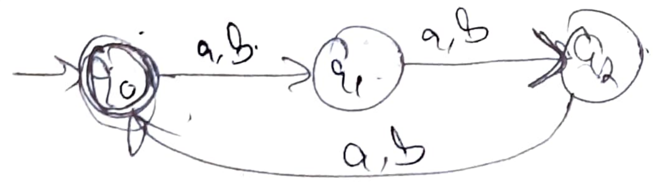
8) not start with abb, $\Sigma = \{a, b\}$



Counter design :-

1) Design a DFA over alphabet $\Sigma = \{a, b\}$ for $L = \{w \mid |w| \bmod 3 = 0\}$

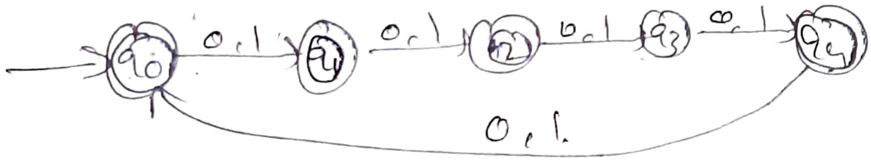
$\Rightarrow$



[Remainders = 0, 1, 2]

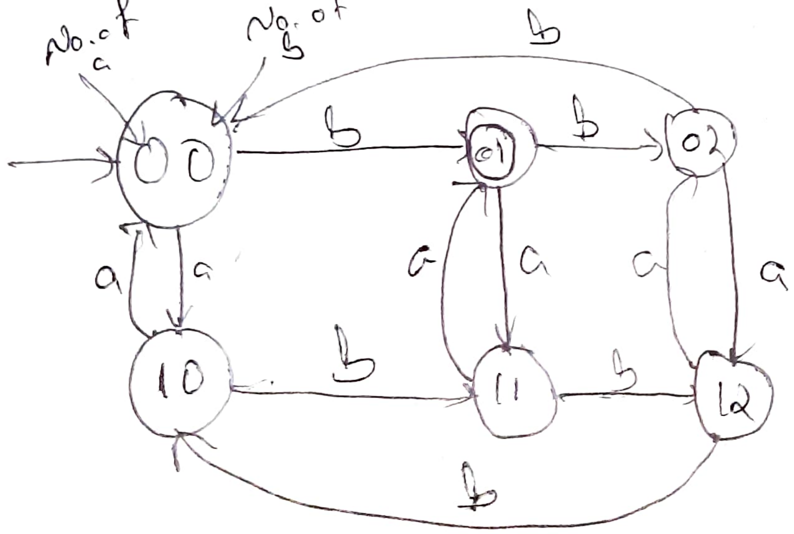
q_0 - final state.

2) Design a DFA over $\Sigma = \{0, 1\}$ for $L = \{w \mid |w| \bmod 5 \neq 3\}$



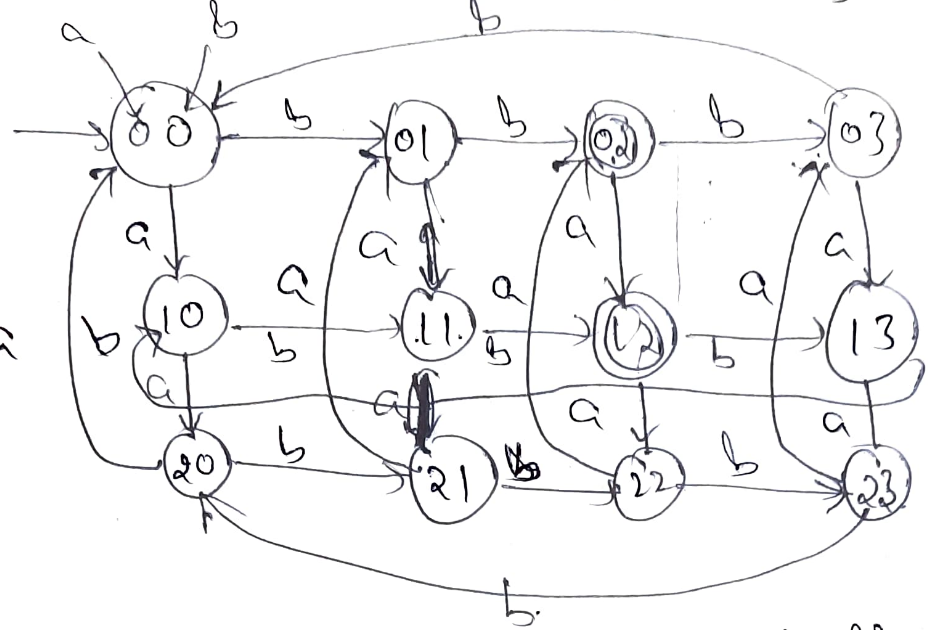
3) Design a DFA over $\Sigma = \{a, b\}$ for $L = \{w \mid N_a(w) \bmod 2 = 0$ and $N_b(w) \bmod 3 = 1\}$

$\Rightarrow$



4) $L = \{ w \mid N_a(w) \bmod 3 \neq 2$ and $N_b(w) \bmod 4 = 2 \}$

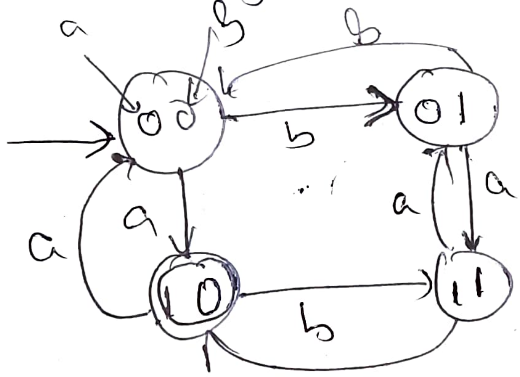
⇒



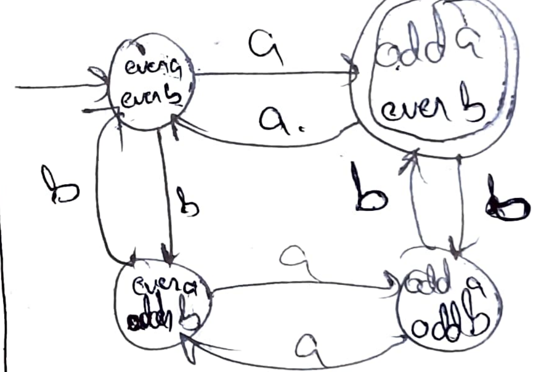
5) Design a DFA over, $\Sigma = \{a, b\}$, which will accept odd no. of a's & even no. of b's

⇒ $N_a(w) \bmod 2 = 1$

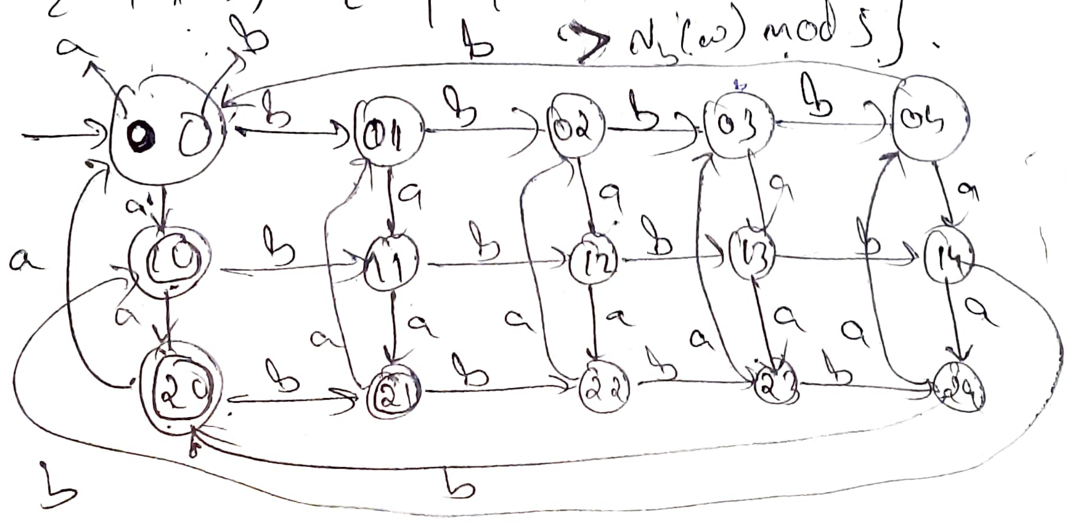
$N_b(w) \bmod 2 = 0$



2)



6) $\Sigma = \{a, b\}$, $L = \{ w \mid N_a(w) \bmod 3 > N_b(w) \bmod 5 \}$.



Divisibility Problem :-

1) Design a DFA which will accept the binary strings whose value is divisible by 5.

or (divisible by 5 & not starting with 0)

⇒ $\Sigma = \{0, 1\}$

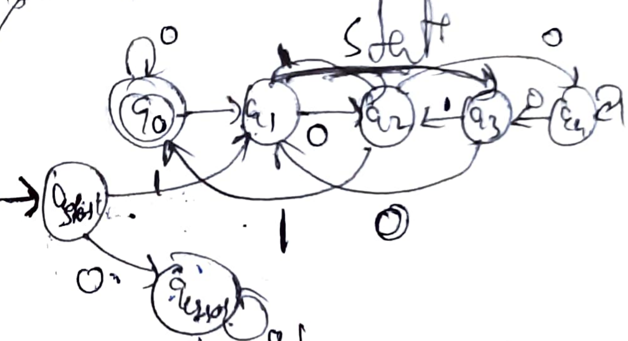
	0	1
$\rightarrow * q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

$(q_i, d) = q_j$

$j = (r * i + d) \bmod k$

r - radix / base
i - current state
j - next state
d - current input

add a new initial state



⇒ $(q_0, 0) = q_j = q_0$

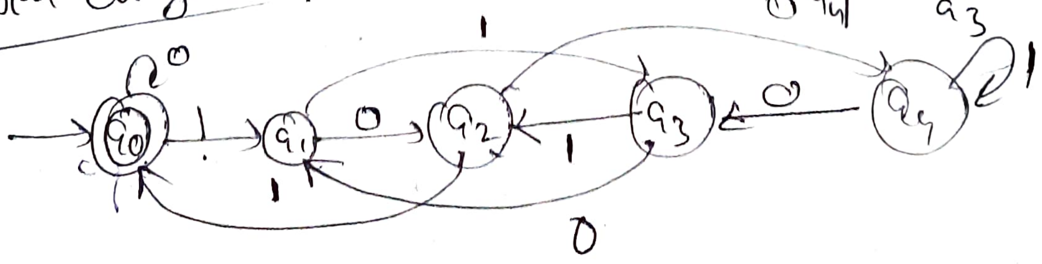
$j = (2 * 0 + 0) \bmod 5 = 0$

⇒ $(q_0, 1) = q_j = q_1$

$j = (2 * 0 + 1) \bmod 5 = 1$

	0	1
$\rightarrow q_{start}$	ϕ	q_1
$* q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

Transition diagram :-

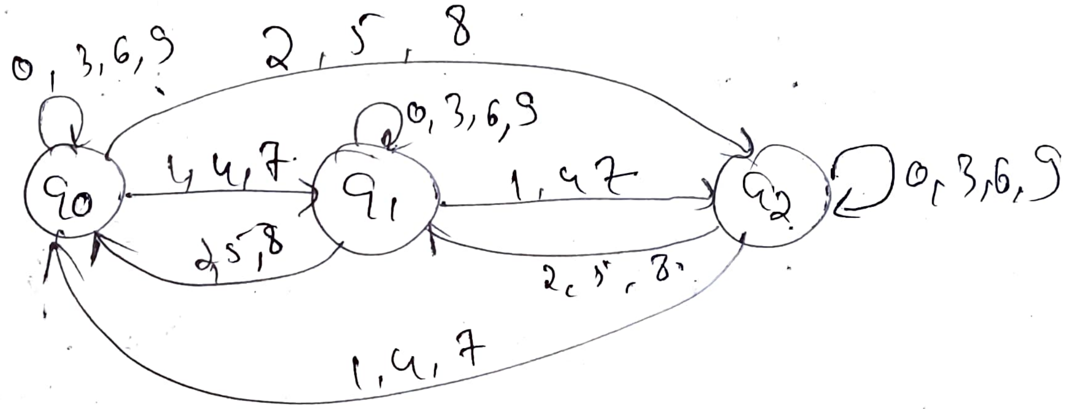


1) Design a DFA which will accept decimal nos divisible by 3

⇒

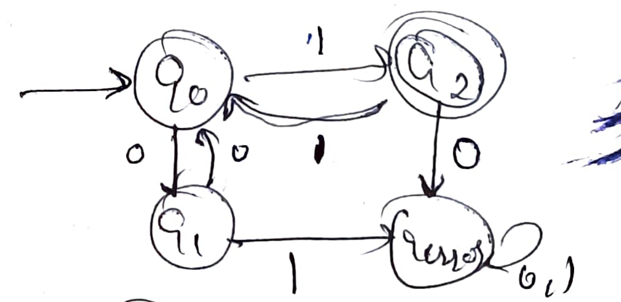
	0	1	2	3	4	5	6	7	8	9
→ * q_0	q_0	q_1	q_2	q_0	q_1	q_2	q_0	q_1	q_2	q_0
q_1	q_1	q_2	q_0	q_1	q_2	q_0	q_1	q_2	q_0	q_1
q_2	q_2	q_0	q_1	q_2	q_0	q_1	q_2	q_0	q_1	q_2

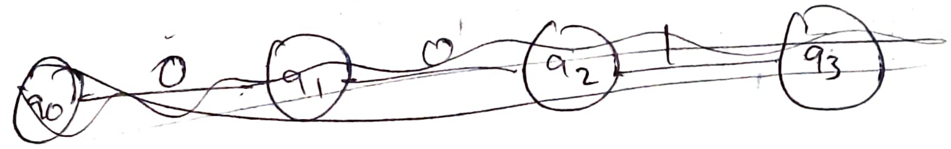
⇒



⇒ L = { w | w is a string of even no. of 0's followed by odd no. of 1's }

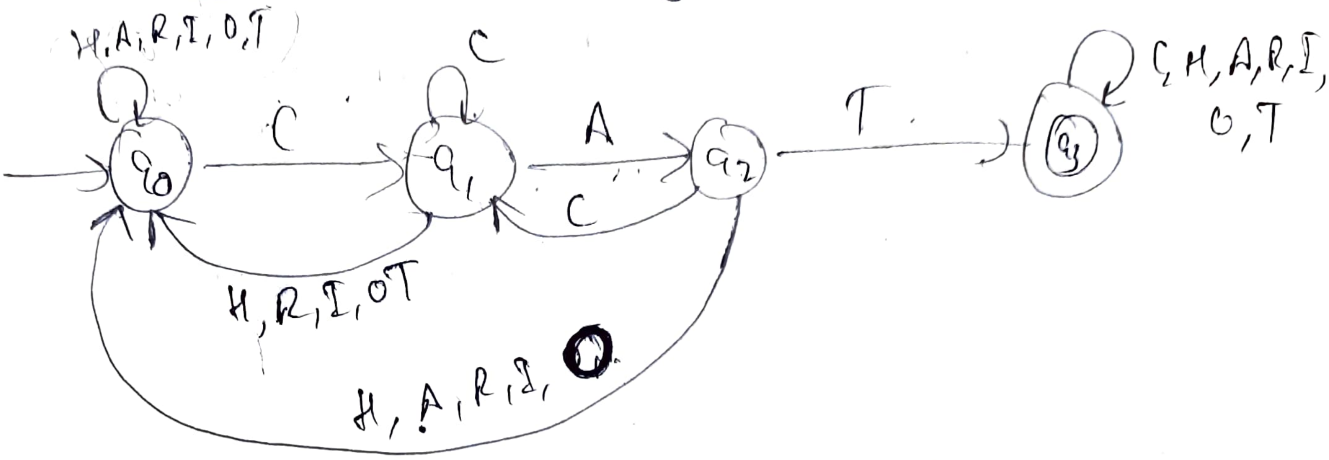
⇒ min. No. of 0's → 2
min. No. of 1's → 1





⇒ Design a DFA to read strings made up of letters "CHARIOT" & recognize these strings that contain the word "CAT" as a substring.

$\Sigma = \{C, H, A, R, I, O, T\}$

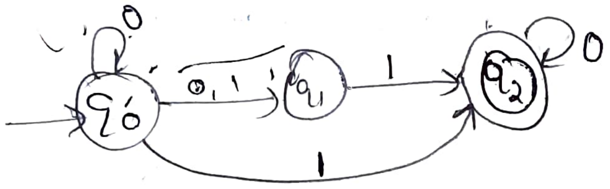


8/01/20

NDFSM (Non-Deterministic Finite State Machine)

NFA ε-NFA

Ex :-



	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_1, q_2\}$
q_1	ϕ	$\{q_2\}$
$* q_2$	$\{q_2\}$	ϕ

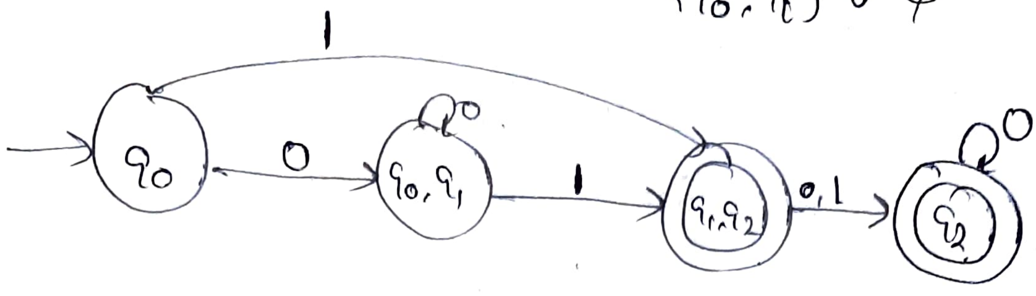
⇒ NFA to DFA :- (Subset construction Method) (Lazy Evaluation Method)

Ex :-

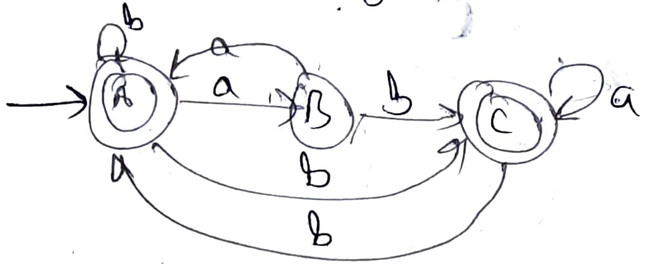
	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_1, q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_1, q_2\}$ ⇒
$* \{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$* \{q_2\}$	$\{q_2\}$	ϕ

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

$= \{q_0,q_1\} \cup \phi = \{q_0,q_1\}$



2) Convert the following NDFSM to DFSM



⇒

	a	b
→ * A	{B}	{A, C}
B	{A}	{C}
* C	{C}	{A}

⇒ NFA transition table

⇒

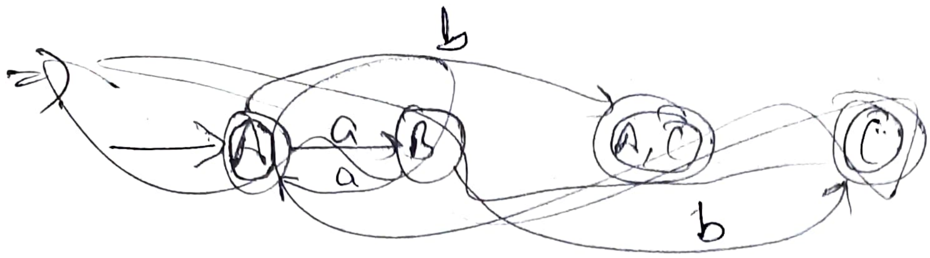
	a	b
→ * A	{B}	{A, C}
{B}	{A}	{C}
* {A, C}	{B, C}	{A, C}
* {C}	{C}	{A}
* {B, C}	{A, C}	{A, C}

$\delta_D(\{A\}, a) = \{B\}$

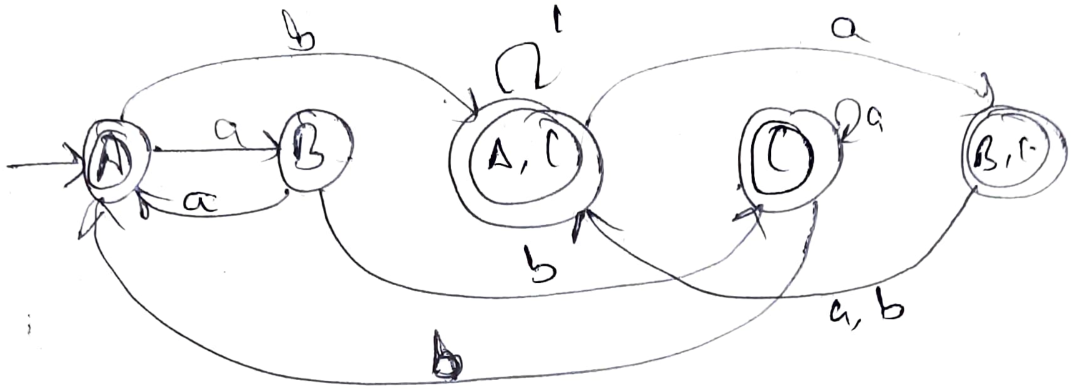
$\delta_D(\{A\}, b) = \{A, C\}$

$\delta_D(\{A,C\}, a) = \{B\} \cup \{C\}$
$= \{B, C\}$

$\delta_D(\{A,C\}, b) = \{A, C\}$



=)

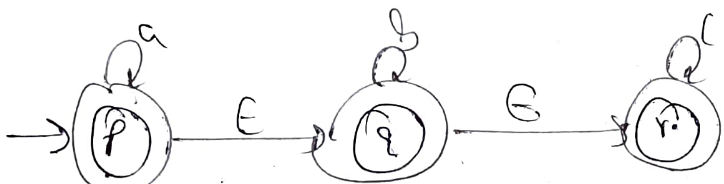


Design

TT

TD

TF

3) 

Step-1:- Find ε Close:-

=) $\varepsilon Close(P) = \{P, q, r\}$

$\varepsilon Close(q) = \{q, r\}$

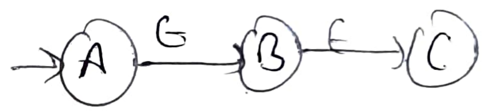
$\varepsilon Close(r) = \{r\}$

DFA:-

	a	b	c
$\rightarrow * \{p, q, r\}$	$\{p, q, r\}$	$\{q, r\}$	$\{r\}$
$* \{q, r\}$	ϕ	$\{q, r\}$	$\{r\}$
$* \{r\}$	ϕ	ϕ	$\{r\}$

=)

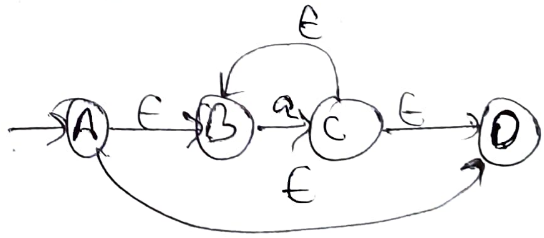
ε-close:-



$\varepsilon Close(A) = \{A, B, C\}$

$\varepsilon Close(B) = \{B, C\}$

$\varepsilon Close(C) = \{C\}$



$\delta(\{p, q, r\}, a) = \{close(\delta(p, a) \cup \delta(q, a) \cup \delta(r, a))$
$= \{close(\{p\} \cup \phi \cup \phi\}$
$= \{close(\{p\}) = \{p, q, r\}$

H/W

4)

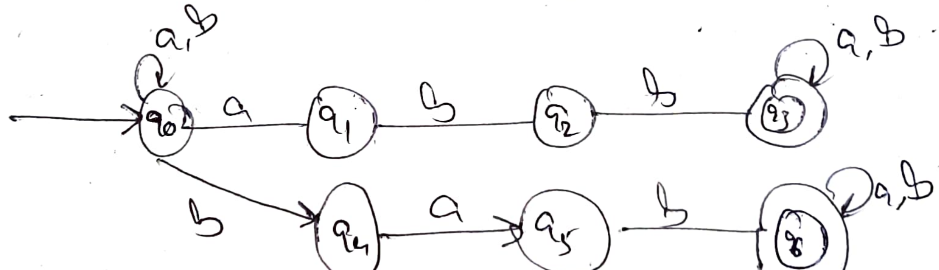
Given =)

	ϵ	a	b	c
$\rightarrow p$	$\{q, r\}$	ϕ	$\{q\}$	$\{r\}$
q	ϕ	$\{p\}$	$\{r\}$	$\{p, q\}$
$*r$	ϕ	ϕ	ϕ	ϕ

=)

5). Design a NDFSM over $\Sigma = \{a, b\}$ which will accept the strings containing substring abb or bab

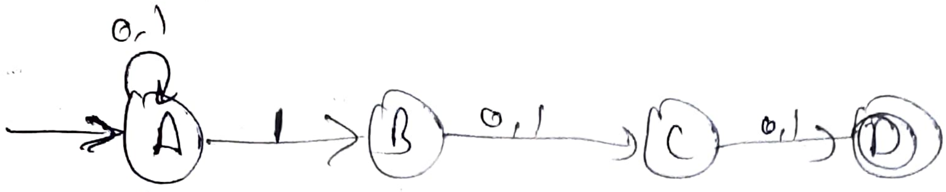
=)



6) Design a NDFSM over $\Sigma = \{0, 1\}$ which will accept the string containing 3rd symbol from RHS as 1, show the processing of a I/P string 00101

7)

(PTO)



string 00101

$\Rightarrow \hat{\delta}(A,0) = \hat{\delta}(A,\epsilon 0) = \delta(\hat{\delta}(A,\epsilon),0) = \delta(A,0) = A$

$\Rightarrow \hat{\delta}(A,00) = \delta(\hat{\delta}((A,0),0)) = \cancel{\delta(\hat{\delta}(A,0}\quad \delta(A,0) = A$

$\hat{\delta}(A,001) = \delta(\hat{\delta}(A,00),1) = \delta(A,1) = \{A,B\}$

$\hat{\delta}(A,0010) = \delta(\hat{\delta}(A,001),0) = \delta(\{A,B\},0) = \delta(A,0) \cup \delta(B,0)$
$= \{A \cup C\} = \{A,C\}$

$\hat{\delta}(A,00101) = \delta(\hat{\delta}(A,0010),1) = \delta(\{A,C\},1) = \{A,B\} \cup \{D\}$
$= \{A,B,D\}$

Accepted

Minimization of DFA

1)

	0	1
→A	B	F
B	G	C
*C	A	G
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

Soln:

Step - 1

P_0	P_1
A, B, D, E, F, G, H	C

Step-2 for i/p = 0

P_0	P_1	P_2
A, B, E, G, H	C	D, F

Step-3 for i/p = 1.

P_0	P_1	P_2	P_3
A, E, G	C	D, F	B, H

Step-4 i/p = 0.

P_0	P_1	P_2	P_3	P_4
G	C	D, F	B, H	A, E

Step 5 check for P_2 for i/p = 0 & 1.

No separation.

Step 6 check for P_3 for i/p = 0 & 1.

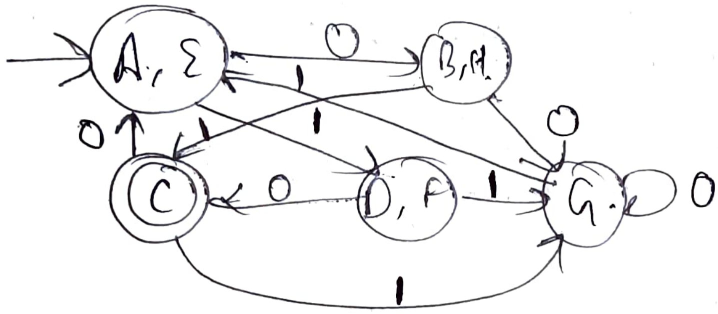
No separation

St-7 :- check for P_4 for i/p = 0 & 1

No separation.

∴

P_0	P_1	P_2	P_3	P_4
G	C	D, F	B, H	A, E.



Distinguishable & Indistinguishable state :-

Distinguishable :- Two states A & B are distinguishable if $\delta(A, a) \in F$ (Final state) & $\delta(B, a) \notin F$ (Final state)

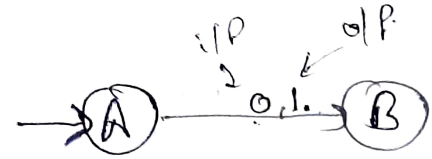
Indistinguishable :- Two states A & B are indistinguishable if both $\delta(A, a)$ & $\delta(B, a) \notin F$ (Final state)

$\delta(A, a)$ & $\delta(B, a) \in F$ (Final state)

Transducer (Mealy M/c & Moore M/c) (Mealy & Moore are DFA)

=> ~~1's Complement~~

Mealy M/c :-



→ O/P associated with transition.

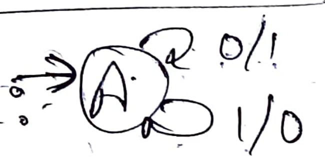
Moore M/c :-

→ o/p associated with state

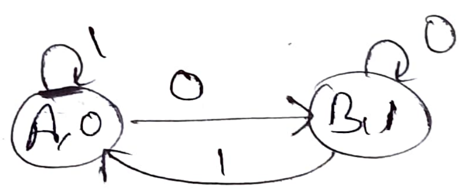
There is no final state in Mealy & Moore M/c

1) 1's Complement :-

Mealy M/c :-



Moore M/c :-



i/p - string → 11001.

Moore M/C :-

	1	1	0	0	1
A	A	A	B	B	A
0	0	0	1	1	0

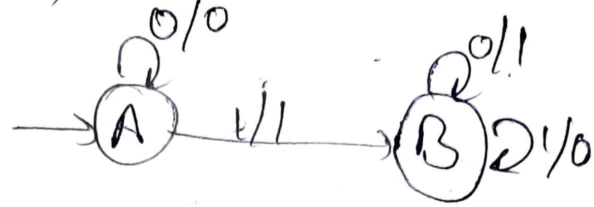
discard

2) 2's Complement :-   10111.
0 0 0

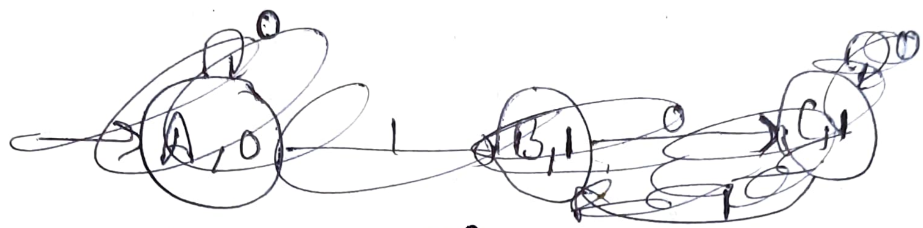
i/p string :- 1 0 1 0 0.

2's compl :- 0 1 1 0 0

Mealy M/C :-



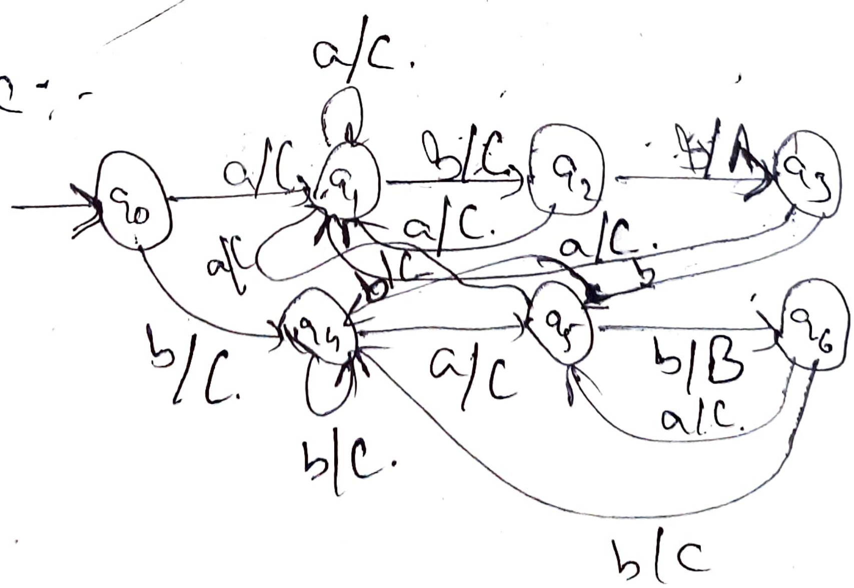
Moore M/C :-



3) Design a Mealy & Moore M/C over $\Sigma = \{a, b\}$ which will give o/p as A on encountering abb, o/p as B on encountering bab, else C.

→

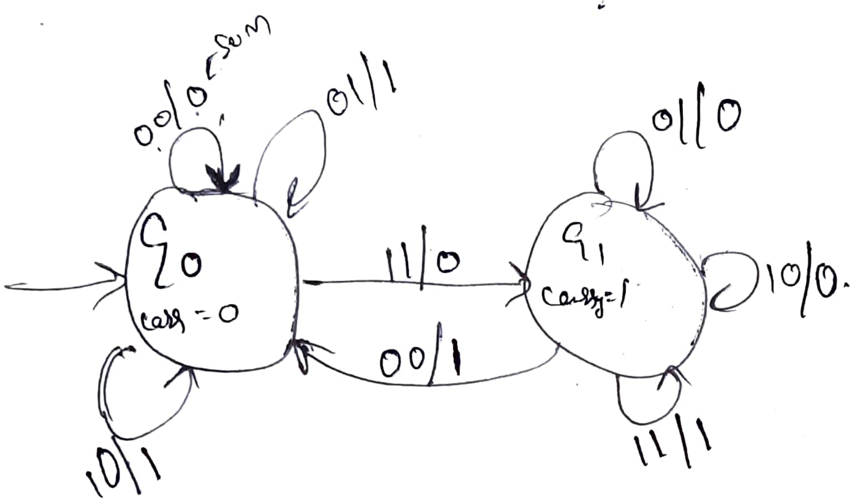
Mealy M/C :-



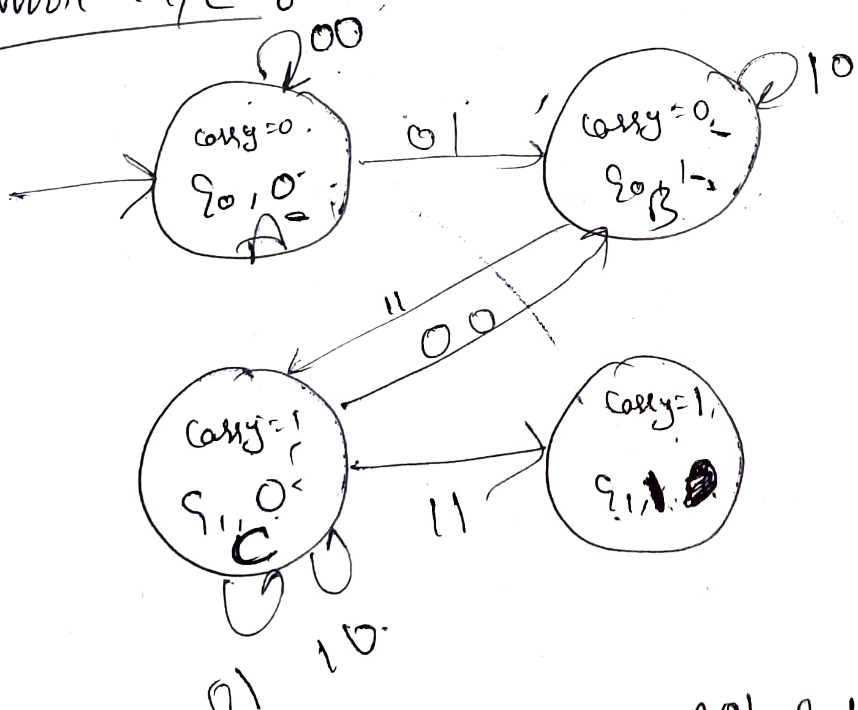
Full adder Circuit :-

C_i	B_i	A_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Mealy M/C :-



Moore M/C :-



1 1 0 1 0
1 0 0 1 1

A $\xrightarrow{01}$ B $\xrightarrow{11}$ C $\xrightarrow{00}$ B $\xrightarrow{10}$ B $\xrightarrow{11}$ C

(discard) ← ~~0~~ 1 0 1 1 0

∴ O/P is from LSB to MSB ⇒ 0110