FLAT

01/08/24

Theory of Computation (TCP)

- nature of computation - systematic application of mathematical and logical methods to describe and model algorithms, computational processes and their efficiency.
- design of computing machines, understanding their limitation of computation, classifying problems based on their complexity and study of automata and formal languages.

Module 1:- Fundamentals.

a ka power

a^* = aaa → 0 or n numbers of a

is

Kleene closure $\boxed{a^+}$ → atleast one a

Turing Machine

Basics:-

1) Alphabet- An alphabet (Σ) is defined as finite set of input symbols

   eg: $\Sigma = \{0, 1, 2\}$

   where 0, 1, 2 are symbols or letters.

2) Strings/sentence/word:- Finite sequence of symbols over given alphabet.

   0, 1, 2, 01, 02, 12, etc over $\Sigma\{0,1,2\}$.

word/String length:- No. of symbols present in given string

01/8/24

Empty String- String of length 0 or null or empty is called epsilon (ϵ)
Null transition- Jump from one state to two without taking inputs.

Language- set of strings which ~~are~~ over alphabet.
eg:- $L = \{x \mid x$ ends in 'ba' over $\Sigma = \{a, b\}\}$
string
should have condition.

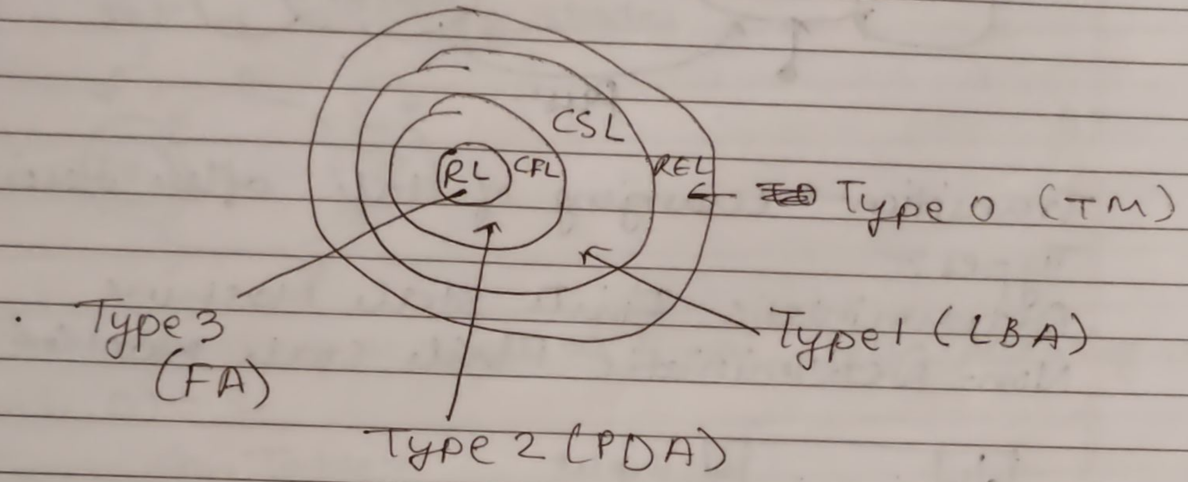
Power of Alphabet- $(\Sigma)^k$
$\Sigma^1 = \{a, b\}$
$\Sigma^2 = \{aa, ab, ba, bb\}$
$\Sigma^0 = \{\} = \{\epsilon\}$

Substring:- u, v String ke substring, if U occurs as it is in same order.
W = GATE
U = {ϵ, G, A, T, E, GA, ~~GT~~, ~~GE~~, AT, ~~AE~~, TE, GAT, ~~GTE~~, ~~GAE~~, ATE, GATE}
Proper substrings = 9
complete word and ϵ not counted in proper substring.
Improper " = 11

$$\text{Proper} = \frac{n(n+1)}{2} \qquad \text{Improper} = \frac{n(n+1)}{2} + 1$$

Types of languages

Regular languages	Context Free Language	Context Sensitive Language	Recursively Enumerable language
↓	↓	↓	↓
Finite Automata	Push Down Automata		Turing Machines



TM > LBA > PDA > FA

T0 > T1 > T2 > T3

Finite State Machine

Classification

- FA without output
  - DFA
  - NFA
  - ε-NFA
- With Output
  - Moore
  - Mealy.

I/P → [FSM] → O/P

3 Tuples:-

I - Input

O - Output

S - State

Σ → [FA] → F

5 Tuples:-

Σ - Input

Q - States

q_0 - Initial State

F - Final State

δ - Transitions

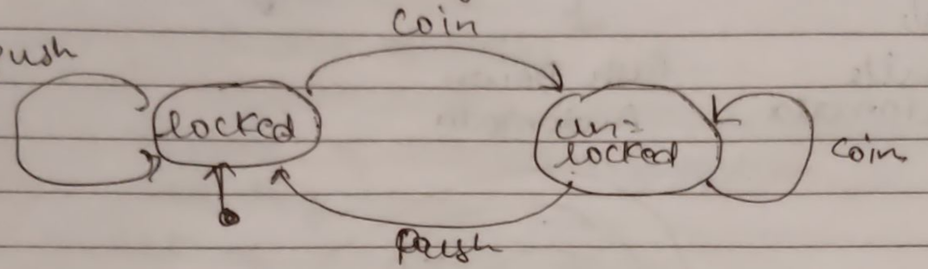
Teacher's Signature:........

02/08/24

PAGE No.
DATE / /

Basics of FSM

1) Finite State Machine or Finite state Automata
- abstract Model of any problem.

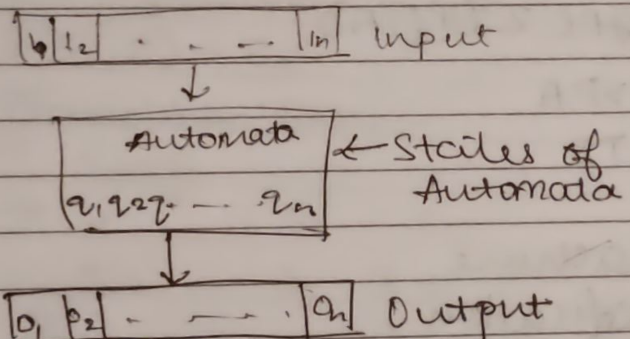


Transition Diagram

Transition → changing of state after receiving input

Types:-
Deterministic Finite State Machine
Non-Deterministic Finite State Machine

* diagram



Transition Table

Current State	Input	Next State	Output
Locked	coin	unlocked	unlocks
	push	locked	None
unlocked	coin	unlocked	None
	push	locked	locks.

No. of rows = No. of Transitions.

Teacher's Signature:..................

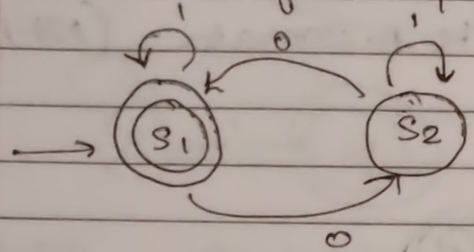
Teacher's Signature:..................

02/08/24

PAGE NO.
DATE / /

Deterministic FSM.
- Specific Output — Does not accept epsilon as input
- Five element Tuple

Q - Finite set of states
Σ - Finite, non empty input alphabet.
δ - series of transition function
$q0$ - starting state
F - set of accepting states



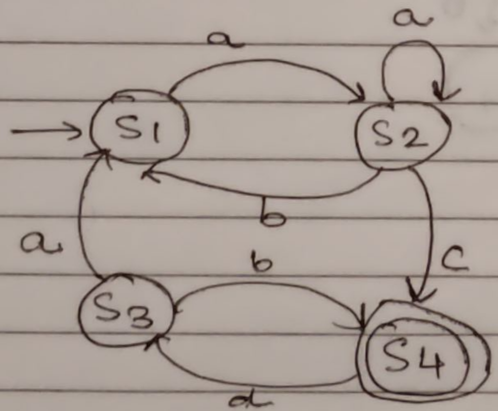
◎→ starting as well as Final state / Accepting state
$\delta(S_1, 0) = S_2$

Transition Table:-

State	0	1
→ S1 *	S2	S1
S2	S1	S2

4 elements - 4 transitions

1)

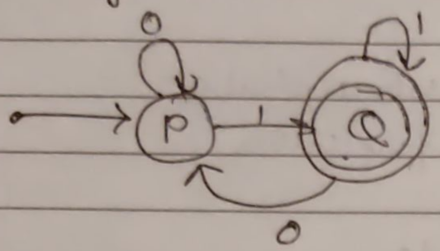


	a	b	c	d
S1	S2			
S2	S2	S1	S4	
S3	S1	S4		
S4				S3

abacdaac ✓
abac ✓
aaaac ✓
aaaacd ✗

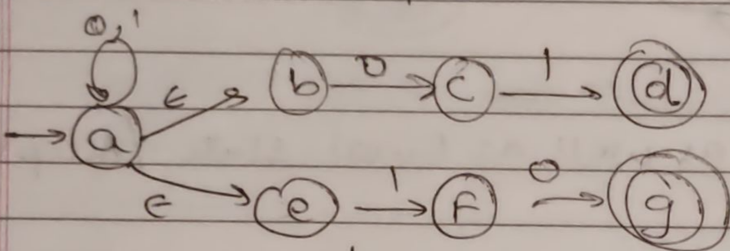
PAGE No.
DATE / /

2) Draw a diagram for DFA. Language of all strings that end with 1.

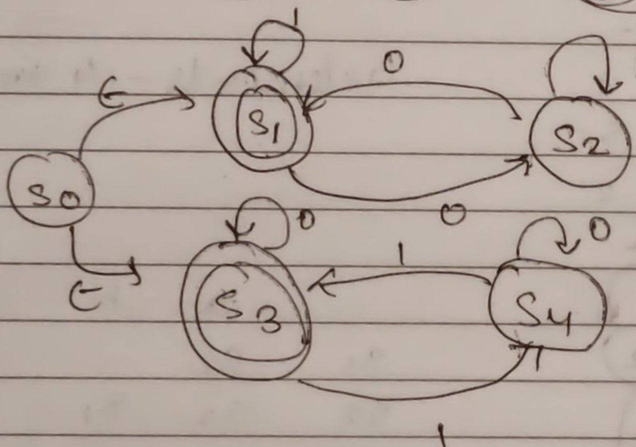


2) Non-Deterministic Finite Automata:- (NFA)

- Accept epsilon as input
- Five element tuple

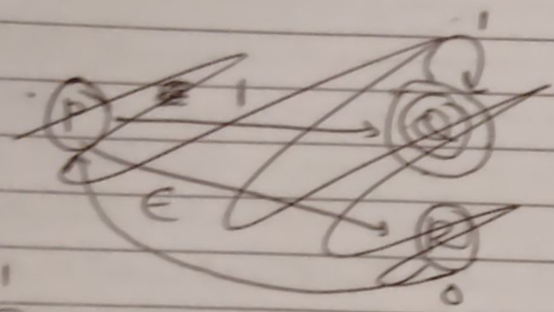


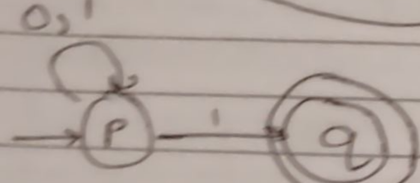
eg.



a) 1 ✓
b) 01001 ✓
c) 1011101 ✓
d) 1000 ✗
e) 0 ✓

2) Language of all strings that end with 1.





Type 3) NFA with epsilon ~~wit~~ Transition [Not compulsory]

Teacher's Signature:..................

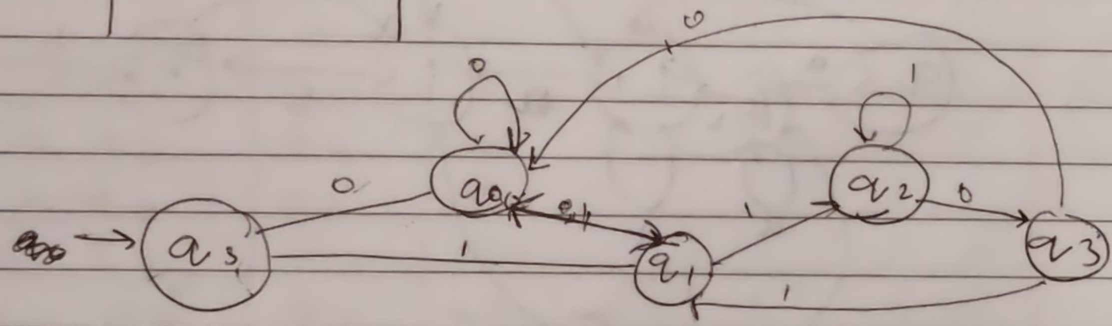
FLAT   Type 1 :- Divisible

08/8/24   Type 2 :→ Ending with .

Design a FSM to accept strings which are ending with '110' over the $\Sigma = \{0, 1\}$
→

1. Theory
2. Logic
   $I = \{0, 1\}$
   $O = \{Y, N\}$
   $S = \{q_s, q_0, q_1, q_2, q_3\}$
3. Implementation.

		0	1
	→ q_s	q_0 / N	q_1 / N
0	q_0	q_0 / N	q_1 / N
1	q_1	q_0 / N	q_2 / N
11	q_2	q_3 / Y	q_2 / N
110	q_3 *	q_0 / N	q_1 / N



6. Simulation.
   $(q_0, 011010)$
   $\vdash (q_0, 11010, N)$
   $\vdash (q_1, 1010, NN)$
   $\vdash (q_2, 010, NNN)$
   $\vdash (q_3, 10, NNNY)$
   $\vdash (q_1, 0, NNNYN)$
   $\vdash (q_0, NNNYNN)$
   Reject

Teacher's Signature:.................

2) 'babb' over $\Sigma = \{a, b\}$

→ 1. Theory

2. Logic

$I = \{a, b\}$

$O = \{Y, N\}$

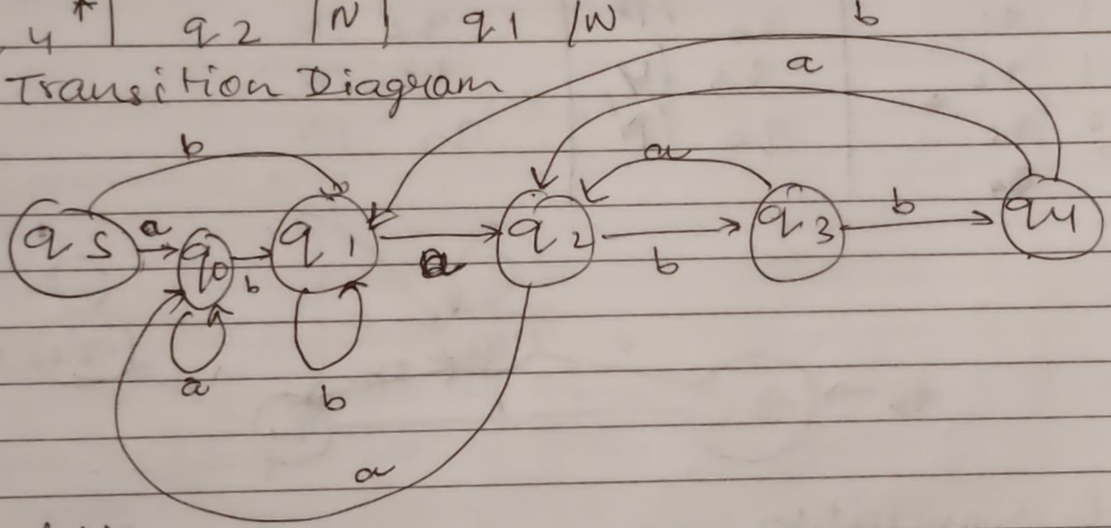
$S = \{q_s, q_1, q_2, q_3, q_4\}$

3. Implementation

baba q_0

		a	b
→	q_s	q_0 | N	q_1 | N
a	q_0	q_0 | N	q_1 | N
b	q_1	q_2 | N	q_1 | N
ba	q_2	q_0 | N	q_3 | N
bab	q_3	q_2 | N	q_4 | Y
babb	q_4 *	q_2 | N	q_1 | N

4. Transition Diagram



5. Simulation

8/8/24 Type 3:- Strings containing Type Trap State

1) Design a FSM to accept strings which contains '1011' over inputs $\Sigma = \{0, 1\}$

→ 1. Theory (Remains same)

2. Logic

		0	1
	$\rightarrow q_s$	q_0/N	q_1/N
0	q_0	q_0/N	q_1/N
1	q_1	q_2/N	q_1/N
10	q_2	q_0/N	q_3/N
101	q_3	q_2/N	q_4/Y
1011	q_4^*	q_4/Y	q_4/Y

2) 'bba' $\Sigma = \{a, b\}$

1.

2.

3.

		a	b
	$\rightarrow q_s$	q_0	q_1
a	q_0	q_0	q_1
b	q_1	q_0	q_2
bb	q_2	q_3	q_1
bba	q_3^*	q_3	q_3

Type 4:- String Not containing Type

Type 4 :- Does not contain

1) Does not contain consecutive b's over $\Sigma = \{a,b\}$

		a	b
	q_s *	q_0/Y	q_1/Y
a	q_0 *	q_0/Y	q_1/Y
b	q_1 *	q_0/Y	q_2/Y
bb	q_2 *	q_0/Y	q_3/N
bbb	q_3	q_3/N	q_3/N

Trap State ~~or~~ Dead State

Accepting state or Final State — Not final or Not Accepting State.

Type 5 :- Strings containing ODD/EVEN SEQUENCE

1. FSM to accept strings containing ODD number of a's.

→

		a	b
	→ q_s	q_0	q_1
ODD	q_0 *	q_1	q_0
Even	q_1	q_0	q_1

Type 6 :- EVEN AND ODD.

PAGE NO.
DATE: / /

- Design string containing even number of a's and odd numbers of b's.

	a	b
→ q_s	q_1	q_1
odd odd q_0	q_2	q_1
odd even q_1	q_3	q_0
even odd q_2 *	q_0	q_3
even even q_3	q_1	q_2

abaab ✗ baaabb. ✓

{q_1 baab}

{q_1 aab}

{

Type 7 :- Strings containing mixed sequence.

1. FSM :- a) Exactly 3 a's → q_3 (Final states)
   b) ~~Atleast~~ 3 a's → q_3, q_4
   c) Atmost 3 a's → q_s, q_1, q_2, q_3

	a	b
q_s	q_1	q_0
q_0	q_1	q_0
→ 1 q_1	q_2	q_1
→ aa q_2	q_3	q_2
→ aaa q_3	q_4	q_3
More than 3a aaaa q_4	q_4	q_4

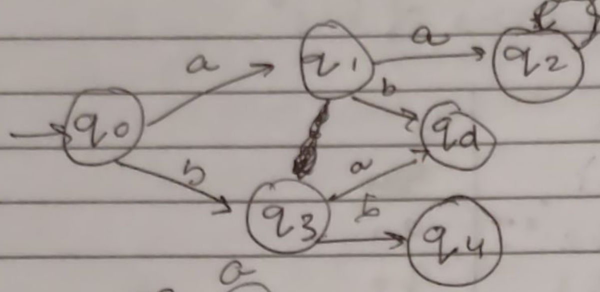
$\boxed{\text{FLAT}}$

Type 8:- Misc. Examples.

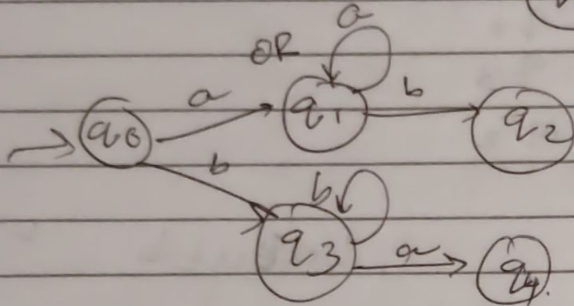
"Start with"

DFA

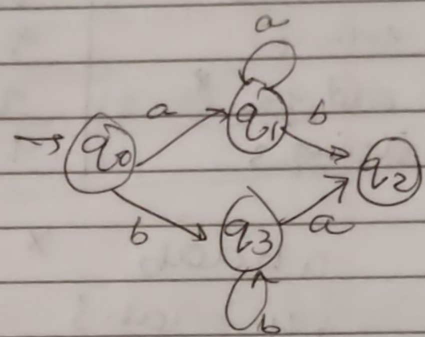
1. Starts with aa or with bb from $\{a, b\}$



OR

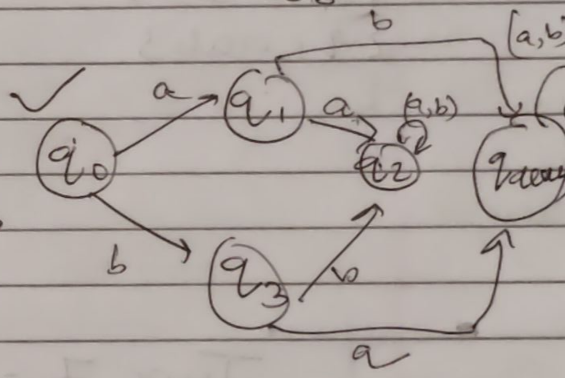


OR



	a	b
q_0	q_1	q_3
q_1	q_2	q_{Dead}
q_3	q_{dead}	q_2
q_2	q_2	q_2
q_{dead}	q_{dead}	q_{dead}

$\rightarrow$



09/8/24

Module 2 :-

Regular Languages and Expressions

The set of strings accepted by finite automata is known as Regular ~~Expression~~ Language. Expression used to describe regular language is known as Regular Expression.

- $+ \Rightarrow$ OR / Union
- $* \Rightarrow$ Closure / All combi
- $\cdot \rightarrow$ Concatenation / AND
- $\phi \Rightarrow$ Empty set
- $\epsilon \Rightarrow \{\}$

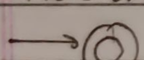
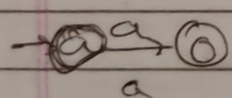
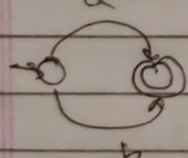
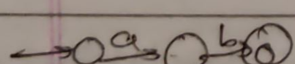
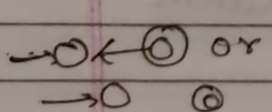
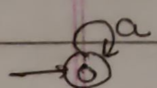
Kleene's Closure

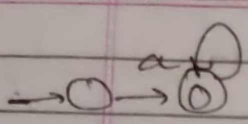
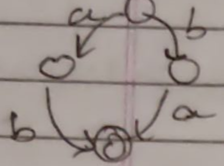
$a^* = \{\epsilon, a^0, a^1, a^2, \dots\}$

Additive Closure

$a^+ = \{a^0, a^1, a^2, a^3 \dots\}$

we can say that $a^+ = a \cdot a^*$

Automata	Language	Regular Expression
	$\{\epsilon\}$	ϵ
	$\{a\}$	a
	$\{a, b\}$	$a+b$, $a \mid b$
	$\{a, b\}$	ab, $a \cdot b$
	ϕ	ϕ
	$\{\epsilon, a, aa, aaa, \dots\}$	a^*

	{a, aa, aaa.}	a^+, $a \cdot a^*$
	{ab, ba}	$(ab + ba)$ or $(ab \mid ba)$

1.) Set of all the strings that start with "a" over $\Sigma = \{a, b\}$

$R.E = a \cdot (a+b)^*$
$L(R) = \{a, aa, ab, aaa, \ldots\}$

2. Set of all strings ends with 0 or 1.
$\Sigma = \{0, 1\}$

$R.E = (0+1)^* \cdot (0+1)$ or $(0+1)^* \cdot 0 + (0+1)^* \cdot 1$
$L(R) = \{0, 1, 00, 01, 10, 11, \ldots\}$

3. Strings that starts with a and ends with b.
Over $\Sigma = \{a, b\}$
→ $R.E = a \cdot (a+b)^* \cdot b$
$L(R) = \{ab, aab, abb, \ldots\}$

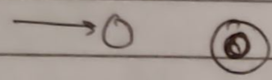
4. String start with x and ends with xy over
$\Sigma = \{x, y\}$
Merge pattern
→ $R.E = x \cdot (x+y)^* \cdot (x \cdot y) + xy$
$L(R) = \{xy, xyxy, xxyyxy, \ldots\}$

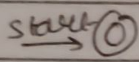
5. starts with abb, ends with bba.

→ $abb(a+b)^* bba + abba + abbba$

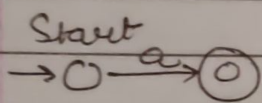
R.E to FA

Rules of Conversion.

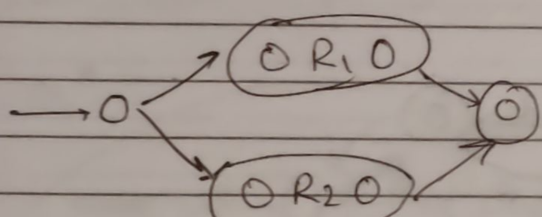
Rule 1:-  For $R.E = \phi$

Rule 2:- 

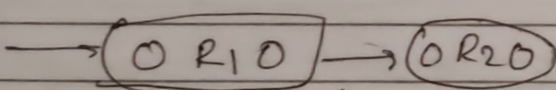
For $RE = \epsilon$

Rule 3:- 

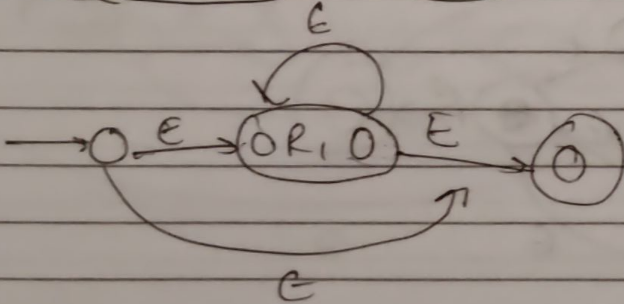
For $RE = a$

Rule 4:- 

For $R.E = R_1 + R_2$

Rule 5:- 

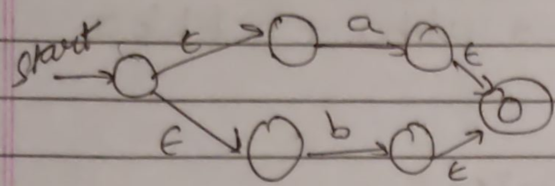
For $R.E \Rightarrow R1 . R2$

Rule 6:- 

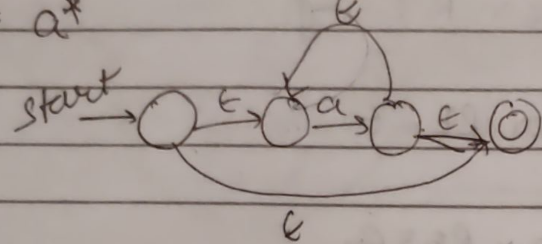
For $RE \Rightarrow R_1^*$

1. Convert :-

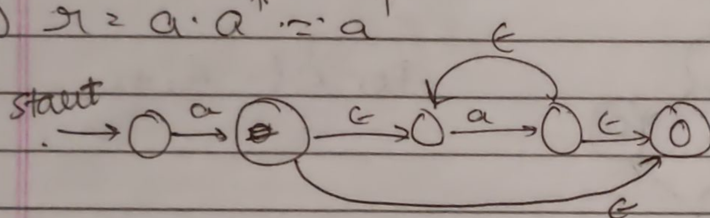
1) $r = a + b$



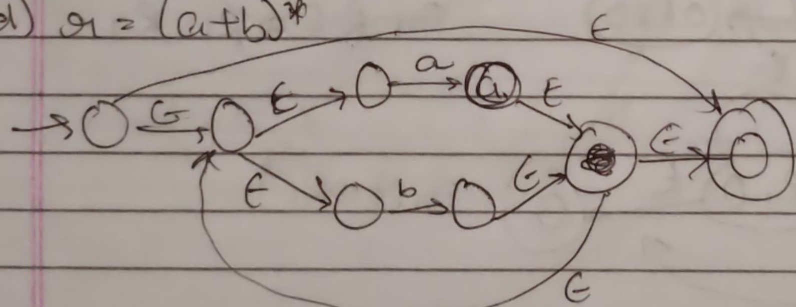
b) $r = a^*$



c) $r = a \cdot a^* = a^+$



d) $r = (a+b)^*$



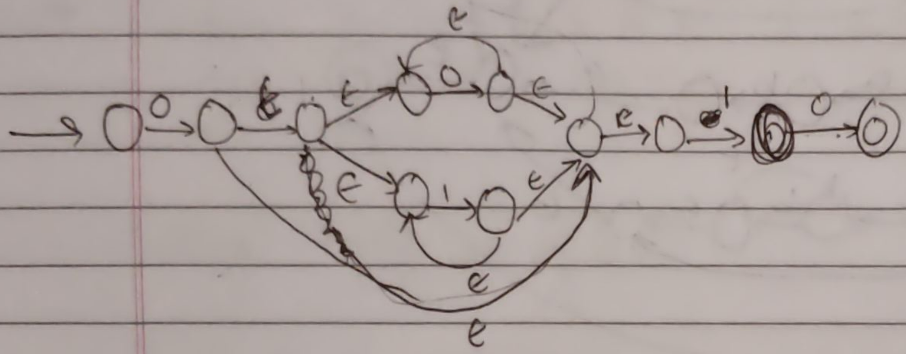
09/08/24

PAGE No.
DATE / /

2. Construct F.A.

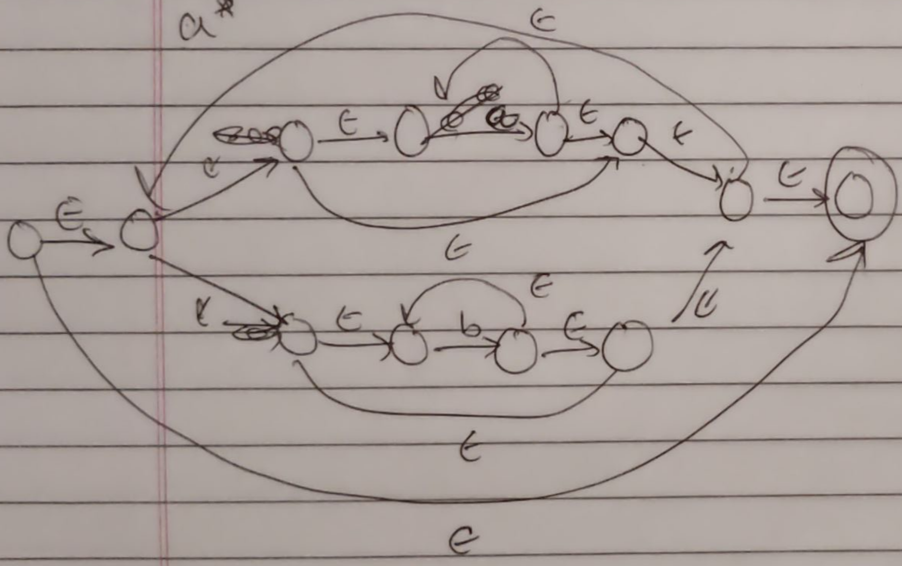
$0.(0+1)^*.10$
↓ ↓ ↓
A B C

$A.(B)^*.C$



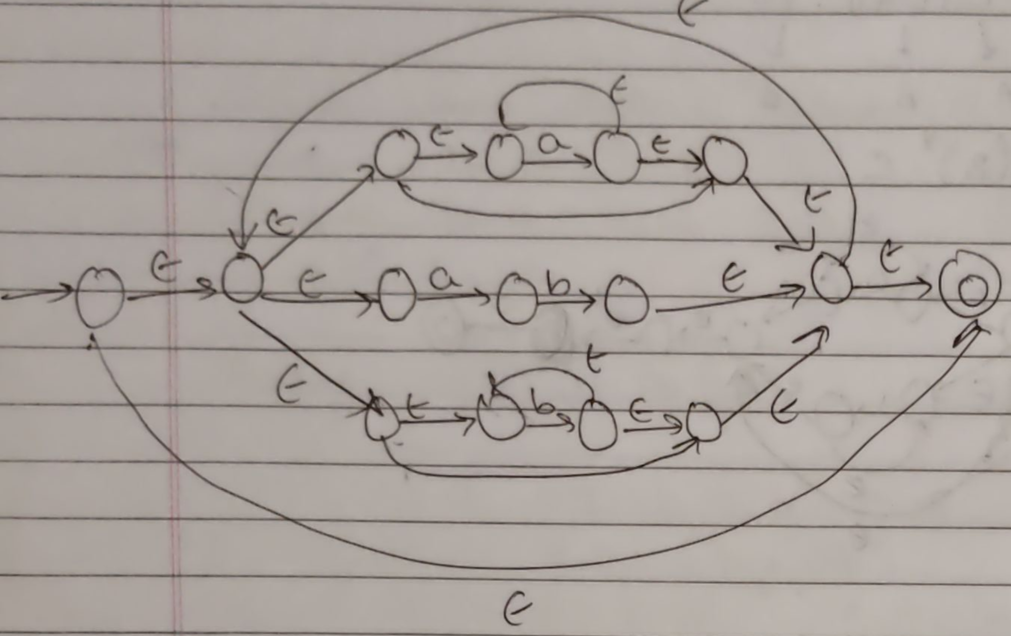
3) $r = (a^* + b^*)^*$ $C = (A+B)^*$

a^*



Teacher's Signature:..................

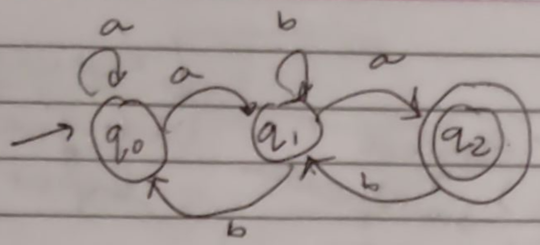
$(a^* + ab + b^*)^*$



FLAT

16/8/24

PAGE NO.
DATE: / /



Solve using Arder's Theorem. FA to R·E

→ $q_0 = q_0 \cdot a + q_1 \cdot b + \epsilon$ — ①

$q_1 = q_0 \cdot a + q_1 \cdot b + q_2 \cdot b$ — ②

$q_2 = q_1 \cdot a$ — ③

Putting eq ③ in eq ②

$q_1 = q_0 \cdot a + q_1 \cdot b + q_1 \cdot a \cdot b$

$\;\; = q_0 \cdot a + q_1 (b + ab)$ $\qquad R = Q + Rb$

$q_1 = q_0 \cdot a + (b+ab)^*$ → further solve for $[q_0 \cdot a]$

PAGE NO.
DATE: / /

DFA Minimization. → Myhill-Nerode Theorem (BOX method)

Classical method involving ϵ moves

Direct method (No ϵ) ← subset generation

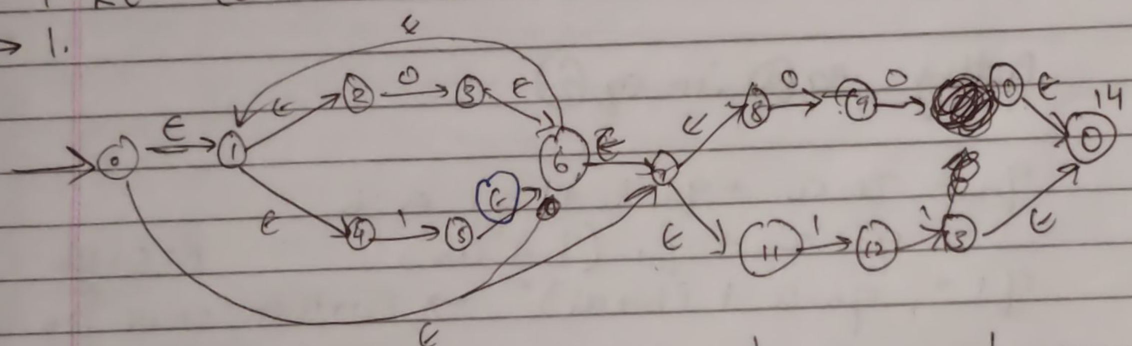
DFA given as input

1. RE to NFA
2. NFA to DFA
3. DFA to Min DFA

1. NFA to DFA
2. DFA to Min DFA

1. RE = $(0+1)^*(00+11)$

→ 1.



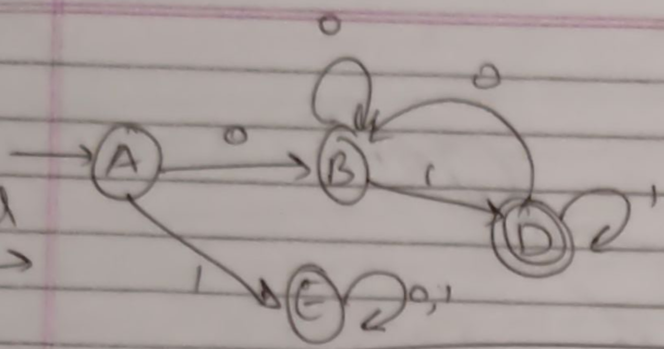
2. ϵ closure Table

	x	$y = \epsilon$ - closure(x)	$\delta(y,0)$	$\delta(y,1)$
A	{0}	{0,1,2,4,7,8,11}	{3,9}	{5,12}
B	{3,9}	{3,6,7,8,11,1,2,4,9}	{3,9,10}	{5,12}
C	{5,12}	{5,6,7,8,11,1,2,4,12}	{3,9}	{5,12,13}
D	{3,9,10}	{1,2,3,4,6,7,8,9,11,10,14}	{3,9,10}	{5,12}
E	{5,12,13}	{1,2,4,5,6,7,8,11,12,13,14}	{3,9}	{5,12,13}

Q | ϵ	0	1
A	B	C
B	D	C
C	B	E
D*	D	C
E*	B	E

] 14 initial Final state ϵ closure mei 14 D aur E mei hai

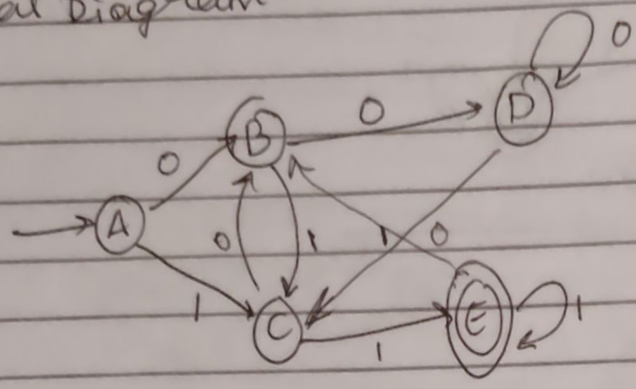
Final DFA →



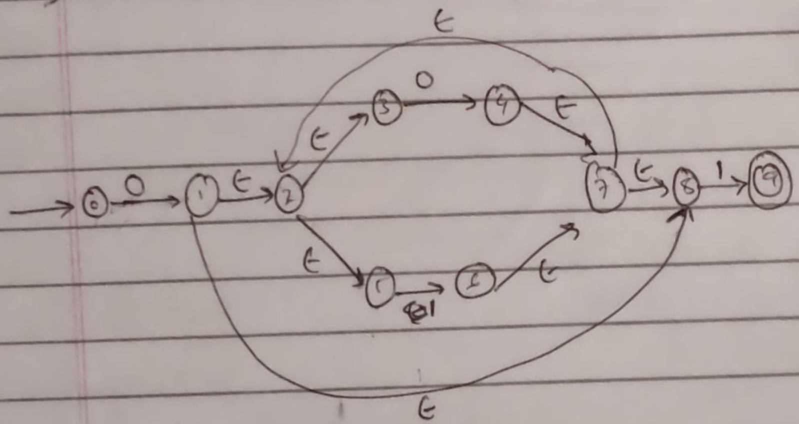
16/10/24

PAGE NO.
DATE / /

Final Diagram.



2) RE to DFA ⇒ $0(0+1)^* 1$



	x	y = ε-closure(a)	δ(y, 0)	δ(y, 1)
A	{0}	{0}	{1}	{ }
B	{1}	{1, 2, 3, 5, 8}	{4}	{6, 9}
C	{4}	{4, 7, 2, 3, 5, 8}	{4}	{6, 9}
D	{6 9}	{6, 7, 8, 2, 3, 5, 9}	{4}	{6, 9}
E	{ }	{ }	{ }	{ }

→ A	B	E
B	C	D
C	C	D
D*	C	D
E	E	E

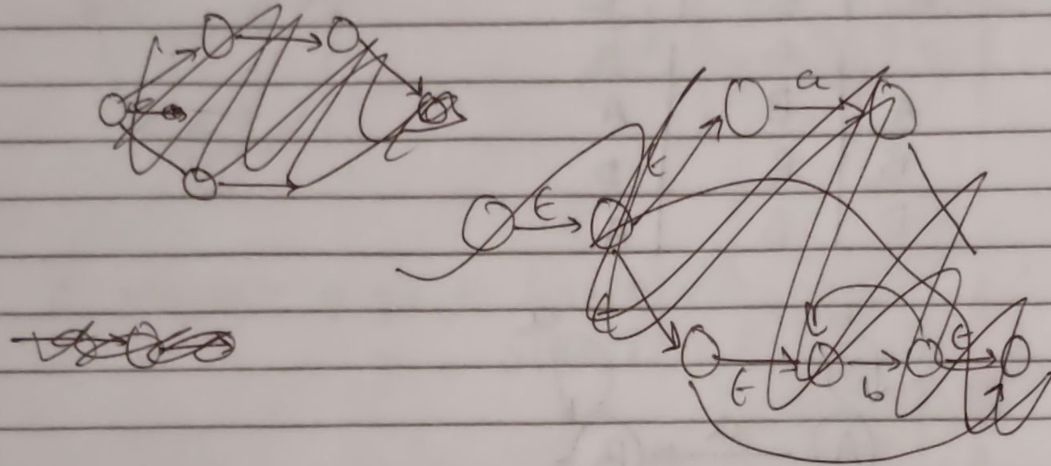
→

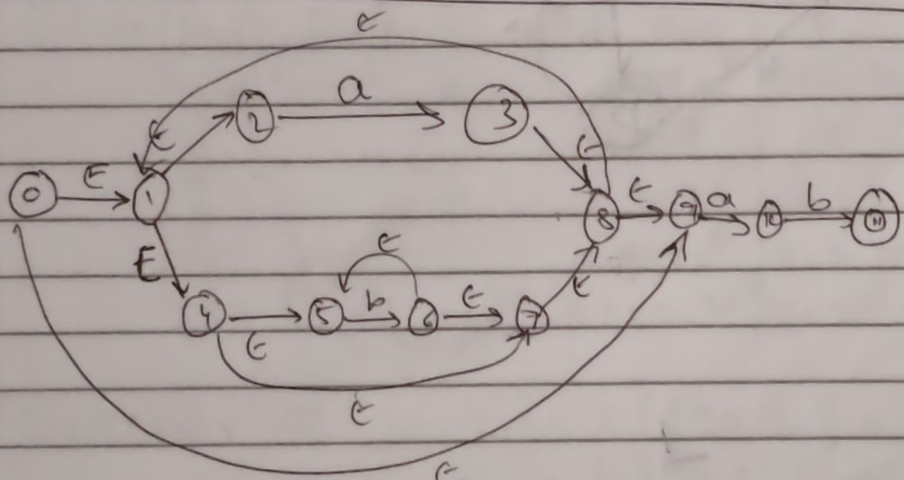
A	B	E
B	B	D
D*	B	D
E	E	E

FLAT

17/08/24

1) R.E $(a+b^*)^* ab$.

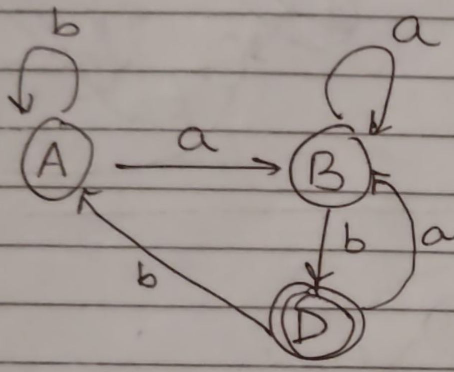




	x	$y = \epsilon$ - closure(x)	$\delta(y, a)$	$\delta(y, b)$
A	{0}	{0, 1, 2, 4, 5, 7, 8, 9}	{3, 10}	{6}
B	{3, 10}	{3, 8, 9, 1, 2, 4, 5, 7, 10}	{3, 10}	{6, 11}
C	{6}	{5, 6, 7, 8, 1, 2, 4, 5, 9}	{3, 10}	{6}
D*	{6, 11}	{6, 5, 7, 8, 9, 1, 2, 4, 11}	{3, 10}	{6}

	a	b
→ A	B	C
B	B	D
C	B	C
D*	B	C

	a	b
A	B	A
B	B	D
D	B	A

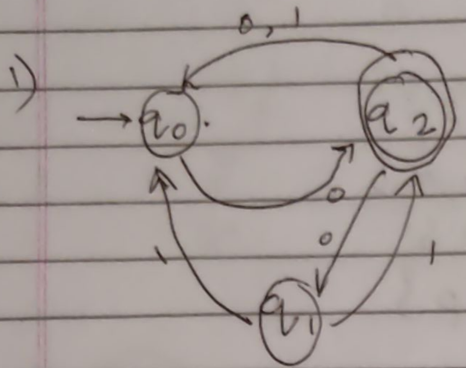


PAGE NO.
DATE: / /

• 17/08/24

Subset generation or Direct Method.

1)

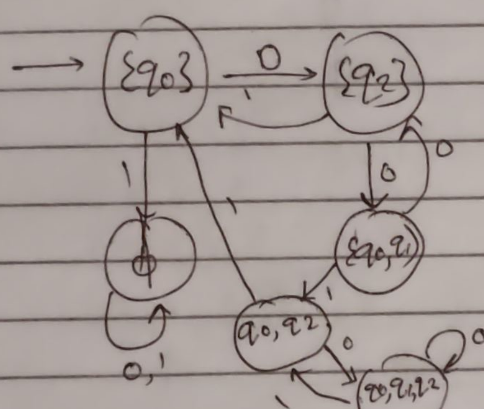


	0	1
$\rightarrow q_0$	$\{q_2\}$	ϕ
q_1	ϕ	$\{q_0, q_2\}$
q_2^*	$\{q_0, q_1\}$	$\{q_0\}$

$\longrightarrow$

Step 1. 0 successor of $q_0 \Rightarrow \delta(q_0, 0) = q_2$

1 " " $\delta(q_0, 1) = \phi$



		0	1
A	q_0	$\{q_2\}$	ϕ
~~B~~	~~q_1~~	ϕ	$\{q_0, q_2\}$
C	q_2^*	$\{q_0, q_1\}$	$\{q_0\}$
D	ϕ	ϕ	ϕ
E	q_0, q_1	$\{q_2\}$	$\{q_0, q_2\}$
F	q_0, q_2^*	$\{q_0, q_1, q_2\}$	$\{q_0\}$
G	q_0, q_1, q_2^*	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$

successor of $\{q_0, q_1\} - \delta\{\{q_0, q_1\}, 0\}$

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

$= q_2 \cup \phi$

$= \{q_2\}$

AW

2) NFA to DFA

	0	1
$\{p\}$	$\{q, s\}$	$\{q\}$
q^*	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
s^*	ϕ	$\{p\}$

3)

	0	1
p	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
s^*	ϕ	ϕ
t^*	ϕ	ϕ

FLAT

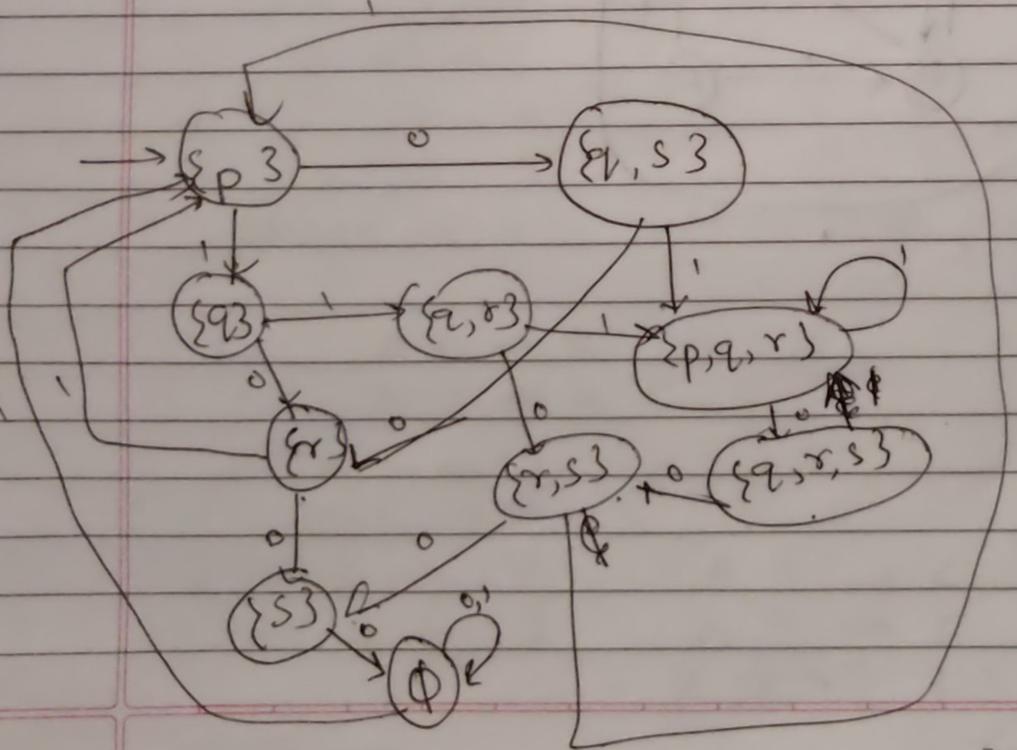
22/08/24

NFA to DFA

1.

	0	1
{p}	{q, s}	{q}
q*	{r}	{q, r}
r	{s}	{p}
s*	ϕ	{p}

	0	1
{p}	{q, s}	{q}
{q}*	{r}	{q, r}
{q, s}*	{r}	{p, q, r}
{r}	{s}	{p}
{q, r}*	{r, s}	{p, q, r}
{p, q, r}*	{q, r, s}	{p, q, r}
{s}*	ϕ	{p}
{r, s}*	{s}	{p}
{q, r, s}*	{r, s}	{p, q, r}
ϕ	ϕ	ϕ

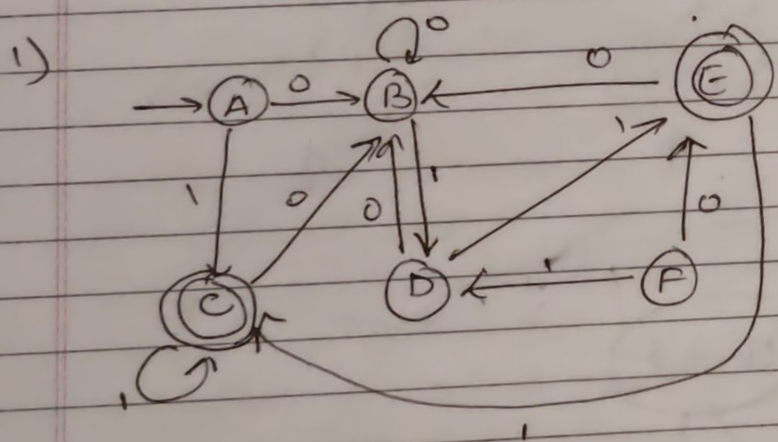


3)	0	1
P	{p, q}	{p}
{p, q}	{p, q, r, s}	{p, t}
{p, q, r, s}	{p, q, r, s}	{p, t}
{p, t}	{p, q}	{p}

Box Method (Myhill Nerode Theorem)

1. Remove all states which are not reachable from start.
2. Create tables such that rows or columns can be identified by state numbers.
3. All those cells in which 1 state is final & other is non-final should be marked (1).

1)



Teacher's Signature:..................

22/8/24

Step 1:- Eliminate F (not reachable)

2:- Table:-

Q\E	0	1
→A	B	C
B	B	D
C*	B	C
D	B	E
E*	B	C

[Omit first row for col]
[Omit las row for rows]

3.

B	2			
*C	1	1		
D	X	2	1	
*E	1	1	X	1
	A	B	C*	D

→ one final one nonfinal mark 1

4. a is input {p, q} state
{p, a} = r
{q, a} = r] No decision

if {p, a} = r
{q, a} = s
check {r, s} if marked then {p, q} as 2

like A, 0 → B
B, 0 → B

A, 1 → C
B, 1 → D

A, 0 - B
D, 0 - B]

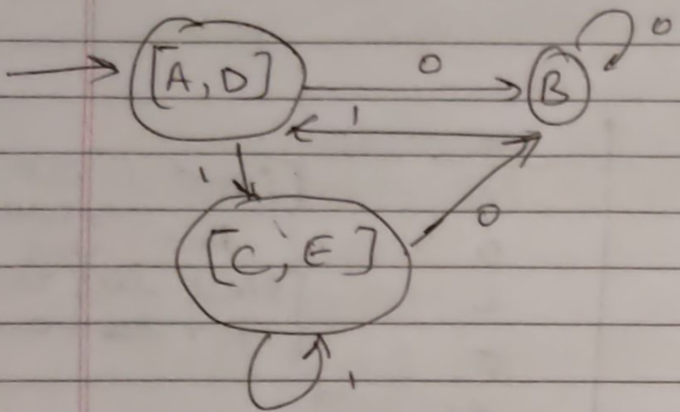
A, 1 → C
D, 1 → E] Not marked

(A, D) ⟹ (C, E) → list them together → May Merge

Teacher's Signature:........................

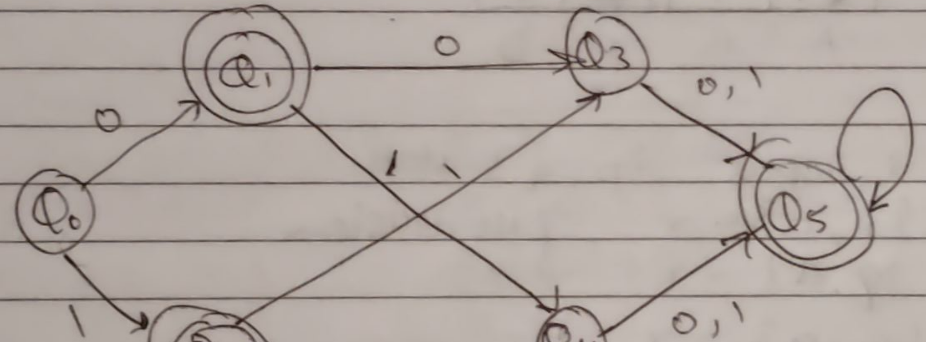
B, 0 → B ✗   B, 1 − D

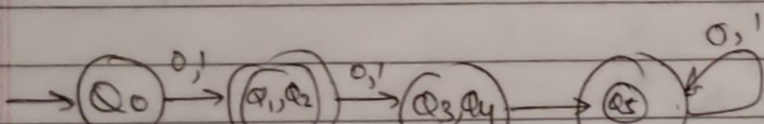
D, 0 → B   B, 1 = E



	0	1
→ A,D	B	C,E
C,E *	B	C, E
B	B	A, D

~~HW~~ 2)





States	Inputs 0	1
Q_0	Q_1, Q_2	$Q_1 Q_2$
$Q_1 Q_2$	$Q_3 Q_4$	$Q_3 Q_4$
$Q_3 Q_4$	Q_5	Q_5
Q_5	Q_5	Q_5

FLAT

24/08/24 FA with output : Moore & Mealy Measures.

- No final states
1. Moore → output depends on each state
2. Mealy → output Associated with Transitions.

Mealy Machines. $(Q, \Sigma, O, \delta, X, q_0)$

Q → Finite set of states

Σ → Input

O → output

δ → input transition function $\delta = Q \times \Sigma \to Q$

X → output " " " " $X : Q \times \Sigma \to O$

q_0 → initial state

1. Design a mealy machine to output x, if the i/p ends in 101, otherwise y over $\Sigma = \{0,1\}$

→

Step 1. Theory.

Step 2. Logic

$M = (Q, \Sigma, O, \delta, X, q_0)$

$\Sigma = \{0,1\}$ $\quad \delta = Q \times \Sigma \to Q$

$O = \{x, y\}$ $\quad q_0 = \{q_s\}$

$X = Q \times \Sigma \to O$ $\quad Q = \{ \ldots \}$

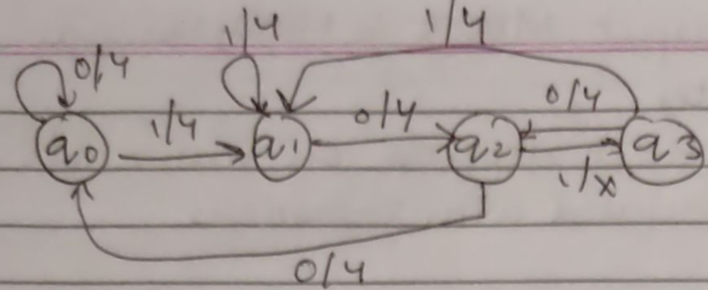
3: Transition Table. !!

$\delta \Rightarrow Q \backslash \Sigma$	0	1
→ q_s	q_0 y	q_1 y
0 q_0	q_0 y	q_1 y
1 q_1	q_2 y	q_1 y
10 q_2	q_0 y	q_3 x
101 q_3	q_2 y	q_1 y

] Merge

MAF $X_1 : Q \times \Sigma \to O$

	0	1
→ 0 q_0	y	y
1 q_1	y	y
10 q_2	y	x
101 q_3	y	y

4. 

5.

$\delta(q_0, 101101)$	$\delta(q_0, 110100)$
$\delta(q_1, 01101)$ 4	$\delta(q_1, 10100)$ 4
$\delta(q_2, 1101)$ 44	$\delta(q_1, 0100)$ 44
$\delta(q_3, 101)$ 44x	$\delta(q_2, 100)$ 444
$\delta(q_1, 01)$ 44x4	$\delta(q_3, 00)$ 444x
$\delta(q_2, 1)$ 44x44	$\delta(q_2, 0)$ 444x4
$\delta(q_3, \epsilon)$ 44x44x	$\delta(q_3, \varepsilon)$ 444x44

2) Design a mealy M/C to change each occurrence of abb to aba

abbb

→ Table:-

	a	b	
→ q_s	q_0	q_1	— merg
a q_0	q_0	q_2	
b q_1	q_0	q_1	— merg
ab q_2	q_0	q_3*	
abb q_3	q_0	q_1	

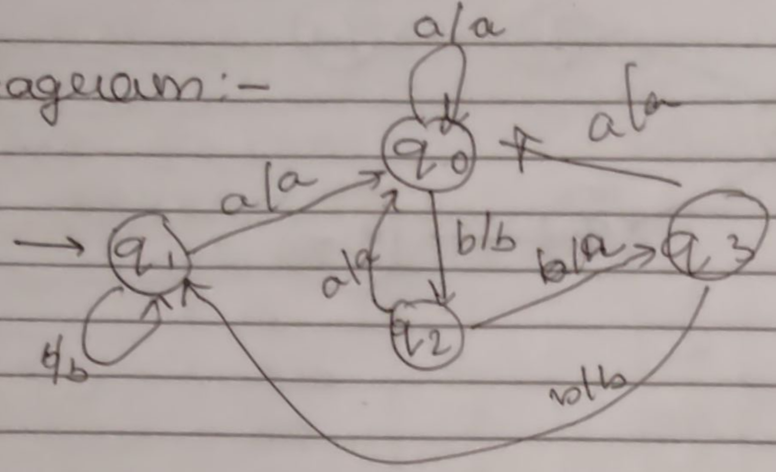
a	q_0	q_0	q_2
b →	q_1	q_0	q_1
ab	q_2	q_0	q_3
abb	q_3	q_0	q_1

Sirf yaha change baki jo input wohi output

MAF

	a	b
q_0	a	~~b~~
q_1	a	b
q_2	a	a
q_3	a	b

Diagram:-



Example.

$\delta(q_1, ababb)$

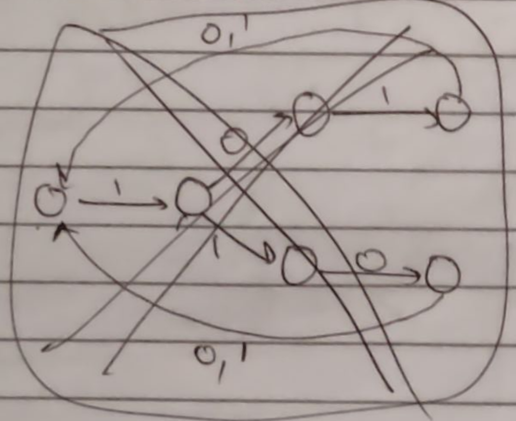
$(q_0, babb)\ a$

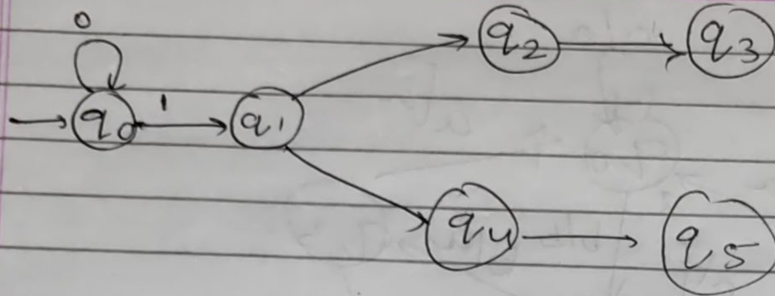
$(q_2, abb)\ ab$

(q_0, bb).

3) binary i/p sequence such that if it has substring 110, m/c should output A if it has 101, m/c should output B otherwise C.

→




0
1
q0
q1
q2
q3
q4
q5

Moore Machine → state

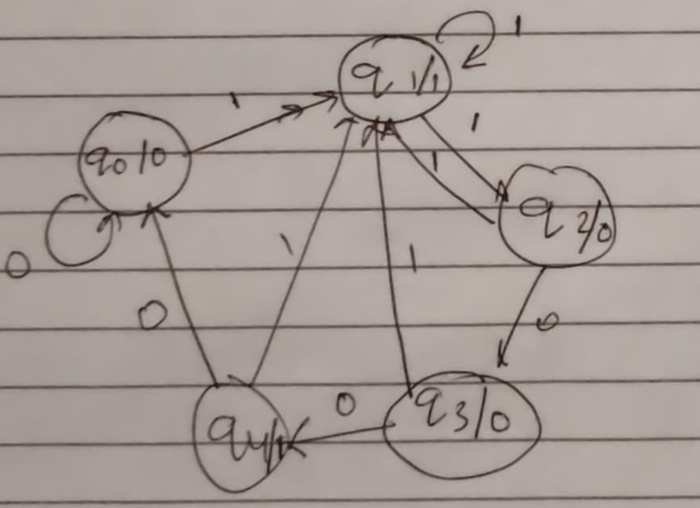
n+1 → output

1) each 1000 → 1001 $\Sigma = \{0, 1\}$
2) 〃 120 → 122 $\{0, 1, 2\}$
3) 〃 aab $\Sigma (a, b\}$

1)

	0	1	λ
q_s	q_0	q_1	0
0 q_0	q_0	q_1	$\lambda(q_0) = 0$
1 q_1	q_2	q_1	$\lambda(q_1) = 1$
10 q_2	q_3	q_1	$\lambda(q_2) = 0$
100 q_3	q_4	q_1	$\lambda(q_3) = 0$
$\boxed{1000}$ q_4	q_0	q_1	$\lambda(q_4) = 1$

	0	1	λ
0 q_0	q_0	q_1	0
1 q_1	q_2	q_1	1
10 q_2	q_3	q_1	0
100 q_3	q_4	q_1	0
1000 q_4	q_0	q_1	1



$\delta(q_s, 101101)$ 0
$\delta(q_1, 01101)$ 01
$\delta(q_2, 1101)$ 010
$\delta(q_1, 101)$ 0101
$\delta(q_1, 01)$ 01011
$(q_2, 1)$ 010110
(q_1, ϵ) 0101101

No change