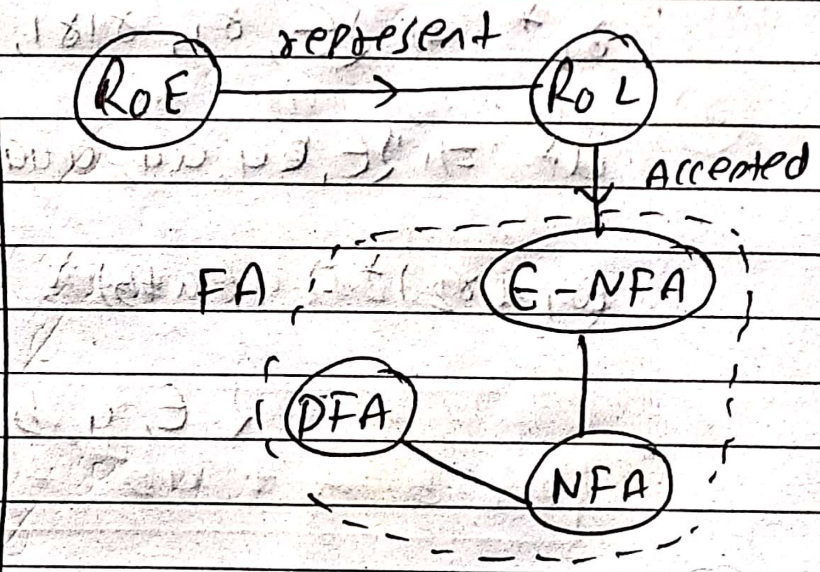
Unit-3 Credits goes to Bibek Dahal

Regular Expressions

→ Regular expression is a algebric notation that represent regular language. The language accepted by finite Automata.

It's uses are:-
1. Tokenization
2. Searching
3. Validation.



It's types are:-

① Basic Regular Exp

$\epsilon \Rightarrow L(\epsilon)$
$= \{\epsilon\}$

$\phi \Rightarrow L(\phi)$
$= \{\ \}$

$a \Rightarrow L(a)$
$= \{a\}$

Let r_1 and r_2 be regular expression

operators :-

	Precedence
ⓐ Union (+)	low
ⓑ Concatenation (·)	Mid
ⓒ Kleen closure (*)	high

Let, $r_1 = a$ and $r_2 = b$

$(r_1 + r_2)$ is also R.E

$r_1 + r_2 \Rightarrow L(r_1 + r_2) = \{r_1, r_2\}$

$r_1 \cdot r_2 \Rightarrow L(r_1 \cdot r_2) = \{r_1 \cdot r_2\}$

$r_1^* \Rightarrow \{\epsilon, r_1, r_1r_1, r_1r_1r_1\}$

$a^* \Rightarrow \{\epsilon, a, aa, aaa, ----\}$

$(r_1 + r_2)^* \Rightarrow (a+b)^*$

$= \{\epsilon, a, b, aa, bb, aba, bab, ---\}$

Q write a Regular Expression over the given alphabet $\Sigma = \{a, b\}$

(1) set of all string ending with ab

$L = \{ab, aab, bab, aaab, bbab ---\}$

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

Date
Page

(2) Set of all String Starting with ab

L = {ab, aba, abb, abab, abba, ---- }

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

(3) set of all string containing exactly 3 'a'

L = {aaa, baaa, aaab, abaa, aaba, aaab ---}

$$\boxed{r = b^* a b^* a b^* a b^*}$$

(4) set of all string exactly two length

L = {aa ab, ba, bb}

$$\boxed{r = aa + ab + ba + bb}$$

(5) set of all string even length

L = {aa, bb, ab, ba, aaaa, bbbb, abaa, --- }

$$\boxed{r = (aa + ab + ba + bb)^*}$$

Date
Page

(6) Set of all string containing even number of 'a'

$L = \{aa, aab, baa, aba, aaab, baab, --- \}$

$$\boxed{r = b^* + (b^* a b^* a b^*)^*}$$

(7) Set of all string containing odd number of 'a'

$L = \{a, ba, ab, bba, bab, abb, baaa, --- \}$

$$\boxed{r = b^* a + ab^* + (b^* a b^* a b^*)^* . a}$$

(8) Set of all string ~~to~~ starting with ab and ending with even length

$L = \{ab, abba, abbb, abaa, abab, --- \}$

$$\boxed{r = ab ((a+b) . (a+b))^*}$$

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

Date
Page

(9) set of all string containing at least three 'a'

$L = \{aaa, aaab, aaba, abaa, baaa, aaaa, ----\}$

$$\boxed{r = ((ba^*)(ba^*)(ba^*))^*}$$

(10) set of all string containing at most three 'a'

$L = \{a, aa, aaa, b, ba, bba, bbb, ----\}$

$$\boxed{r = b^* + b^*ab^* + b^*ab^*ab^* + b^*ab^*ab^*ab^*}$$

(11) set of all string containing atleast three length

$L = \{aaa, bbb, ababab, bbbaaab, ----\}$

$$\boxed{r = (a+b)^3 (a+b)^*}$$

CamScanner

(12) Set of all strings containing atmost three length.

$L = \{a, bb, aba, bab, aaa, ---\}$

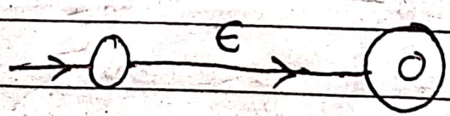
$$\boxed{r = \epsilon + (a+b) + (a+b)(a+b) + (a+b)(a+b)(a+b)}$$

Conversion of RoE into FoA

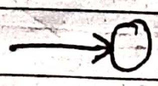
$\text{RoE} \xrightarrow[\text{rep}]{} \text{RoL} \longrightarrow \underbrace{\epsilon\text{-NFA, NFA, DFA}}_{\text{FA}}$

RoE	FoA

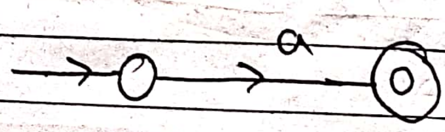
ϵ



ϕ



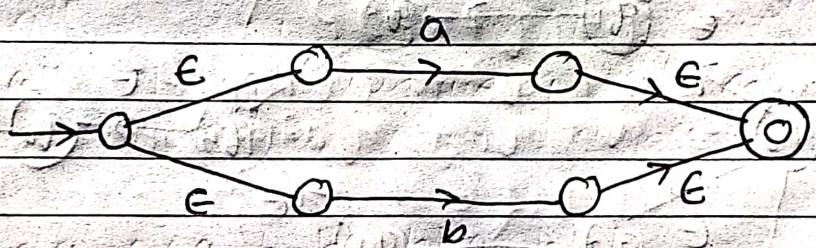
a



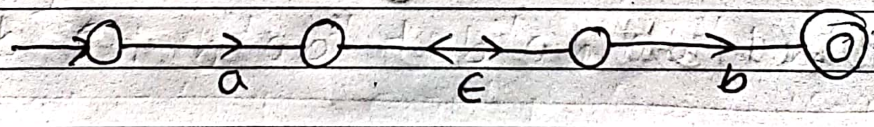
a → {a}   b → {b}

a+b   a·b   a*

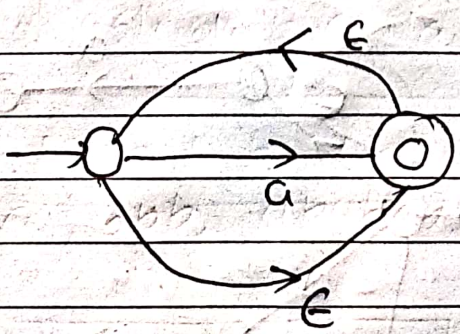
① a+b



② a·b

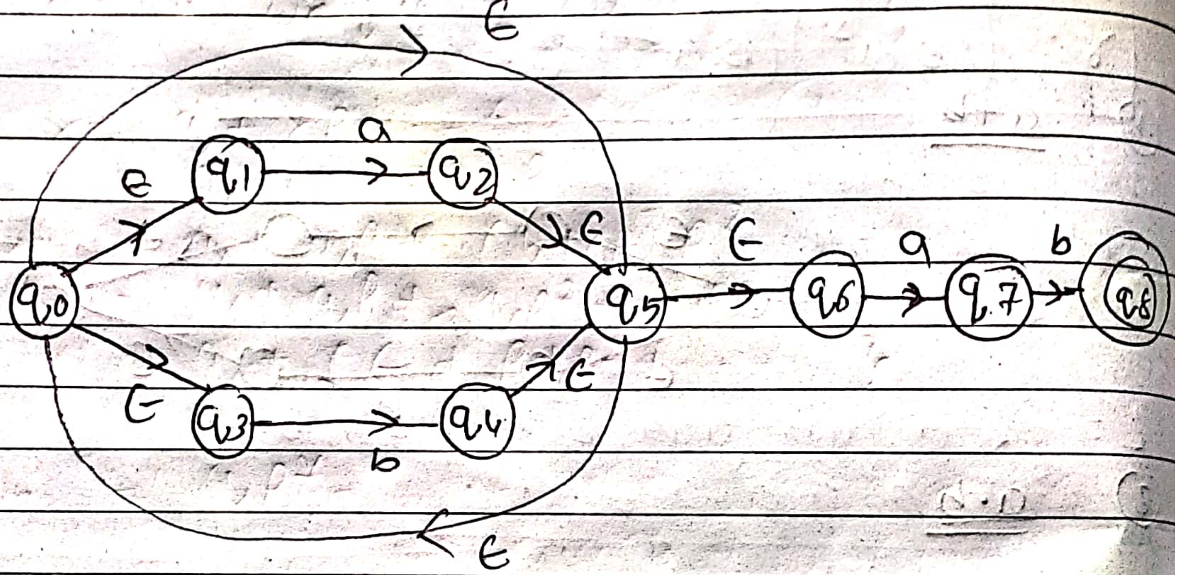


③ a*

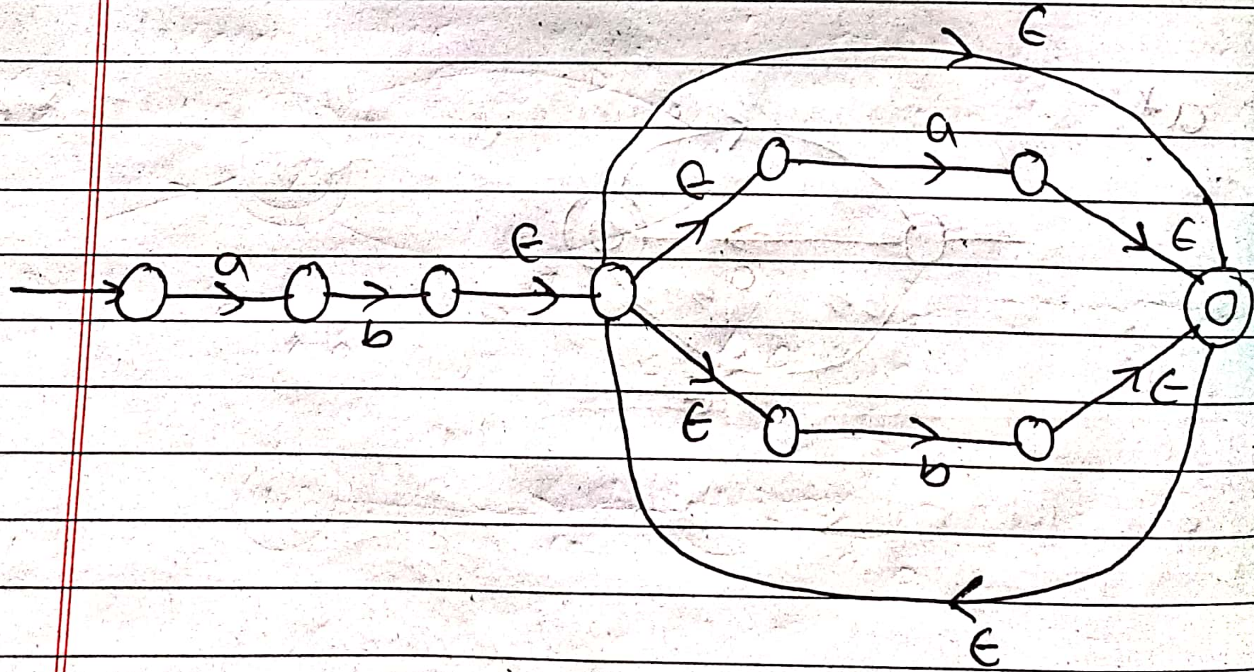


Convert RoE into FA

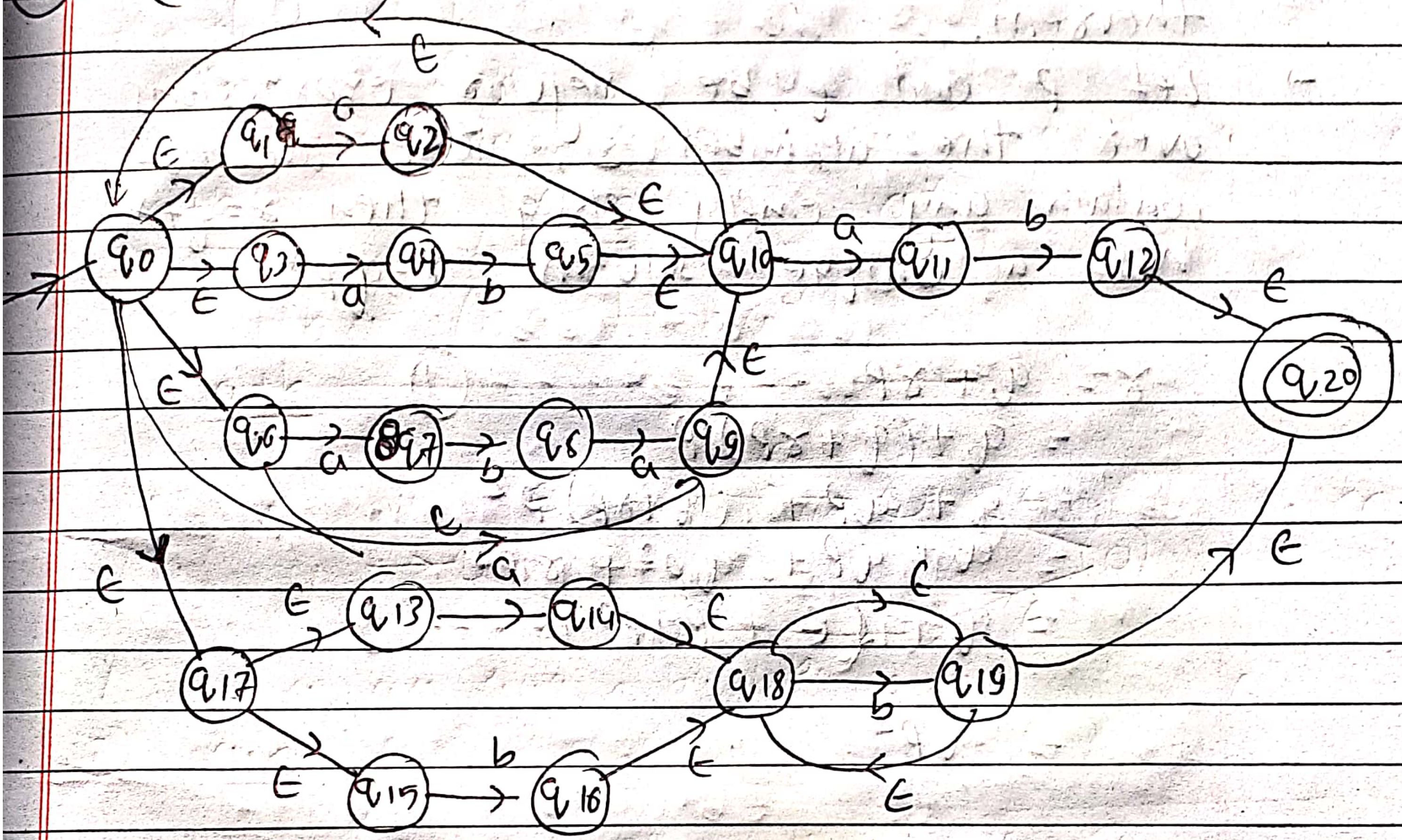
① $(a+b)^* ab$



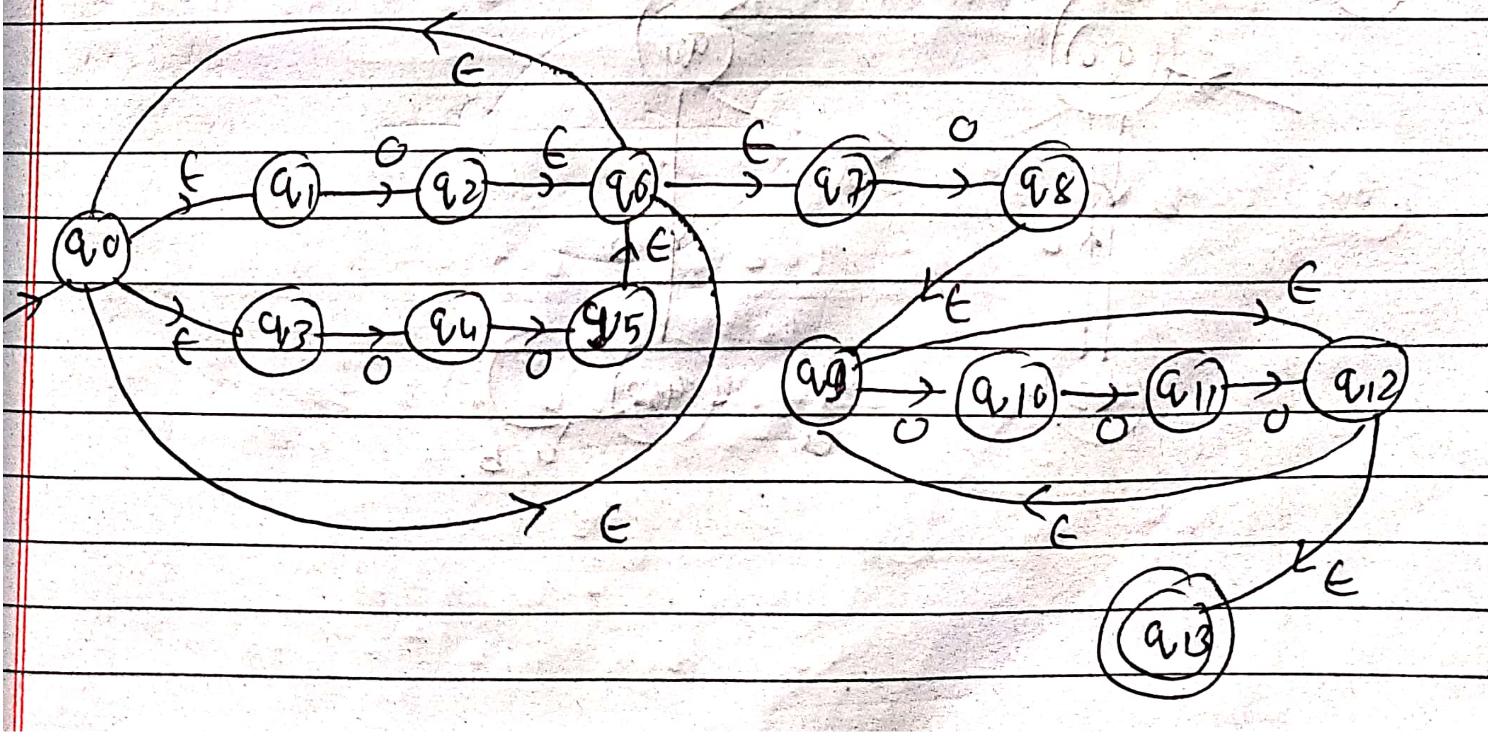
② $ab \cdot (a+b)^*$



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



4) $(0+00)^* 0(000)^*$



Conversion of FA into R.E using Arden's theorem.

→ Let p and q be regular expressions over the alphabet Σ, if p does not contain any empty string then $r = q + rp$ has a unique solution $r = qp^*$

$r = q + rp$ ------------ (1)

$= q + (q + rp)p$

$= q + qp + (q + rp)p^2$

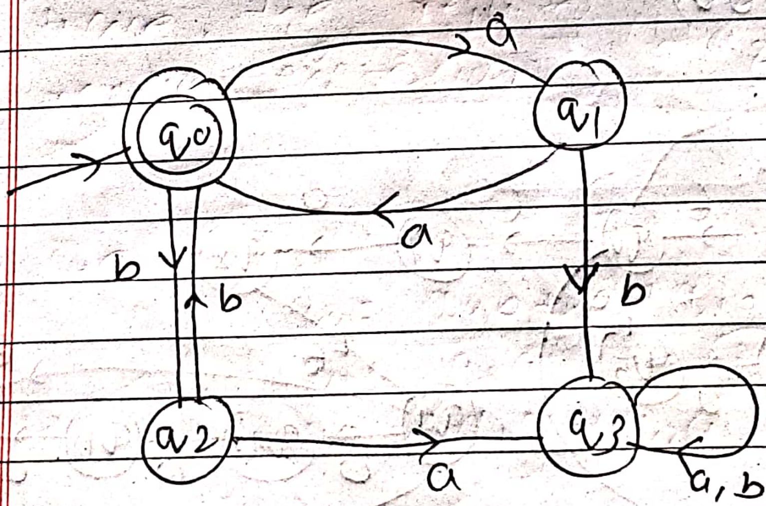
$= q + qp + qp^2 + rp^3$

$= q + \{\epsilon + p + p^2 + \cdots\}$

$= qp^*$

• ϵ - Should not be transmitted.

(1)



Date
Page

$q_0 = q_1 a + q_2 b + \epsilon$ ---- (1)
$q_1 = q_0 a$ ----- (2)
$q_2 = q_0 b$ ----- (3)

$q_3 = q_1 b + q_2 a + q_3 a + q_3 b$ ---- (4)

Putting the value of q_1 and q_2 in eq1 (1) we get,

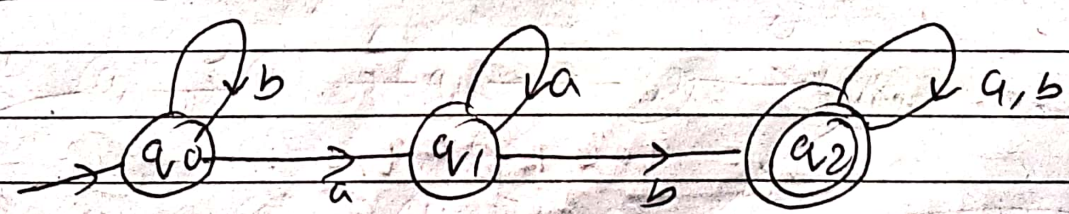
$q_0 = q_0 aa + q_0 bb + \epsilon$
$q_0 = \epsilon + q_0 (aa+bb)$ [∵ Arden's rule]

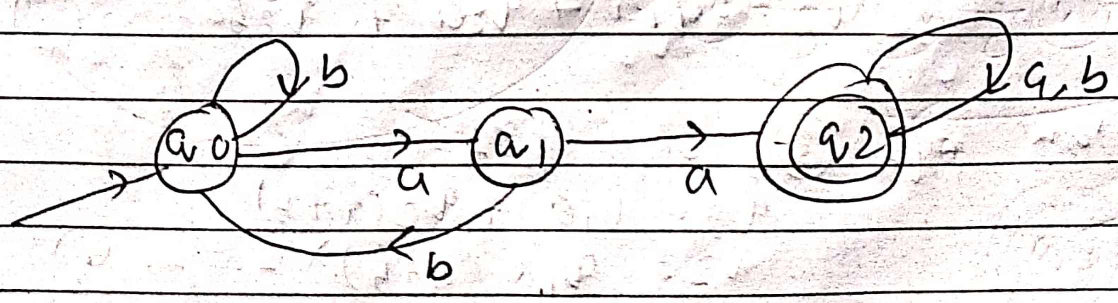
$r = q + rp$

$q_0 = \epsilon (aa+bb)^*$

Here, q_0 is a final state. So required Regular expression (R.E) is $(aa+bb)^*$.

Homework

(1)


(2)


① we have,

$$q_0 = q_0 b + \epsilon \quad \text{---- ①}$$
$$q_1 = q_0 a + q_1 a \quad \text{---- ②}$$
$$q_2 = q_1 b + q_2 a + q_2 b \quad \text{----- ③}$$

From eqn ① we get

$$q_0 = \epsilon + q_0 b$$
$$q_0 = \epsilon b^*$$

Substituting the value of q_0 in eqn ②

$$q_1 = \epsilon b^* a + q_1 a$$
$$q_1 = \epsilon b^* a a^*$$

Substituting the value of q_1 in eqn ③

$$q_2 = \epsilon b^* a a^* b + q_2 (a+b)$$
$$q_2 = \epsilon b^* a a^* b (a+b)^*$$

Hence, the required expression for regular expression is $\epsilon b^* a a^* b (a+b)^*$

② we have,

$$q_0 = q_0 b + \epsilon + q_1 b \quad \text{---- ①}$$
$$q_1 = q_0 a \quad \text{--- ②}$$
$$q_2 = q_1 a + q_2 a + q_2 b \quad \text{--- ③}$$

From eqn ③ we get

$$q_2 = q_1 a + q_2 (a+b)$$
$$q_2 = q_1 a (a+b)^* \quad \text{---- ④}$$

from eq^n (1) and (2)

$$q_0 = q_0 b + \epsilon + q_0 ab$$
$$q_0 = \epsilon + q_0 (b + ab)$$
$$q_0 = \epsilon (b + ab)^*$$

Then, we get eq^n (2) as,

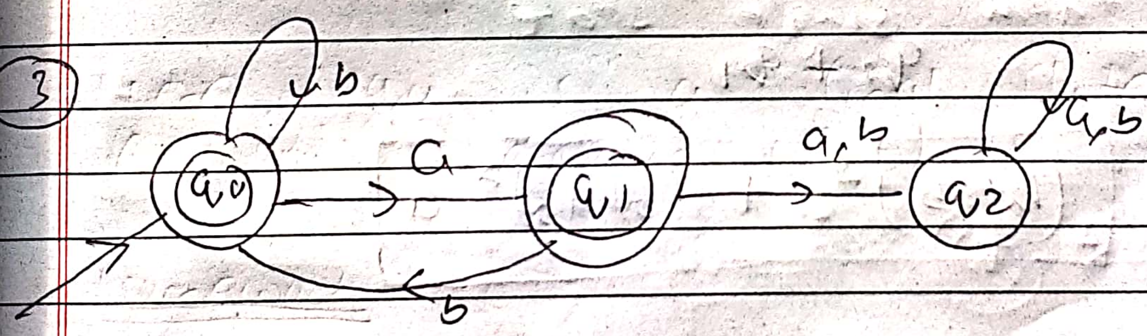
$$q_1 = \epsilon (b + ab)^* a$$

substituting the value of q_1 in eq^n (4)

$$q_2 = \epsilon (b+ab)^* aa (a+b)^*$$

where $\epsilon (b+ab)^* aa (a+b)^*$ is the required R.E

(3)



$q_0 = q_0 b + q_1 b + \epsilon$ ---- (1)

$q_1 = q_0 a$ ---- (2)

$q_2 = q_1 (a+b) + q_2 (a+b)$ ---- (3)

Putting the value of q_1 in eq^n (1)

$$q_0 = q_0 b + (q_0 a) b + \epsilon$$

$q_0 = q_0 b + q_0 ab + \epsilon$

$q_0 = \epsilon + q_0(b+ab)$

$(r = q + rp)$

Applying Arden's rule.

$= (b+ab)^*$

Putting the value of q_0 in eq (2)

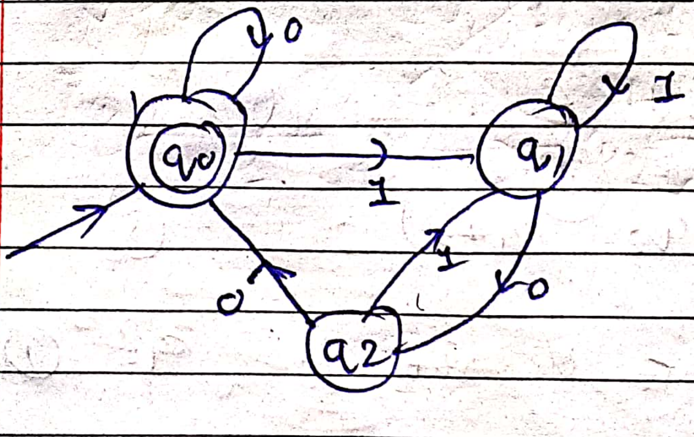
$q_1 = (b+ab)^* a$

Here q_0 and q_1 are final state so required R.E is

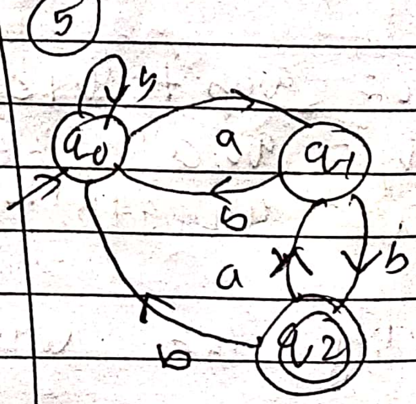
$q_0 + q_1$

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

(4)



(5)



$q_0 = q_0.0 + q_2.0 + \epsilon$ ---- (I)
$q_1 = q_0.1 + q_1.1 + \cancel{q_2.1} + q_2.1$ - (II)
$q_2 = q_1.0$ ---- (III)

Now, $q_0 = q_0.0 + q_1.00 + \epsilon$ (eqn (III) in (I))
$q_1 = q_0.1 + q_1.1 + q_1.01$ (eqn (III) in (II))
$\Rightarrow q_1 = q_0.1 + q_1(1+01)$ $(r = q + rP)$
$\Rightarrow q_1 = q_0.1\ (1+01)^*$ ---- (IV)

$q_0 = q_0.0 + q_0.1(1+01)^*.00$ (using (IV))
$\Rightarrow q_0 = \epsilon + (0 + 1(1+01)^*.00)\ q_0$

$\Rightarrow \boxed{q_0 = (0 + 1(1+01)^*.00)^*}$ Ans.

(5) $q_0 = q_0b + q_1b + q_2b + \epsilon$ ---- (I)
$q_1 = q_0a + q_2.a$ ---- (II)
$q_2 = q_1b$ ---- (III)

Now,
$q_1 = q_0a + q_1ba$
$\Rightarrow q_1 = q_0a(ba)^*$ ---- (IV)

$q_0 = q_0b + q_0a(ba)^*b + q_0a(ba)^*bb + \epsilon$
$\Rightarrow q_0 = \epsilon + q_0(b + a(ba)^*b + a(ba)^*bb)$
$\Rightarrow q_0 = (b + a(ba)^*b + a(ba)^*bb)^*$

$q_2 = q_0a(ba)^*b$
$= (b + a(ba)^*b + a(ba)^*bb)^*a(ba)^*b$
Ans.

Minimization of DFA

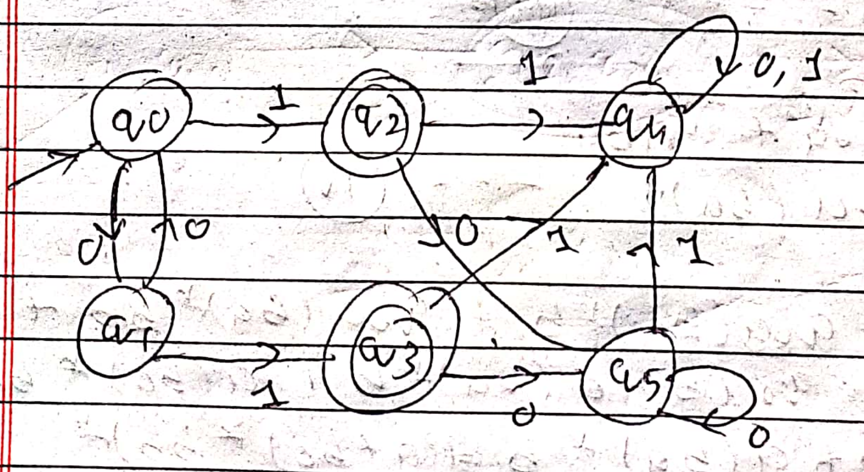
~~Myill~~ Myhill-Nerode theorem

1. Draw a table for all pair of states {P, Q}

2. Mark all pairs $P \in F$ and $Q \notin F$

3. If there are any unmarked pair {P, Q} such that $\{\delta(P,x), \delta(Q,x)\}$ is marked then mark {P, Q}, where x is an input symbol. Repeat this until no more marking can be made.

4. Combine all the unmarked pair and make them single state



F/NF

q_0
q_1
q_2
q_3
q_4
q_5

	q_0	q_1	q_2	q_3	q_4	q_5
q_0						
q_1						
q_2	✓	✓				
q_3	✓	✓				
q_4	✓	✓	✓	✓		
q_5	✓	✓	✓	✓		

q_1 q_4

$\{q_0, q_1\}$

$\delta(q_0, 0) = q_1$ | $\delta(q_0, 1) = q_2$ $\delta(q, 0$

$\delta(q_1, 0) = q_0$ | $\delta(q_1, 1)$ q_3 $\delta($

$\{q_2, q_3\}$

$\delta(q_2, 0) = q_5$ | $\delta(q_2, 1) = q_4$

$\delta(q_3, 0) = q_5$ | $\delta(q_3, 1) = q_4$

$\{q_0, q_4\}$

$\delta(q_0, 0) = q_1$ | $\delta(q_0, 1) = q_2$

$\delta(q_4, 0) = q_4$ | $\delta(q_4, 1) = q_4$

$\{q_1, q_4\}$

$\delta(q_1, 0) = q_0$ | $\delta(q_1, 1) = q_3$

$\delta(q_4, 0) = q_4$ | $\delta(q_4, 1) = q_4$

$\{q_0, q_5\}$

$\delta(q_0, 0) = q_1$ | $\delta(q_0, 1) = q_2$

$\delta(q_5, 0) = q_5$ | $\delta(q_5, 1) = q_4$

$\{q_1, q_5\}$

$\delta(q_1, 0) = q_0$ | $\delta(q_1, 1) = q_3$

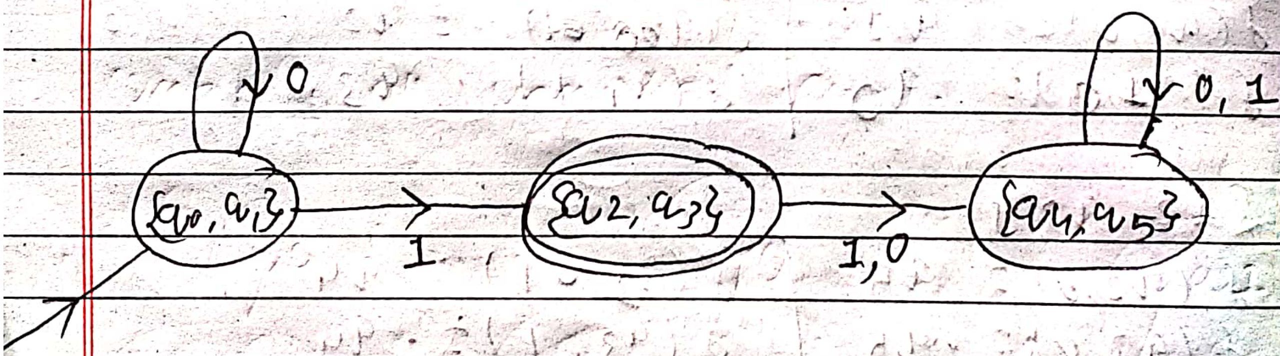
$\delta(q_5, 0) = q_5$ | $\delta(q_5, 1) = q_4$

$\{q_4, q_5\}$

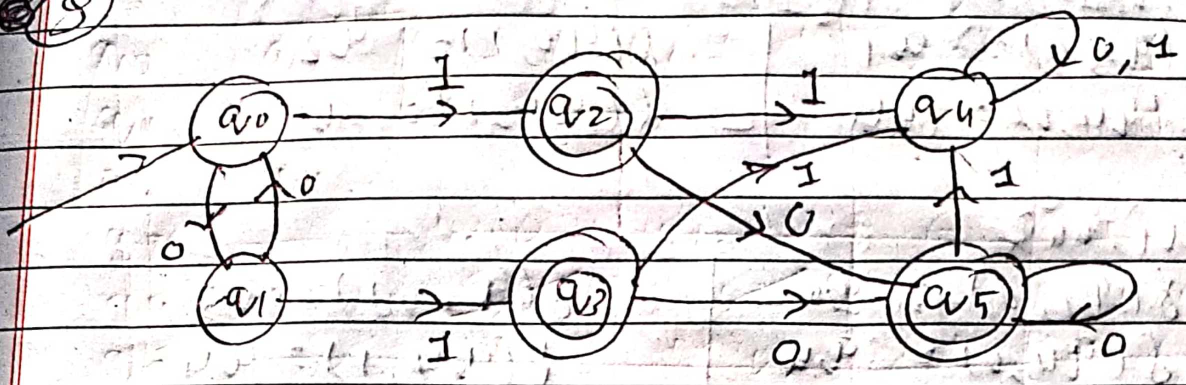
$\delta(q_4, 0) = q_4$ | $\delta(q_4, 1) = q_4$

$\delta(q_5, 0) = q_5$ | $\delta(q_5, 1) = q_4$

combining all unmarked pair



Q9



	q0	q1	q2	q3	q4	q5
q0						
q1						
q2	✓	✓				
q3	✓	✓				
q4	✓	✓	✓	✓		
q5	✓	✓			✓	

$\{q_0, q_1\}$

$\delta(q_0, 0) = q_1$	$\delta(q_0, 1) = q_2$
$\delta(q_1, 0) = q_0$	$\delta(q_1, 1) = q_3$

$\{q_2, q_3\}$

$\delta(q_2, 0) = q_5$	$\delta(q_2, 1) = q_4$
$\delta(q_3, 0) = q_5$	$\delta(q_3, 1) = q_4$

$\{q_0, q_4\}$

$\delta(q_0, 0) = q_1$ | $\delta(q_0, 1) = q_2$

$\delta(q_4, 0) = q_4$ | $\delta(q_4, 1) = q_4$

$\{q_1, q_4\}$

$\delta(q_1, 0) = q_0$ | $\delta(q_1, 1) = q_3$

$\delta(q_4, 0) = q_4$ | $\delta(q_4, 1) = q_4$

$\{q_2, q_5\}$

$\delta(q_2, 0) = q_5$ | $\delta(q_2, 1) = q_4$

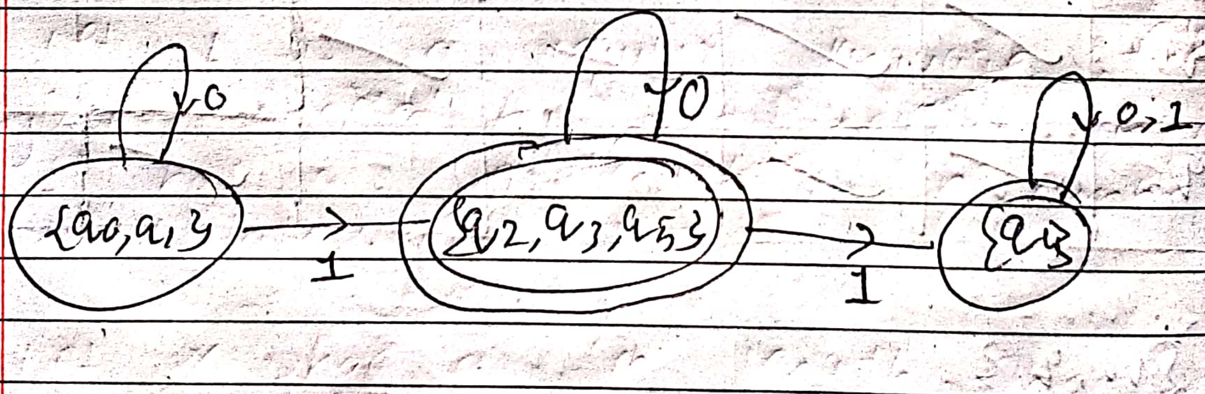
$\delta(q_5, 0) = q_5$ | $\delta(q_5, 1) = q_4$

$\{q_3, q_5\}$

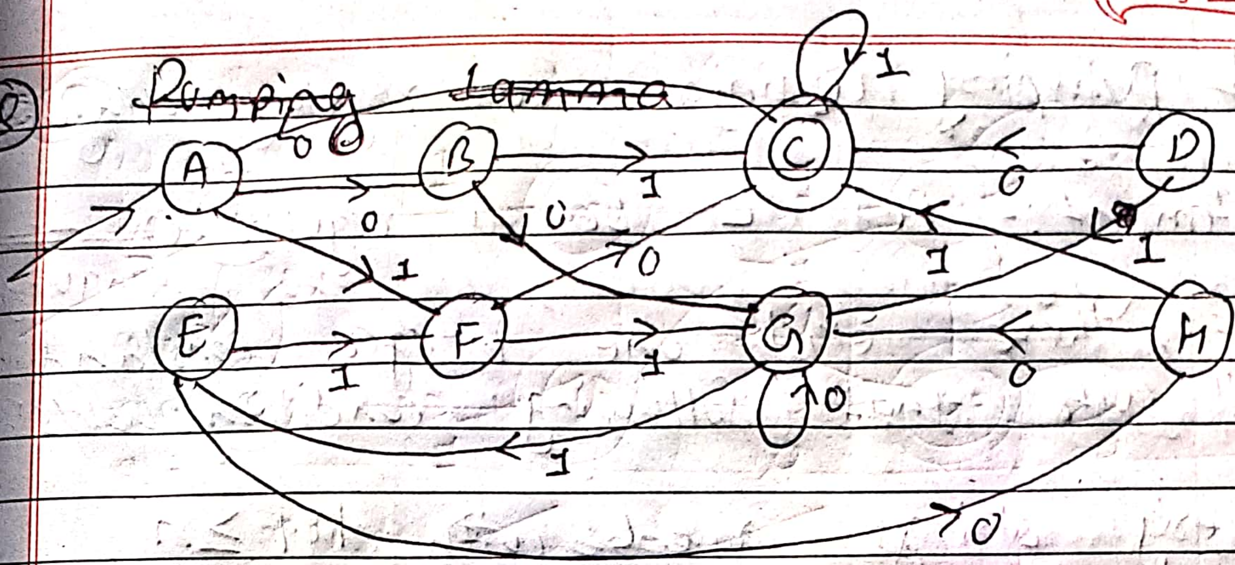
$\delta(q_3, 0) = q_5$ | $\delta(q_3, 1) = q_4$

$\delta(q_5, 0) = q_5$ | $\delta(q_5, 1) = q_4$

combining all unmarked pairs.



Q ~~Pumping Lamma~~



state	0	1
A	B	F
B	G	C
* ~~C~~	~~A~~	~~C~~
~~D~~	~~C~~	~~G~~
~~E~~	~~H~~ B	~~F~~
F	C	G
G	G	~~E~~ A
~~H~~	~~G~~	~~C~~

→ Minimization of DFA using table Partition method

→ Remove unreachable state & Trap state

→ Divide given TT into Final & Non-Final

→ Remove Common states. (H)

→ Replace H with 'B'

Final state

state	0	1
C	A	C

→ combine Non Final & Final

State	0	1
A	B	F
B	G	C
C	A	C
F	C	G
G	G	A

* This is the ~~minized~~ minimized DFA table.

→ To prove not a regular language

Pumping lemma for regular Language

Statement :- Let L be a regular ~~expression~~ Language

Then, there exist one integer constant "n" such that following condition holds :-

n ⇒ number of states

Every strings $x \in L$ with $|x| \geq n$, can be written as $x = uvw$, such that :-

u, v, w ⇒ breakdown of string

1. $|v| > 0$ or $v \neq \epsilon$
2. $|uv| \leq n$
3. $uv^k w \in L \ \forall \ k \geq 0$

k ⇒ number of pumping

Proof :-

Let L is a regular language, that is ~~then~~ accepted by DFA ~~accepted by the regular language~~. It has a 'm' length, 'n' number of states.

Let x has a 'm' length. Therefore

$\boxed{m \geq n}$

$x = a_1, a_2, \ldots, a_m$

Now,

Pigeonhole Principle $\boxed{0 \le i < j \le n}$

Date ____ Page ____

define.

$\hat{\delta}(q_0, a_1 a_2 \cdots a_m)$

~~$= \delta(\hat{\delta}(q_0, a_1 a_2 \cdots), a_m)$~~

~~$= \delta(\delta$~~

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

$\delta(q_0, a_1) = q_1$

$\delta(q_1, a_2) = q_2$

$\delta(q_2, a_3) = q_3$

$\vdots$

~~$\delta(q_{m-1}, a_n) = q_n$~~

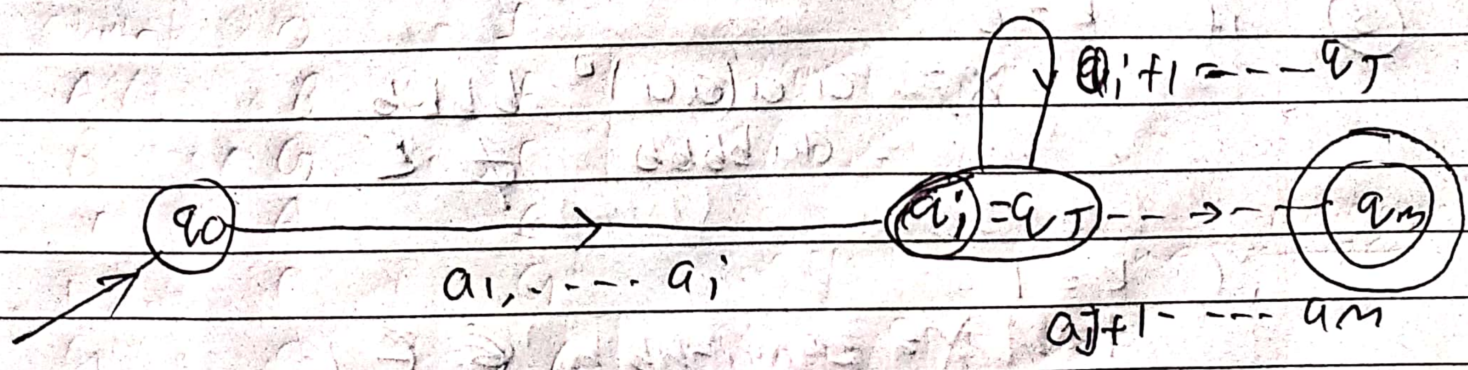
$\delta(q_{n-1}, a_m) = q_n$

$x = a_1 a_2 \cdots a_m = uvw$

$u = a_1 \cdots a_i$

$v = a_{i+1} \cdots a_J$

$w = a_{J+1} \cdots a_m$



① $L = \{a^n b^n \mid n \geq 1\}$ is not a regular language.

$L = \{ab, aabb, aaabbb, aaaabbbb, ----\}$

Suppose given language is a regular

Input string $(X) = aaaabbbb \Rightarrow |m| = 8$

Let, no of state in DFA$(n) = 8$

Case I :-

v is in the a's part

$X = \underbrace{aa}_{u}\ \underbrace{aa}_{v}\ \underbrace{bbbb}_{w}$

Checking conditions :-

(1) $|v| = 2 > 0$ (True)

(2) $|uv| = |aaaa| = 4 \leq n$ (True)

(3) if $k=0$

$X = aa(aa)^0 bbbb$

$= aabbbb \notin L$

if $k=1$

$X = aaaabbbb \in L$

if $k=2$

$X = aaaaaabbbb \notin L$

As third condition is violated. The given language is not a regular language.

② $L = \{a^n \mid n \text{ is a prime}\}$ is not RGL

$L = \{aa, aaa, aaaaa, aaaaaaa, \ldots\}$

Suppose given language is a regular

Input string $(x) = aaaaa \Rightarrow |m| = 5$

let no. of state in DFA $(n) = 5$ (it take equal or less)

case 1 :-

v is in the a's mid part

$x = \underbrace{aaa}_{u}\ \underbrace{a}_{v}\ \underbrace{a\ \ a}_{w}$

checking conditions.

① $|v| = 1 > 0$ (True)

② $|uv| = |aaa| = 3 \le n$ (True)

③ if $k=0$

$$X = aa(a)^0 aa$$
$$= aaaa \notin L$$

As third condition is violated. It is not a regular language.

③ $L = \{a^n a b^{2n} \mid n \geq 1\}$ is not RGL

$L = \{aabb, aaabbbb, aaaabbbbbb \ldots\}$

Suppose given language is regular

Input String $(x) =$ aaabbbb $|m| = 7$

let no. of states of DFA $(n) = 7$

Case I:-

v is in the b's part

$$X = \underbrace{a\ a\ a}_{u}\ \underbrace{b\ b}_{v}\ \underbrace{b\ b}_{w}$$

Checking conditions.

① $|v| = 2 > 0$ (True)

② $|uv| = 5 \leq n$ (True)

Date
Page

(3) if $k=0$

$$x = aaa(bb)^0 bb$$
$$= aaabb \notin L$$

As third conditions is violated, It is not a regular language.

(4) $L = \{ww \mid w \in (0,1)^*\}$ is not RGL

$L = \{\epsilon, 00, 11, 0101, 1010, ----\}$

Suppose given language is regular.

Input string $(x) = 0101$ $\quad |m| = 4$

Let no of states of DFA $(n) = 4$

Let v is in the mid part

$$X = \underbrace{0}_{u}\ \underbrace{10}_{v}\ \underbrace{1}_{w}$$

Checking conditions.

(1) $|v| = 2 > 0$ (True)

(2) $|uv| = 3 \le n$ (True)

(3) if $k=0$

$$x = 0(10)^0 1$$
$$= 01 \notin L$$

As third condition is violated ∴ It is not a regular language.

Finite automata with output.

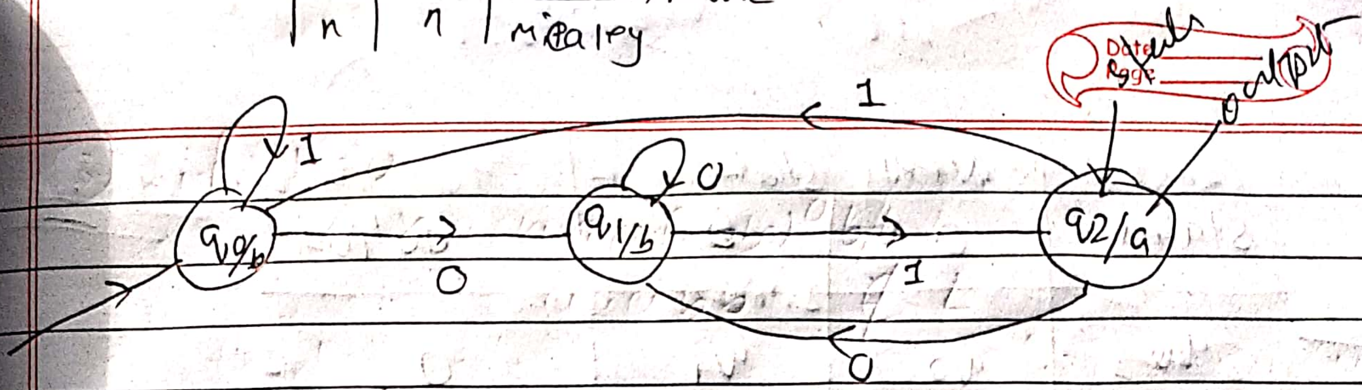
6 tuples.

* Moore machine

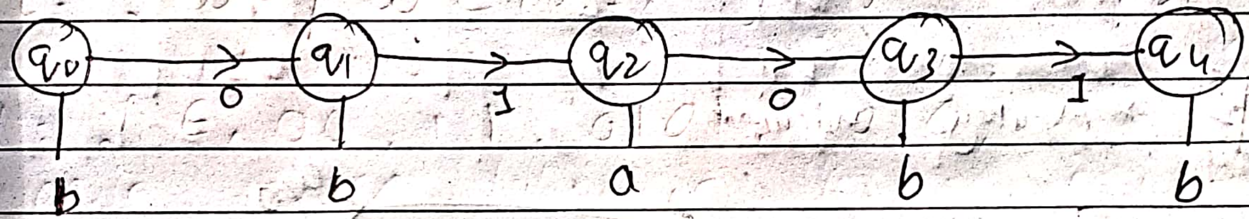
q_0: Initial state
Σ: Input alphabet
δ: Transition function $[Q \times \Sigma \rightarrow Q]$
Q: Set of states
O: Output alphabet
λ: $Q \rightarrow O$ // $Q \times \Sigma \rightarrow O$
(Moore) ~~Millie~~ (Mealy)

(1) Construct a moore machine that prints 'a' whenever sequence of 01 encountered in the input strings.

Input | n | n+1 | output → moore
n | n | mealey

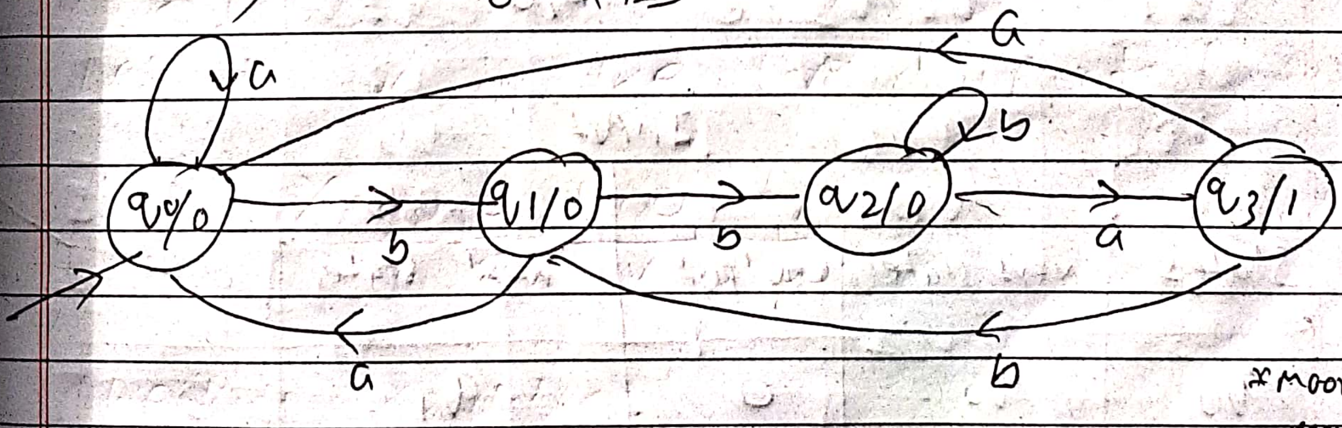


0101 (n) = n+1
(Input) (output)



② Construct a moore machine that counts occurrance of substring bba in input strings

$\Sigma = \{a, b\}$
$O = \{0, 1\}$

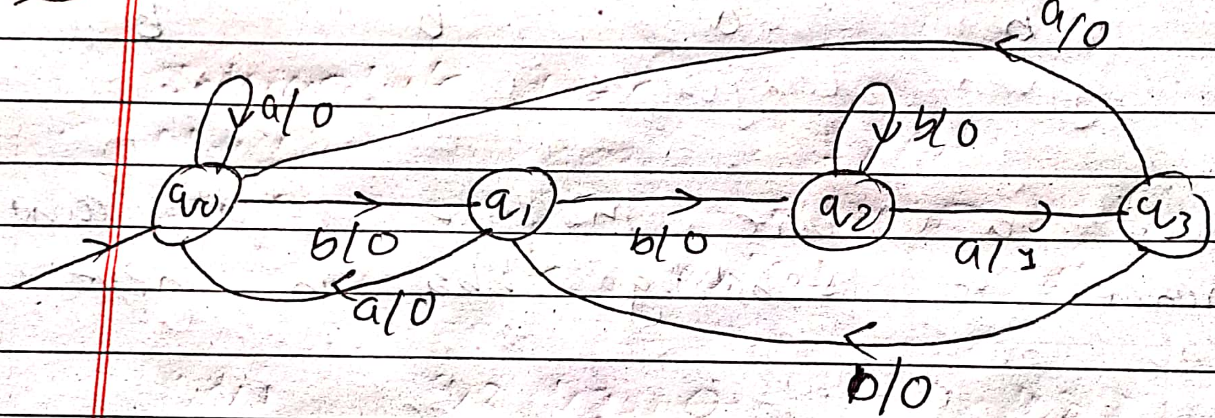


* moore machine.

Now constructing transition table;

States	Next state a	b	output
q0	q0	q1	0
q1	q0	q2	0
q2	q3	q2	0
q3	q0	q1	1

* Mealey machine



	Input			
	a		b	
States	Next state	output	next state	output
q0	q0	0	q1	0
q1	q0	0	q2	0
q2	q3	1	q2	0
q3	q0	0	q1	0

from Moore

Date
Page

⇒ Conversion to Mealy.

States	a		b	
	state	output	state	output
q0	q0	0	q1	0
q1	q0	0	q2	0
q2	q3	1	q2	0
q3	q0	0	q1	0

⇒ Conversion from Mealy to moore

States	0		1	
	state	output	state	output
q0	q0	0	q1	0
q1	q0	0	q2	0
q2	q3	1	q2	0
q3	q0	0	q1	0

states	Next state		output
	0	1	
q0	q0	q1	0
q1	q0	q2	0
q2	q3	q2	0
q3	q0	q1	0

transition → state

Convert Mealy Machine to Moore Machine

Mealy

States	Input 0		Input 1	
	state	output	state	output
q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

q_1:

q_2: $q_2(0)$, $q_2(1)$

q_3:

q_4: $q_4(0)$, $q_4(1)$

Intermediate table :-

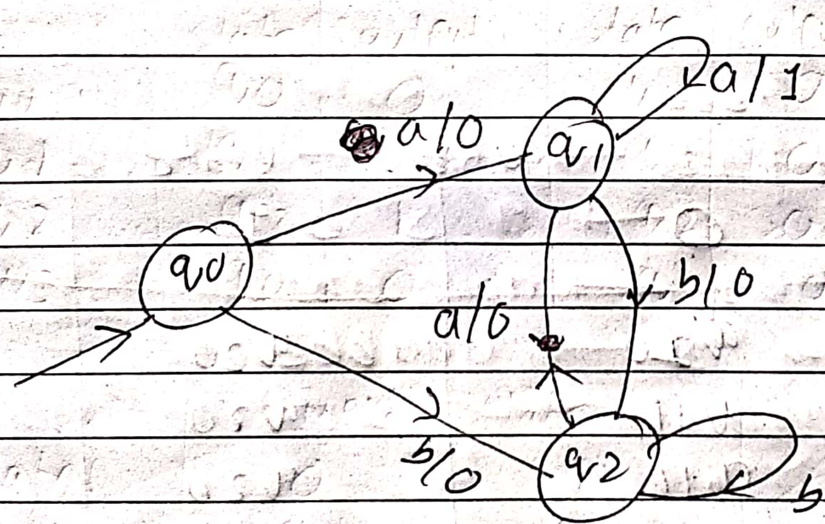
States	Input 0		Input 1	
	state	output	state	output
q_1	q_3	0	q_{20}	0
q_{20}	q_1	1	q_{40}	0
q_{21}	q_1	1	q_{40}	0
q_3	q_{21}	1	q_1	1
q_{40}	q_{41}	1	q_3	0
q_{41}	q_{41}	1	q_3	0

Moore:-

States	Next State 0	1	output
q1	q3	q20	1
q20	q1	q40	0
q21	q1	q40	1
q3	q21	q1	0
q40	q41	q3	0
q41	q41	q3	1

final Moore machine

Q



Mealy to Moore

Table for Mealy.

States	Input a: State	a: output	b: State	b: output
q0	q1	0	q2	0
q1	q1	1	q2	0
q2	q1	0	q2	1

q_0 $q_1 <$ q_{10}, q_{11} $q_2 <$ q_{20}, q_{21}

Intermediate table :-

States	Input			
	0		1	
	State	output	State	output
q_0	q_{10}	0	q_{20}	0
q_{10}	q_{11}	1	q_{20}	0
q_{11}	q_{11}	1	q_{20}	0
q_{20}	q_{10}	0	q_{21}	1
q_{21}	q_{10}	0	q_{21}	1

Moore

State	Next State		output
	0	1	
q_0	q_{10}	q_{20}	0
q_{10}	q_{11}	q_{20}	0
q_{11}	q_{11}	q_{20}	1
q_{20}	q_{10}	q_{21}	0
q_{21}	q_{10}	q_{21}	1

final table

output कस्यरी ??

- q_0 q_1 देन So 100% '0'
- → look three $q_{10} \rightarrow 0$
- → look two $q_{11} \rightarrow 1$