20/1/20

2. REGULAR EXPRESSIONS

Regular expressions are useful for representing certain set of strings in an algebraic fashion. Actually these describe the languages accepted by the finite state Automata

(OR)

The language accepted by finite automata are easily described by simple expressions called regular expressions

→ The Regular expression is the most effective way to represent regular language

→ RE contains 3 operators

1) + (union)
2) . (concatination)
3) * (kleene closure)

→ ϕ is a RE which denotes the empty set.

RE language

$\phi \Rightarrow \{\ \}$

→ ϵ is a RE and it denotes the set $\{\epsilon\}$ & it is a null string

$\epsilon \Rightarrow \{\epsilon\}$

→ For each 'a' in Σ, a is RE and it denotes the set

$a \in \Sigma \Rightarrow \{a\}$

→ ϕ, ϵ, $a \in \Sigma$ are called primitive RE

→ Here r_1, r_2 are regular expressions.

→ $r_1 + r_2$, $r_1 \cdot r_2$, $r_1 * r_2$

→ a-Kleene closure

$a^* = \{\epsilon, a, aa, ---\}$

$a^+ = \{a, aa, aaa, ---\}$

$a^+ = a \cdot a^*$

$(a+b)^* = \{\epsilon, a, b, ab, aa, ba, bb ---\}$

1. Find out a RE whose length is exactly 2.

$L = \{aa, ab, ba, bb\}$

$\Rightarrow aa + ab + ba + bb$

$\Rightarrow a(a+b) + b(a+b)$

$\Rightarrow (a+b)(a+b)$

2) length is exactly 3.

$L = \{aaa, aba, aab, abb, baa, bab, bba, bbb\}$

$(a+b)(a+b)(a+b)$

$\Rightarrow (a+b)(a+b)(a+b)$

3) Find out the RE whose length is atleast 2

$L = \{aa, ab, ba, bb, aaa, ---\}$

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

$* = 0$ to n

4) Atmost 2

$L = \{\epsilon, a, b, aa, ab, ba, bb\}$

$\epsilon + a + b + aa + ab + ba + bb$

$\Rightarrow (a+b+\epsilon)(a+b+\epsilon)$

5) find out RE whose length is even

$L = \{\epsilon, aa, ab, ba, bb, --\}$

L is infinite

$(a+b)^*$

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

Scanned with CamScanner

6) Whose length is odd.

$L = \{a, b, aaa, \ldots\}$

$[(a+b)(a+b)]^{*}(a+b)$

7) Whose length of string is divisible by 3

$L = \{0, 3, 6, 9, 12 \ldots\}$

$\rightarrow [(a+b)(a+b)(a+b)]^{*}$ $*$: 0 to n

8) length of string $\cong$ 2 mod 3

$\Rightarrow (a+b)^{3n+2} \mid n \geq 0$

3) 2 (
2 → rema

$\Rightarrow [(a+b)(a+b)(a+b)]^{*}(a+b)(a+b)$

9) No. of a's are exactly 2 & b are anything

$\rightarrow b^{*}ab^{*}ab^{*}$

10) No. of a's are atleast 2, b are anything

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

11) atmost 2 a's

$\rightarrow b^{*}(\epsilon+a)b^{*}(\epsilon+a)b^{*}$

Identity rules for RE:

1) $\phi + R = R$ (union)

2) $\phi R + R\phi = \phi$ (concat)

3) $\epsilon R = R\epsilon = R$

4) $\epsilon^* = \epsilon$ (and) $\phi^* = \epsilon$

5) $R + R = R$

6) $R^* R^* = R^*$

7) $R R^* = R^* R$

8) $(R^*)^* = R^*$

9) $\epsilon + RR^* = \epsilon + R^*R = R^*$

10) $(PQ)^*P = P(QP)^*$

11) $(P+Q)^* = (P^* + Q^*)^*$

$= (P^* Q^*)^*$

12) $(P+Q)R = PR + QR$ and

$R(P+Q) = RP + RQ$

Algebraic Laws for RE:

let r_1, r_2 & r_3 be the three RE

i) commutative law for union:

$$r_1 + r_2 = r_2 + r_1$$

ii) Associative law for union:

$$(r_1 + r_2) + r_3 = r_1 + (r_2 + r_3)$$

iii) Associative law for concatenation:

$$(r_1 r_2) r_3 = r_1 (r_2 r_3)$$

NOTE:- there is no commutative law for concatination.

iv) Identity for union:-

$$r_1 + \phi = \phi + r_1 = r_1$$

v) Identity for concatination:

$$r_1 \epsilon = \epsilon r_1 = r_1$$

vi) Idempotent law for union:

$$r_1 + r_1 = r_1$$

vii) Annihilator for concatination:

$$\phi r_1 = r_1 \phi = \phi$$

viii) distributive law for concatination:

Concatination is left distributive over union

$$r_1 (r_2 + r_3) = r_1 r_2 + r_1 r_3$$

Concatination is Right distributive Over union

$(r_1+r_2)r_3 = r_1 r_3 + r_2 r_3$

ix) Laws involving closure:

$(r^*)^* = r^*$

$\phi^* = \epsilon$

$\epsilon^* = \epsilon$

$r^+ = r \cdot r^* = r^* \cdot r$

$r^* = r^+ + \epsilon$

Arden's theorem:

If P & Q are two RE over Σ and if P doesnot contain ϵ then the following equation in R given by $R = Q + RP$ has a unique solution i.e,

$R = QP^*$

Sol:- $R = Q + RP \rightarrow (1)$

$= Q + QP^*P$ $\quad R = QP^*$

$= Q(\epsilon + P^*P)$

$= QP^*$ $\quad [\because \epsilon + P^*P = \epsilon + PP^* = P^*]$

Identity

$R = QP^*$

$R = Q + RP$

$= Q + (Q + RP)P$ $\quad (\because R = Q + RP)$

$= Q + QP + RPP$

$= Q + QP + (Q + RP)P^2$

$= Q + QP + QP^2 + QP^3$

$\vdots$

$= Q + QP + QP^2 + QP^3 + \cdots + QP^n + QP^{n+1}$

$= Q + QP + QP^2 + QP^3 + \cdots + QP^n + QP^* P^{n+1}$

$= Q(\epsilon + P + P^2 + P^3 + \cdots +$ $\quad [\because R = QP^*)$

$P^n + P^* P^{n+1})$

$= QP^*$

21|1|20

problems of equivalence of two regular Expressions

1. proove that $(1 + 00^*1) + (1 + 00^*1)(0 + 10^*1)^*$
$(0 + 10^*1)$ is equal to $0^*1(0 + 10^*1)^*$

LHS

$(1 + 00^*1) + (1 + 00^*1)(0 + 10^*1)^*(0 + 10^*1)$

$= (1 + 00^*1)[\epsilon + (0 + 10^*1)^*(0 + 10^*1)^*(0 + 10^*1)]$

$= (1 + 00^*1)(0 + 10^*1)^*$ $\quad (\epsilon + R^*R = R^*)$

Scanned with CamScanner

$\Rightarrow (\epsilon \cdot 1 + 00^*1)(0+10^*1)^*$ $\quad \epsilon R = R$

$= (\epsilon + 00^*)1(0+10^*1)^*$

$= 0^*1(0+10^*1)^*$ $\quad \epsilon + RR^* = R^*$

$=$ RHS

Hence the regular expression are equivalent

2) Show that $(0^*1^*)^* = (0+1)^*$

LHS $\Rightarrow (P^*Q^*)^* = (P+Q)^*$ Identify $(0^*1^*)^*$

$= \{\epsilon, 0, 00, 000, ---, 1, 11, 111, --01, 10, --\}$

$= \{$any combination of 0's, any combination of 1's, any combination of 0, 1, $\epsilon\}$

RHS $\Rightarrow (0+1)^* = \{\epsilon, 0, 00, 1, 11, 01, 10, --\}$

$= \{\epsilon$, any combination of 0's, any combination of 1's, any combination of 0's, 1's$\}$

LHS = RHS is proved.

3) show that $(ab)^* \neq (a^* b^*)^*$

consider, $LHS = (ab)^*$

$= \{\epsilon, ab, abab, ababab, \ldots\}$

$RHS = (a^* b^*)^*$

$= \{\epsilon, a, aa, b, bb, \ldots ab, \ldots\}$

$LHS \neq RHS$

4) show that $(r+s)^* \neq r^* + s^*$

consider $LHS = (r+s)^*$

$= \{\epsilon, r, rr, \ldots s, ss, rs, rsrs, \ldots\}$

$= \{\epsilon$, any combination of r and s$\}$

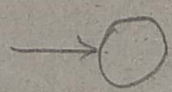
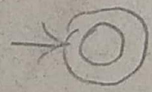
$r^* + s^* = \{\epsilon, r, rr, \ldots s, ss, \ldots\}$

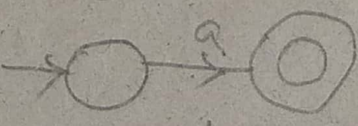
$= \{\epsilon$, any combination of only r or any combination of only s$\}$

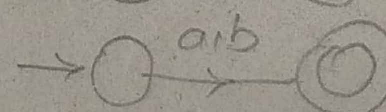
Note: that in RHS there is no combination of r and s together Hence

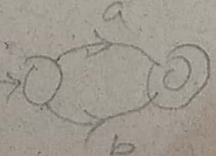
$LHS \neq RHS$ is proved

→ convertion from RE to finite automata

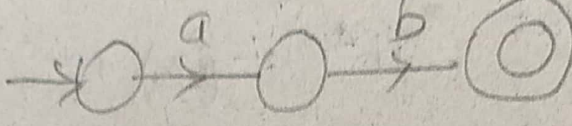
Rules: $\phi \Rightarrow$  $\epsilon \Rightarrow$ 

$a \rightarrow$ 

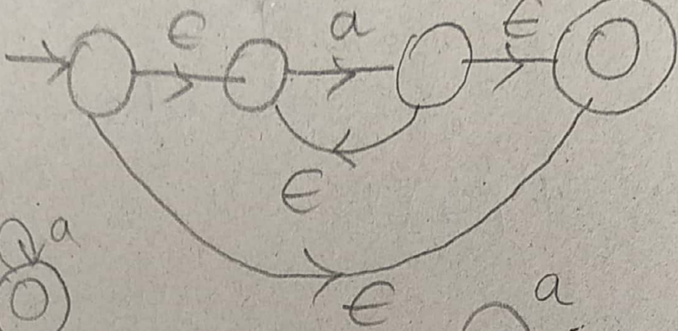
$a+b \Rightarrow$ 



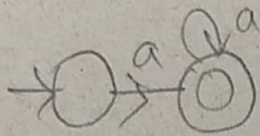
$a \cdot b \Rightarrow$



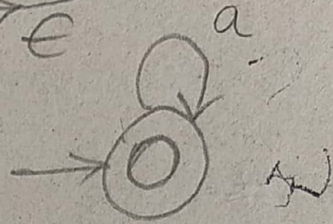
$a^* \Rightarrow$



$a^+ \Rightarrow$

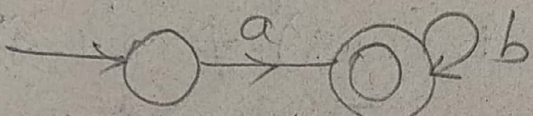


(or)

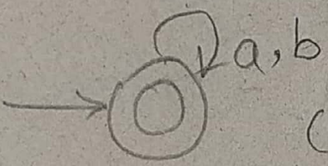


Eg:-

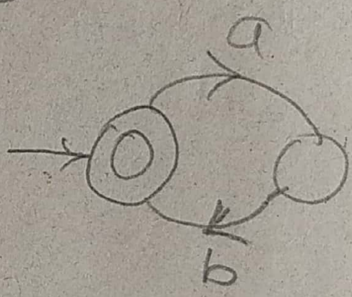
ab^*



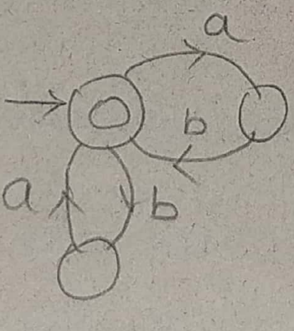
$(ab)^*$



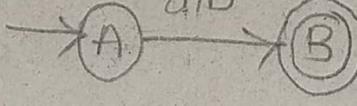
(or)



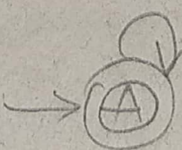
$(ab+ba)^*$



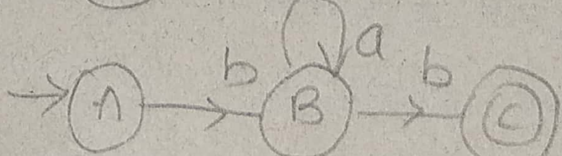
$(a+b)$



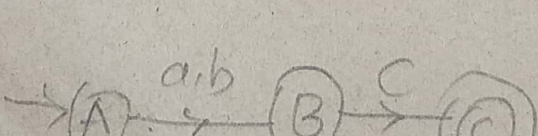
a^*



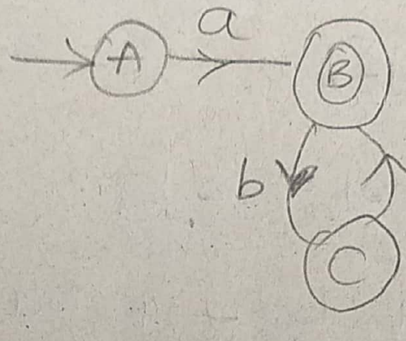
ba^*b



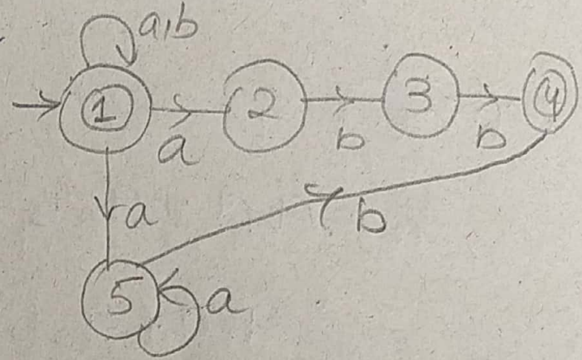
$(a+b)c$



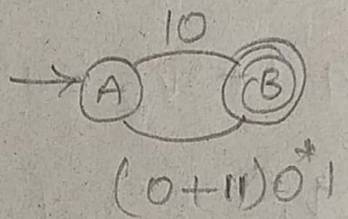
$a(bc)^*$



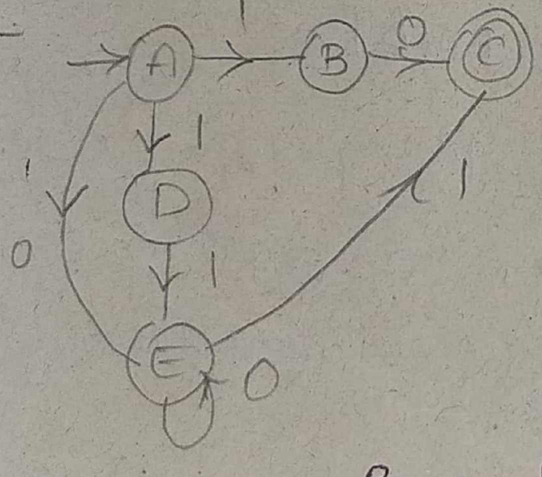
$(a/b)^*(abb/a^+b)$



$10+(0+11)0^*1$



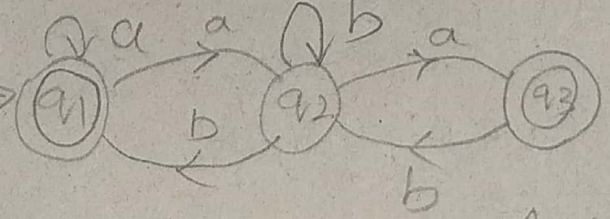
(or)



conversion from FA to RE

NFA:

1.



Incoming edges

$q_3 = q_2 a \rightarrow ①$

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

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

eq → ① $q_3 = q_2 a$

Sub $q_2 \rightarrow$ 2

$= [q_1 a + q_2 b + q_3 b] a$

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

eq $\rightarrow$ 2

$q_2 = q_1 a + q_2 b + q_3 b$

$\Rightarrow q_1 a + q_2 b + [q_2 a] b \quad [\because q_3 = q_2 a]$

$\Rightarrow q_1 a + q_2 b + q_2 ab$

$\Rightarrow q_1 a + q_2 [b + ab]$

$$\underset{R}{\underline{q_2}} = \underset{Q}{\underline{q_1 a}} + \underset{R}{\underline{q_2}} \underset{P}{\underline{[b+ab]}}$$

$R = Q + RP$ then $R = QP^*$

Arden's theorem

$q_2 = (q_1 a)(b+ab)^* \rightarrow$ 5

eq 3: $q_1 = \epsilon + q_1 a + q_2 b$

$q_2 \Rightarrow$ eq 5

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

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

$$\underset{R}{\underline{q_1}} = \underset{Q}{\underline{\epsilon}} + \underset{R}{\underline{q_1}} \underset{P}{\underline{[a + a(b+ab)^*] b}}$$

$$q_1 = \epsilon\left[\left[a+a(b+ab)^* b\right]^*\right.$$

$$= \epsilon\left[\left[a+a(b+ab)^*\right]b\right]^*$$

$$q_1 = \left\{(a+a)(b+ab)^* b\right]^* \rightarrow 6$$

$$[\because \epsilon \cdot R = R]$$

$$q_3 = q_2 a$$

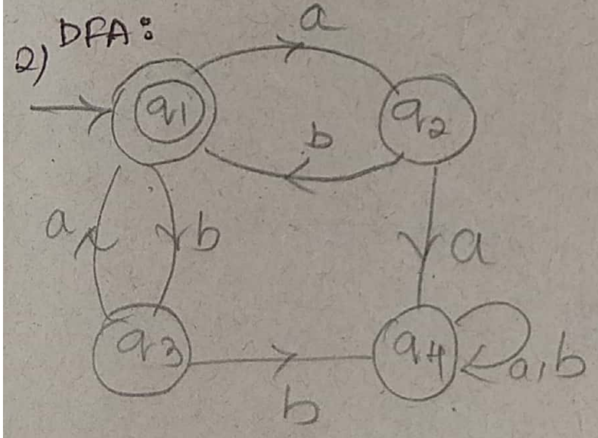
$$= (q_1 a)(b+ab)^* a \quad eq\text{-}5$$

$$q_3 = \left\{\left[a+a(b+ab)^* b\right]^* a\left[b+ab\right]^* a \Rightarrow\right.$$

$$eq\text{-}6$$

$$q_3 = \left[(a+a(b+ab)^*)b\right]^* a(b+ab)^* a.$$

2) DFA:



$$q_1 = \epsilon + q_2 b + q_3 a \rightarrow 1$$

$$q_2 = q_1 a \rightarrow 2$$

$$q_3 = q_1 b \rightarrow 3$$

$$q_4 = q_2 a + q_3 b + q_4 a + q_4 b \rightarrow 4$$

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

q_2 and q_3 eq-(2) & (3)

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

$= \epsilon + q_1 ab + q_1 ba$

$$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \underbrace{(ab + ba)}_{P}$$

$R = Q + RP$

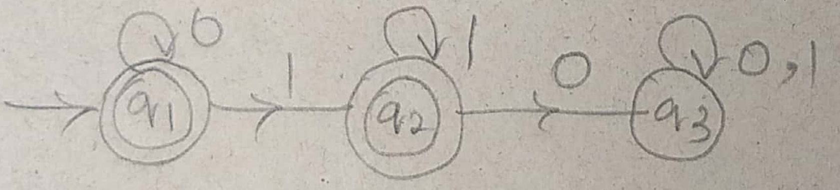
$R = QP^*$

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

$\epsilon \cdot R = R$

$q_1 = (ab + ba)^*$

Find the RE for the following DFA [When there are multiple final states]



$q_1 = \epsilon + q_1 0 \rightarrow (1)$

$q_2 = q_1 + q_2 1 \rightarrow (2)$

$q_3 = q_2 0 + q_3 0 + q_3 1 \rightarrow (3)$

Final state

$$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \underbrace{0}_{P}$$

$$q_1 = \epsilon 0^*$$

$$q_1 = 0^* \rightarrow (4)$$

$q_2 = q_1 1 + q_2 1$ $\qquad q_1 = 0^*$ from eq (4)

$$\underbrace{q_2}_{R} = \underbrace{0^* 1}_{Q} + \underbrace{q_2}_{R} \underbrace{1}_{P}$$

$$q_2 = 0^* 1 (1)^*$$

union of both final states.

$$= q_1 + q_2$$

$$= 0^* + 0^* 1 1^*$$

$$= 0^* [\epsilon + 1 1^*] \qquad E = RR^* = R^*$$

$$R = 0^* 1^*$$

23/1/20.

conversion of FA to RE

State elimination method

This method involves the following steps in finding the RE for the given FA.

Step 1:- The Initial state should not have any incoming edge.

If there exists only incoming edge to the initial state then create a new initial state having no incoming edge to it.

Step2:- The final state of FA should not have any outgoing edge

If there exists any outgoing edge from the final state then create a new final State having no outgoing edge from it with ϵ move.

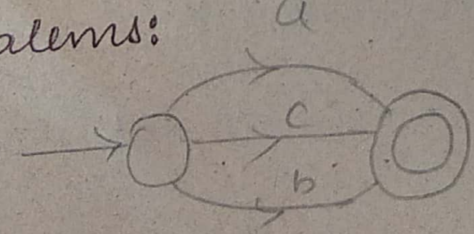
Step 3:-> There must exist only one final state in the FA.

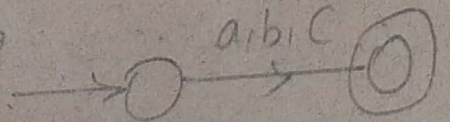
If there exists multiple final states in FA then convert all the final states into non-final states and create a new single final state with ϵ move.

Step-4:- Eliminate all the intermediate States one by one these states may be eliminated in any order.

Other than the initial state and final states eliminate the remaining states one by one

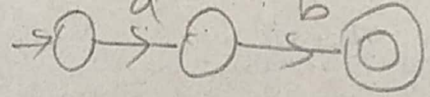
Problems:

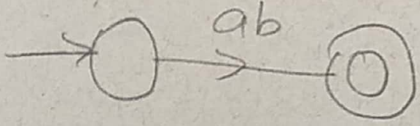
1. 

Sol:-> 

a+b+c

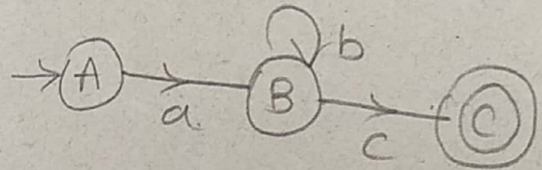
2.

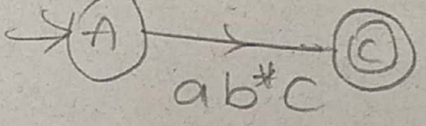




ab

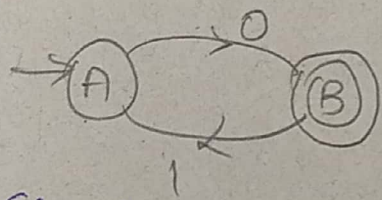
3.



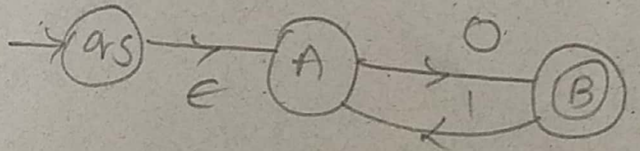


ab^*c

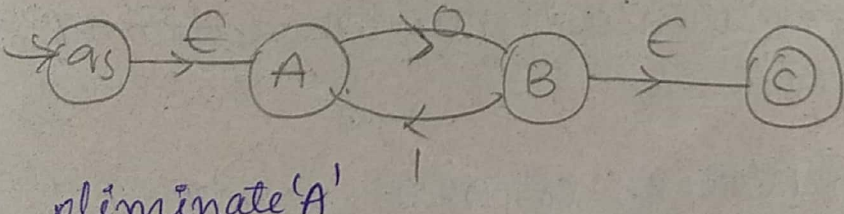
4.



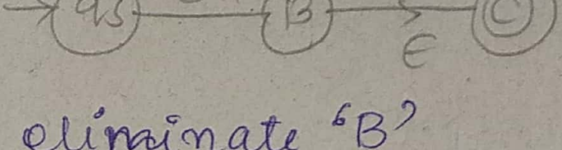
Create a new Initial state with ϵ move



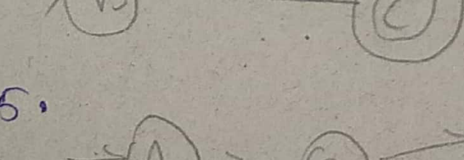
final state having an outgoing edge then create a new final state with ϵ move and make final State as non final state



eliminate 'A'

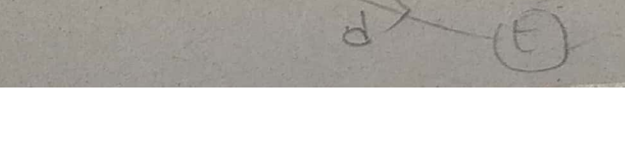


eliminate 'B'

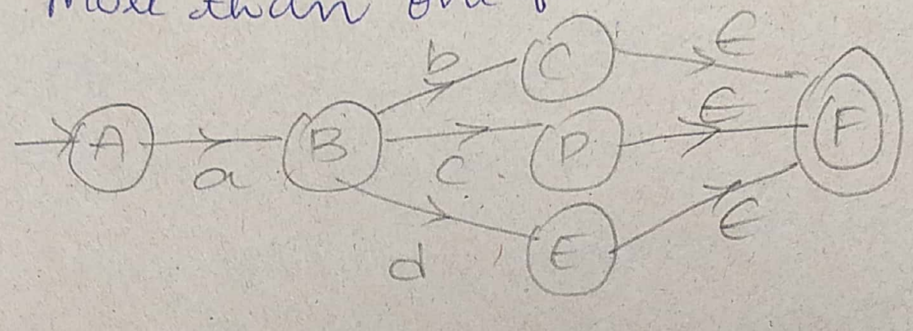


$0(10)^*$

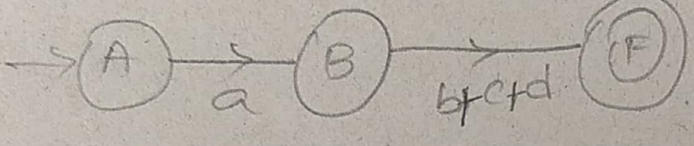
5.

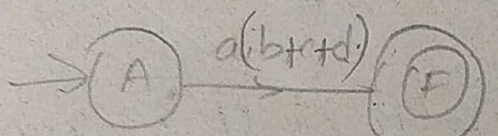


More than one final state



eliminate C, D, E





$a(b+c+d)$

Minimization of FA

For the given FA applying various procedures to decrease the no. of states and constructing a minimal DFA this process is called minimization of DFA.

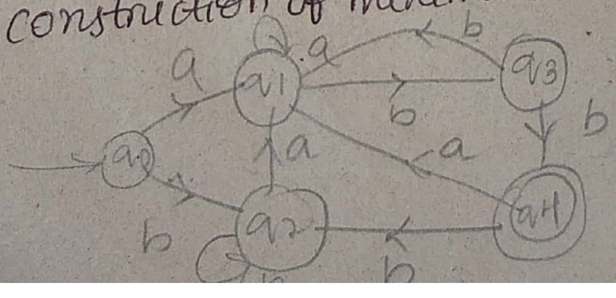
→ Minimization of DFA can be done in various methods-

i) partition method

ii) equivalence method.

partition methods:-

construction of minimal DFA



First is the eliminate or delete the state which is not reachable from initial state but here all are reachable from initial state

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
$*q_4$	q_1	q_2

0 equivalent $[q_0\ q_1\ q_2\ q_3]$ $[q_4]$

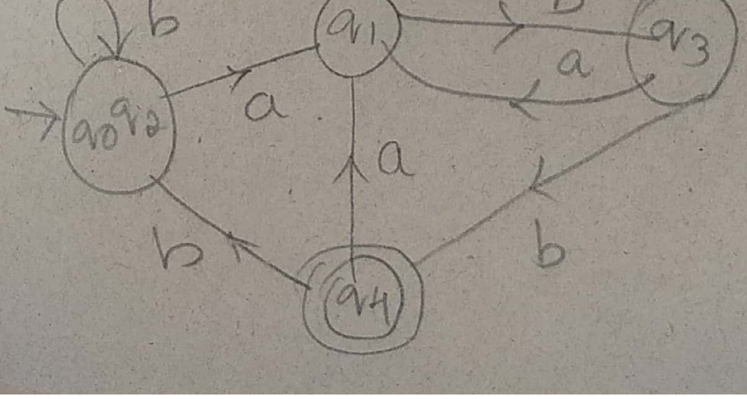
$\downarrow$ all non-final states $\downarrow$ final state

1 equivalent $[q_0\ q_1\ q_2]$ $[q_3]$ $[q_4]$

2 equivalent $[q_0\ q_2]$ $[q_1]$ $[q_3]$ $[q_4]$

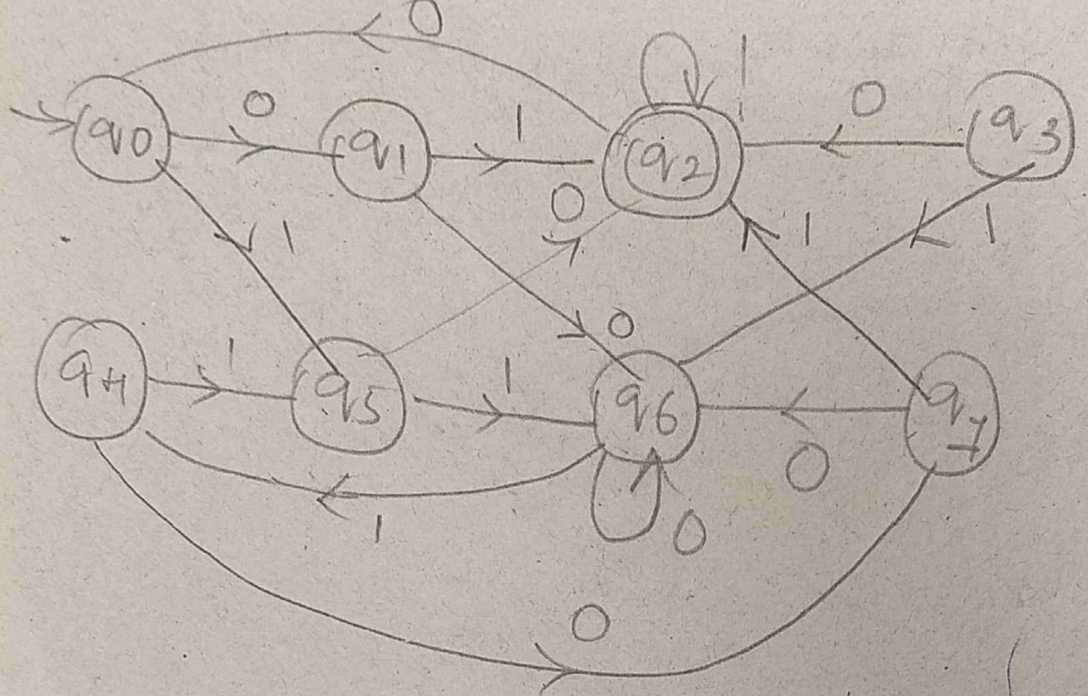
3 equivalent $[q_0\ q_2]$ $[q_1]$ $[q_3]$ $[q_4]$

$[q_0\ q_2]$ are equivalent then replace q_0, q_2 as $[q_0 q_2]$ as single state



Scanned with CamScanner

Q. Construct a minimal DFA for the given DFA.



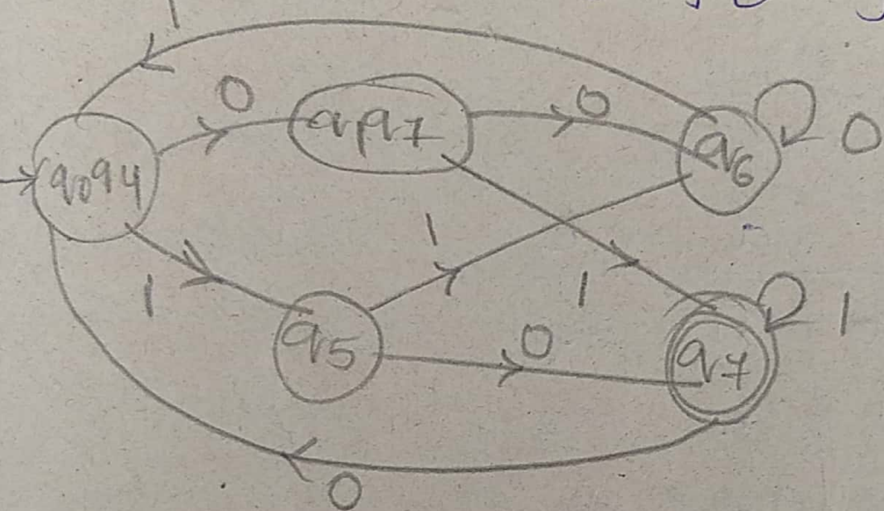
	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2^*	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

0 equivalent [q_0 q_1 q_3 q_4 q_5 q_6 q_7] [q_2]

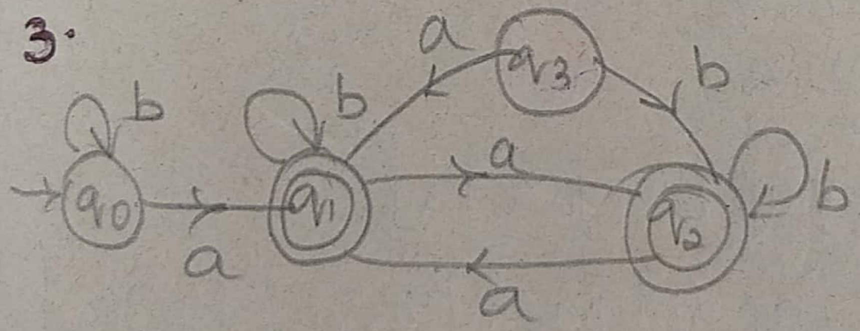
1 equivalent [q_0 q_4 q_6] [q_1 q_7] [q_5] [q_2]

2 equivalent [q_0 q_4] [q_6] [q_1 q_7] [q_5] [q_2]

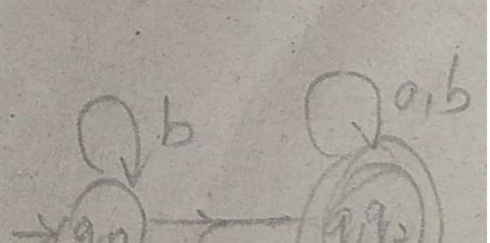
3 equivalent [q_0 q_4] [q_6] [q_1 q_7] [q_5] [q_2]



3.



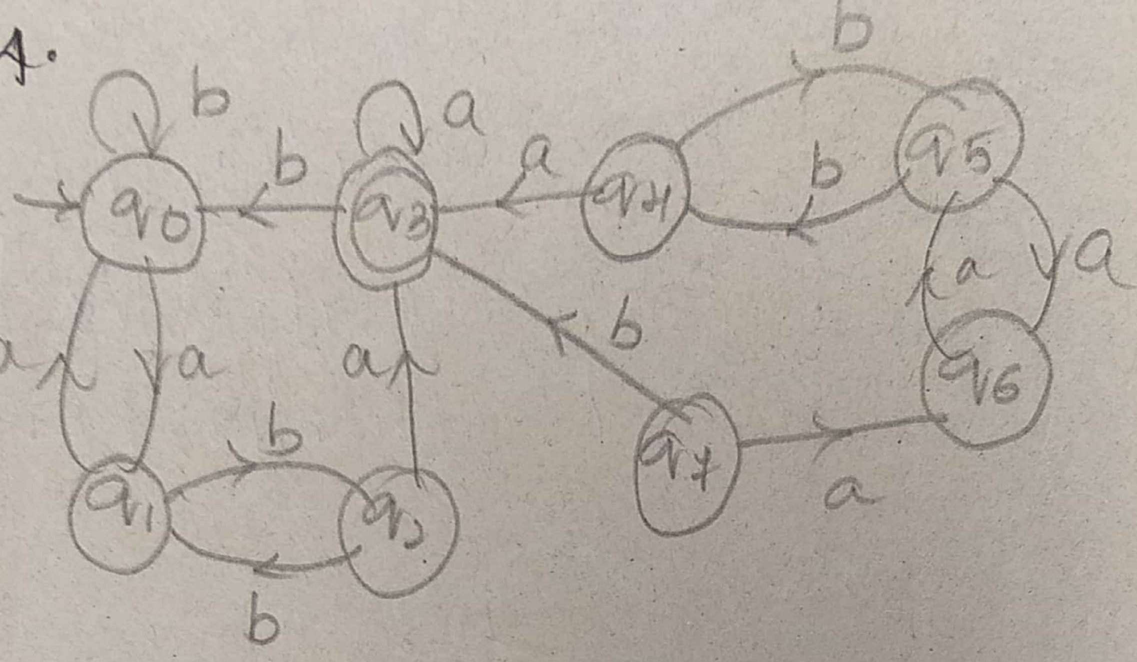
	a	b
q_0	q_1	q_0
q_1^*	q_2	q_1
q_2^*	q_1	q_2



0 equivalence [q_0] [$q_1 q_2$]

1 equivalence [q_0] [$q_1 q_2$]

4.

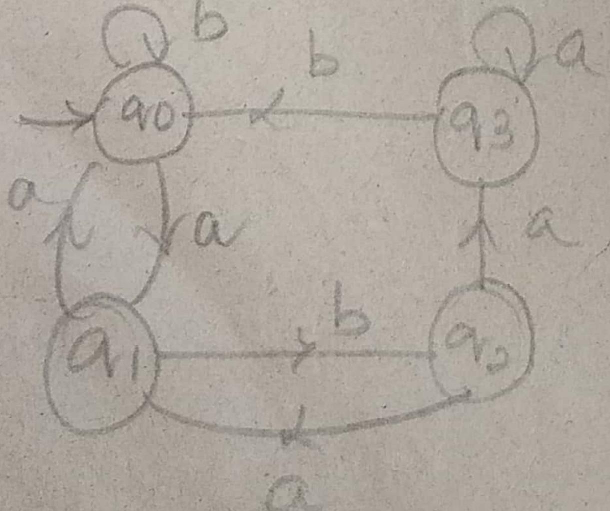


	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3^*	q_3	q_0

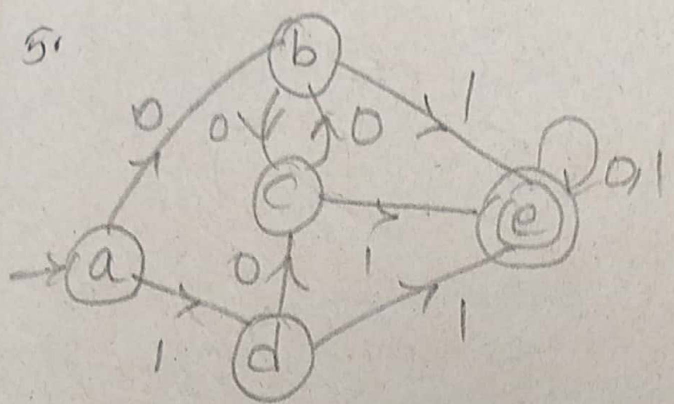
0 equivalent $[q_0 q_1 q_2]$ $[q_3]$

1 equivalent $[q_0 q_1]$ ~~$[q_2]$~~ ~~$[q_3]$~~ $[q_2 q_3]$

2 equivalent $[q_0]$ $[q_1]$ ~~$[q_2]$~~ ~~$[q_3]$~~ $[q_2 q_3]$



5.

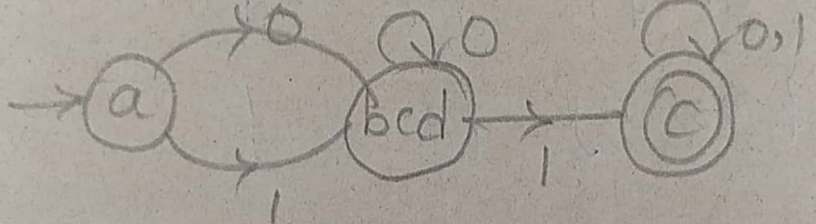


	0	1
→a	b	d
b	c	e*
c	b	e*
d	c	e*
e*	e*	e*

0 equivalent [a,b,c,d] [e]

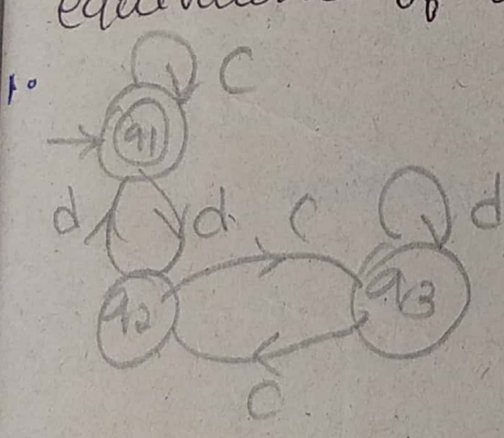
1 equivalent [a] [b,c,d] [e]

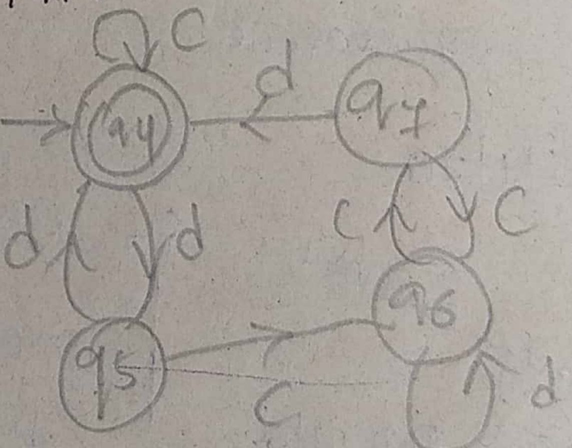
2 equivalent [a] [bcd] [e]



equivalence of two FA.

1.

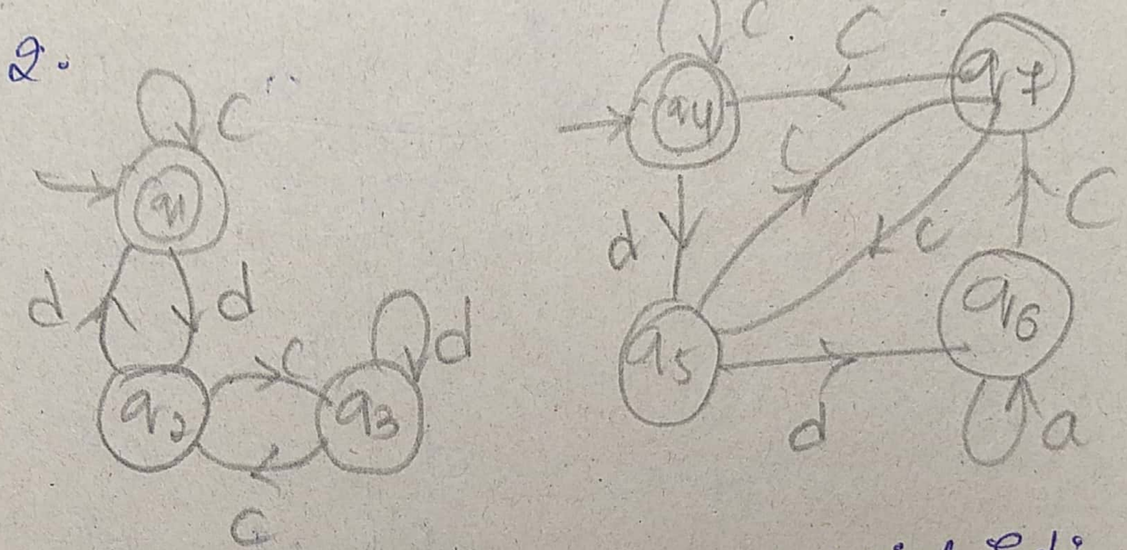




States	c	d
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	$(q_1 q_4)$

Both Finite Automatas are equivalent

2.



Here q_1 & q_4 are both Initial & final state

States	c	d
$[q_1, q_4]$	$[q_1, q_4]$	$[q_2, q_5]$
$[q_2, q_5]$	$[q_3, q_7]$	$[q_1, q_6]$

Then the given automata are not equivalent.

25/1/20

RE Regular expression

represent

Regular Grammar

RG —Generator— RL ←Acceptor— FA

LRG   RRG

Left   Right

RE:→

operators :→

$+$ – union

$*$ – closure

$\cdot$ – concatenation

$\phi \Rightarrow \{\ \}$

$\epsilon \Rightarrow \{\epsilon\}$

$a \in \Sigma \Rightarrow \{a\}$

closure :→ $a^* = \{\epsilon, a, aa, ---\}$

$a^+ = \{a, aa, ----\}$

$a^+ = a \cdot a^*$

→ a's are even b's are any thing.

$$[b^* a b^* a b^*]^* + b^*$$

$$L = \{\epsilon, aab, baab, bbaaaa, babababa, ----\}$$

→ Starts with a & ends with a or b.

$L = \{ab, aa, aba, --\}$

$a(a+b)^*$

ends with a ⇒ $(a+b)^* a$

contains a ⇒ $(a+b)^* a (a+b)^*$

→ starts and ends with different symbols

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

→ Starts and ends with same symbol.

$L = \{aa, bb, aba, aaba,$
$bab, baab, --\}$

$\epsilon+a+b+a(a+b)^*a$ + ~~(or)~~ $b(a+b)^*b$

→ ~~set~~ Find out RE set of all strings over $\{a,b\}$ which does not contain two a's together

$L = \{\epsilon, b, bb, bbb, --- a, ab,$
$aba, --- bab, ----\}$

→ set of all strings no two a's & no two b's are together.

$L = \{\epsilon,$

→ prove that $r(s+t) = rs + rt$

From identity.

$(\because R(P+Q) = RP + RQ)$

$L.H.S = R.H.S$

→ prove that $(\epsilon + 1^*(011)^*(1^*(011)^*))^* = (1+011)^*$

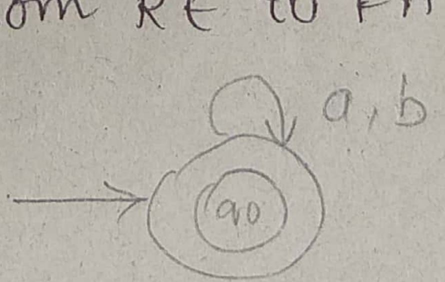
$\Rightarrow (1^*(011)^*)^* \quad \because (\epsilon + R^*R = R^*)$

$(1+011)^* \quad \because (P^* + Q^*)^* = (P+Q)^*$

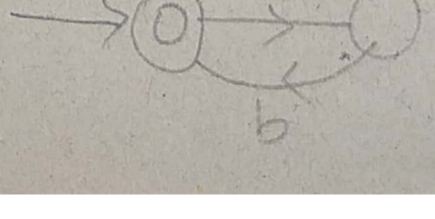
$\therefore L.H.S = R.H.S.$

→ conversion from RE to FA

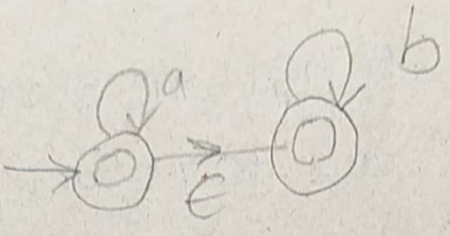
$(a+b)^*$



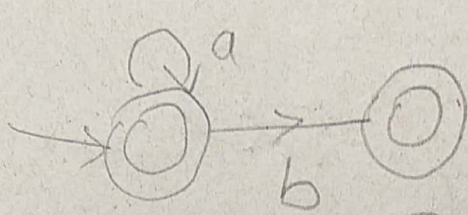
$(ab)^*$



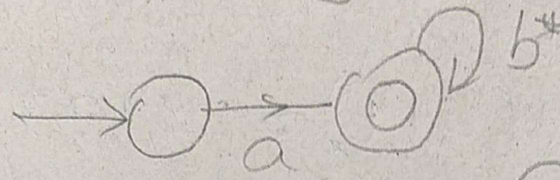
(a^*b^*)



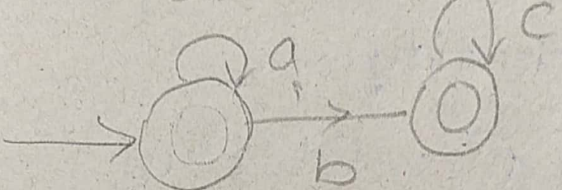
a^*b



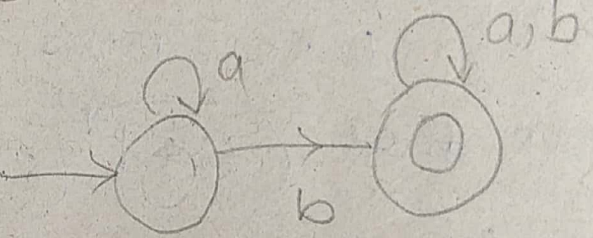
ab^*



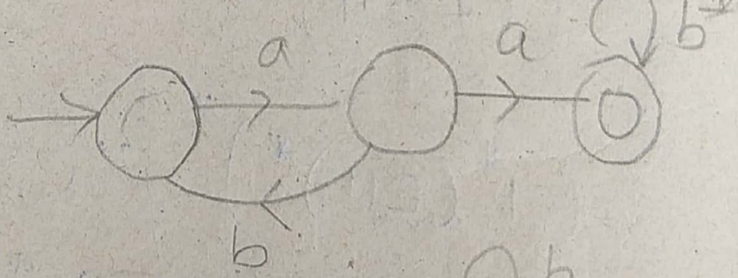
a^*bc^*

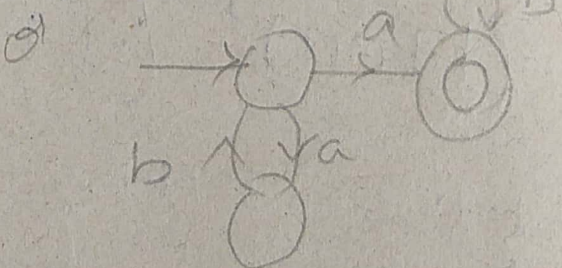


$a^*b(a+b)^*$

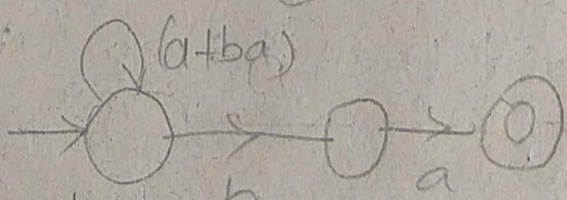


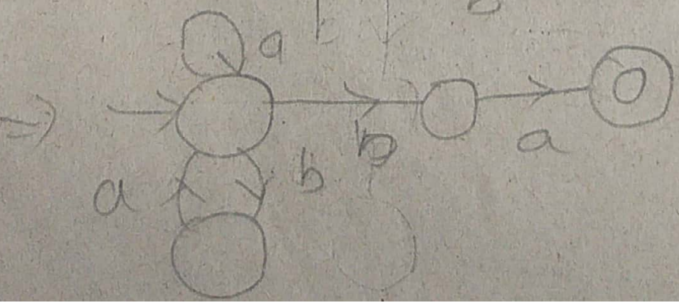
$(ab)^*ab^*$



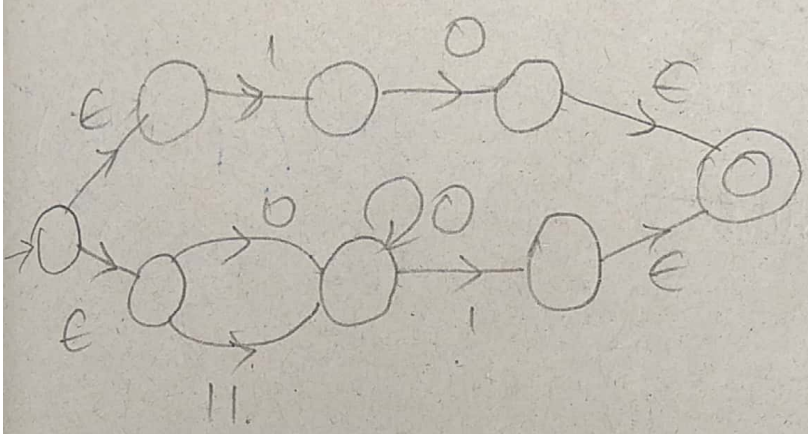


$(a+ba)^*ba$



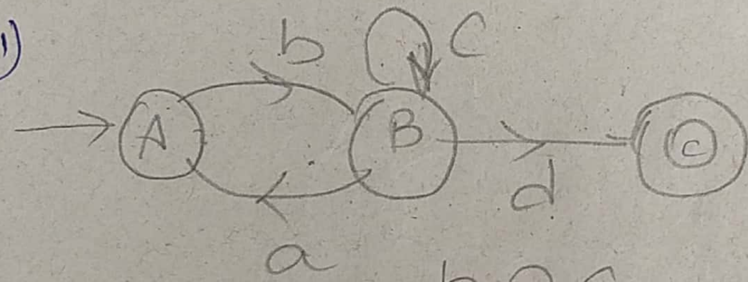


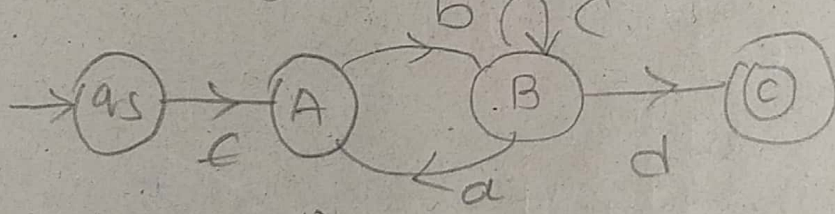
$10(0+11)0^*1$

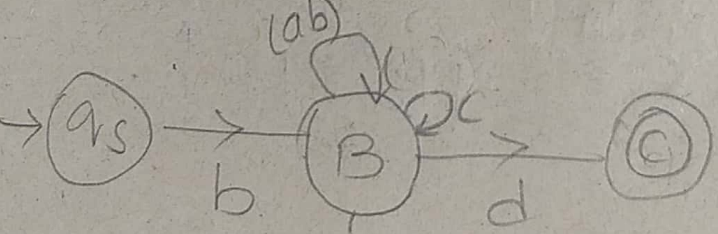


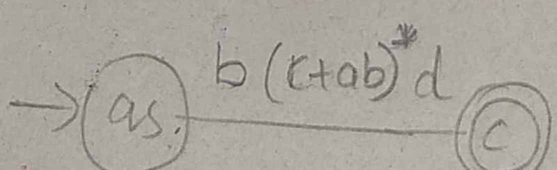
State elimination method.

1)



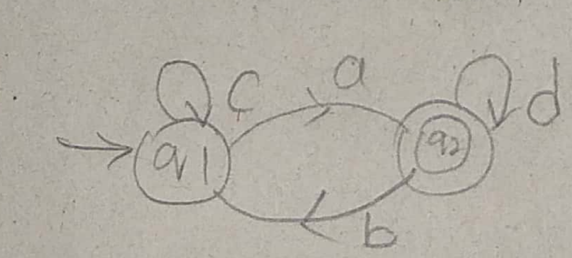


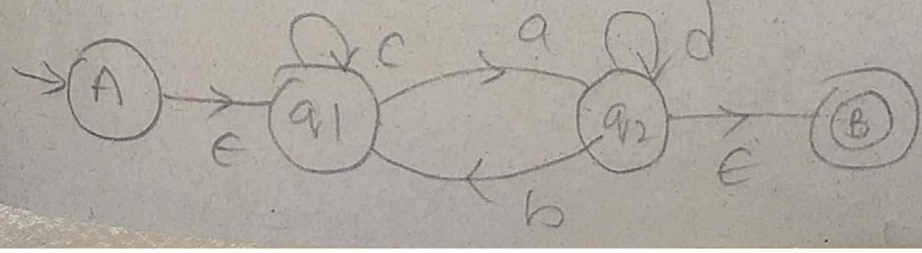


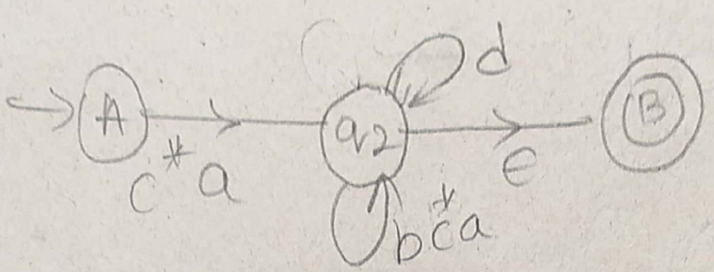


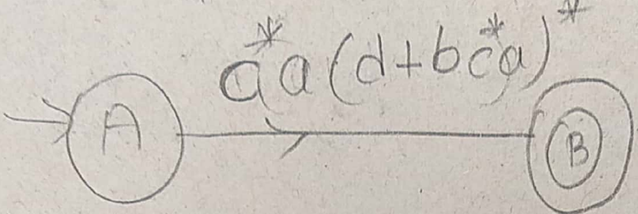
$b(c+ab)^*d.$

2)





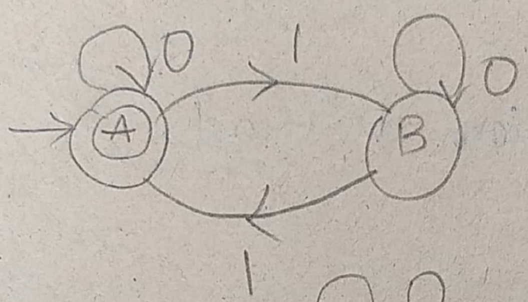


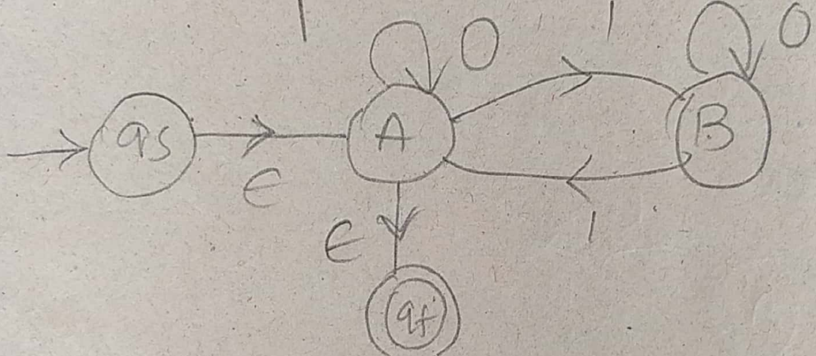


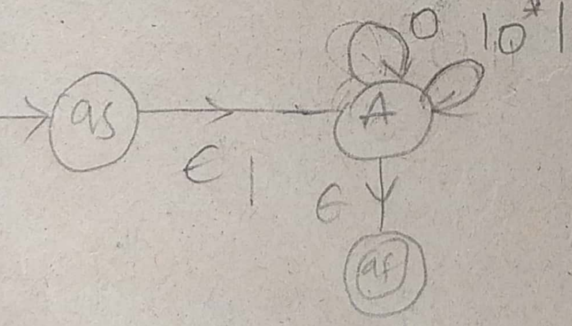
$c^*a(d+bc^*a)^*$

$a(c+d+ab)^*$

3)

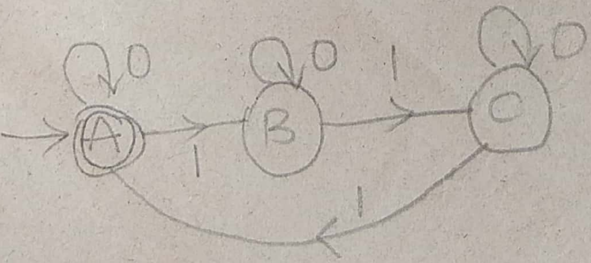


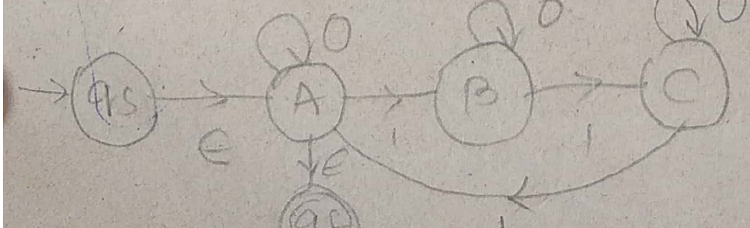


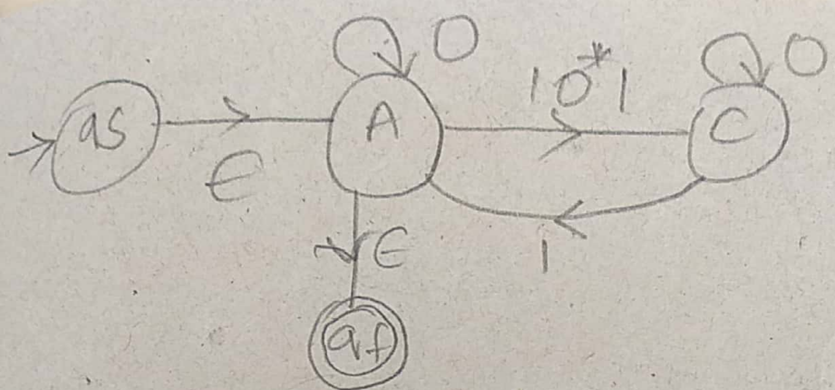


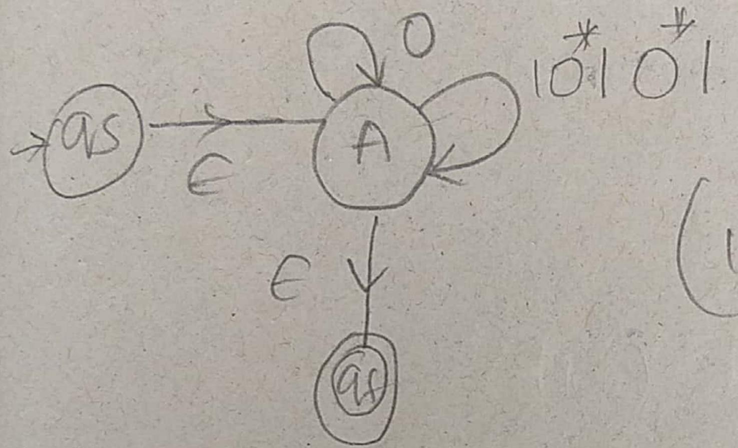
$[10^*1+0]^*$

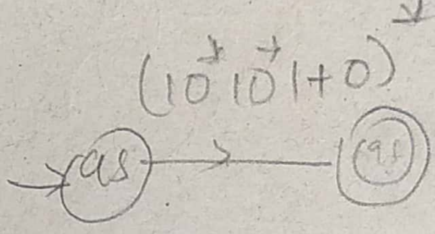
4)











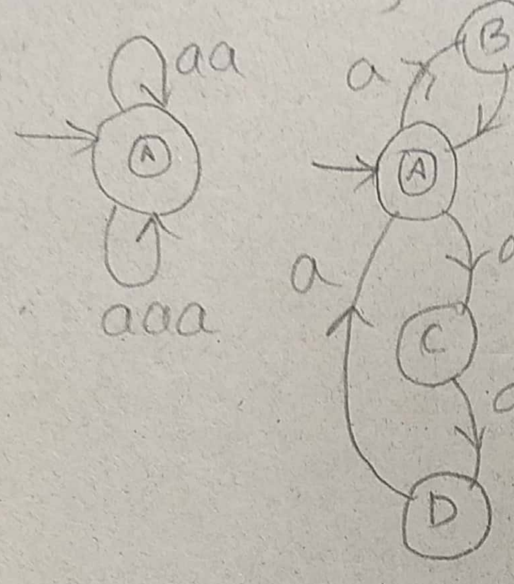
$$(10^*10^*1+0)^*$$

Using ~~Arden's theorem~~

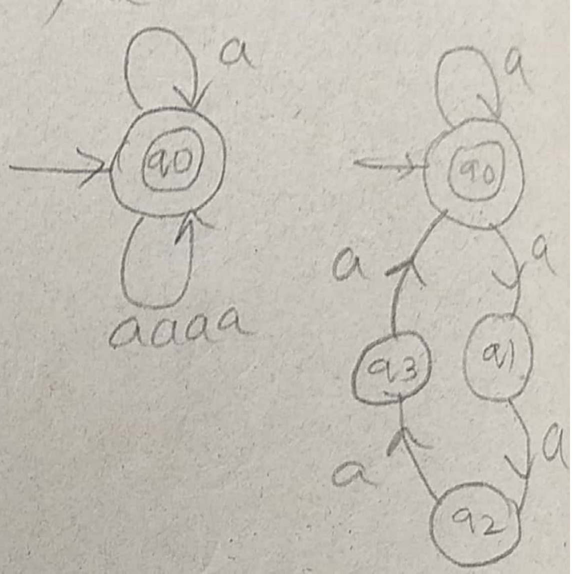
RE to FA.

1. $(aa+aaa)^*$
2. $(a+aaaa)^*$
3. $(ab)^* + (a+ab)^* b^* (a+b)^*$
4. $[a+ba(a+b)]^* a (ba)^* b^*$
5. $b(a+ba+abb)(ba(a+b)^*)$
6. $(a+b+ca)((bab)^* + (a+b)^*)^* (ab)^*$

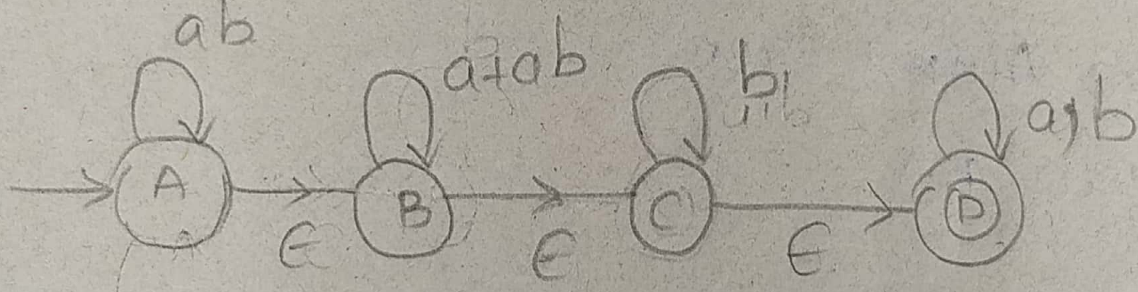
1) $(aa+aaa)^*$

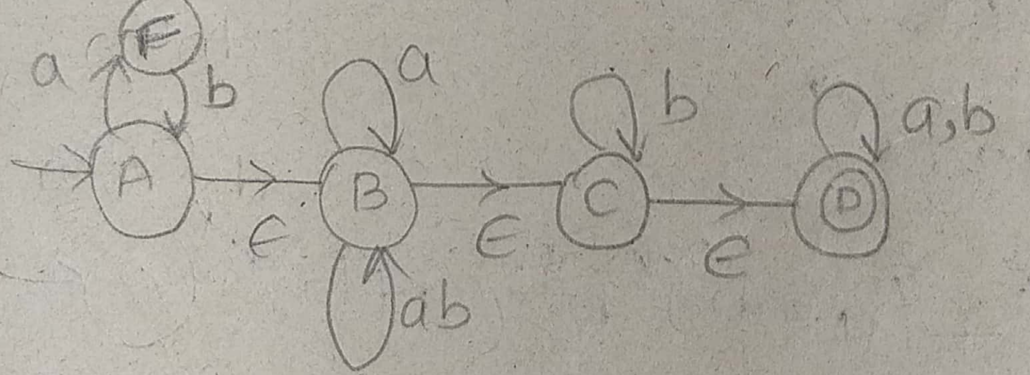


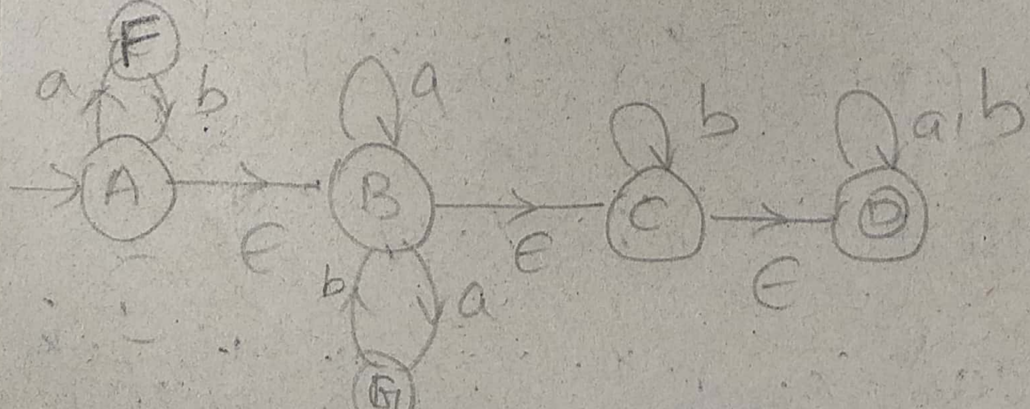
2) $(a+aaaa)^*$



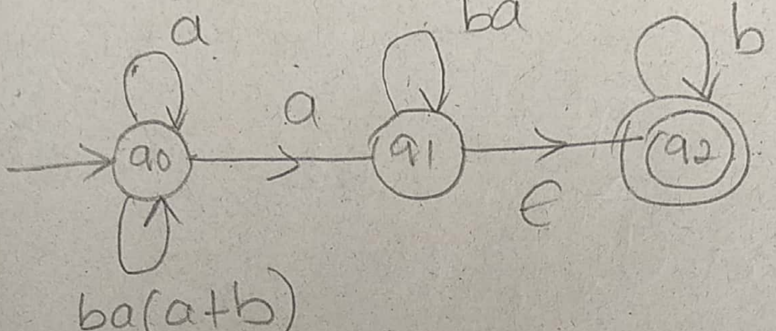
3) $(ab)^* + (a+ab)^* b^* (a+b)^*$



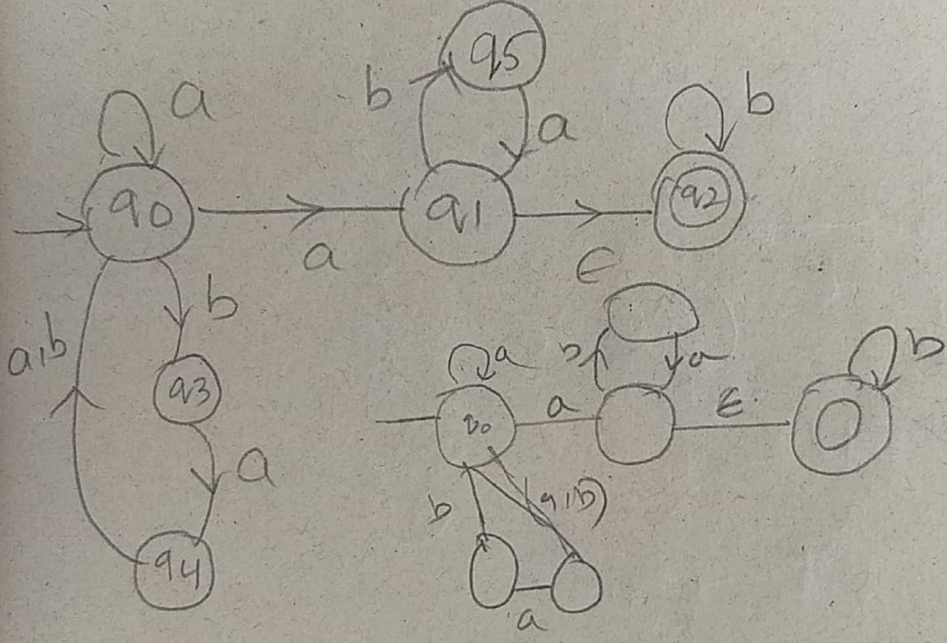




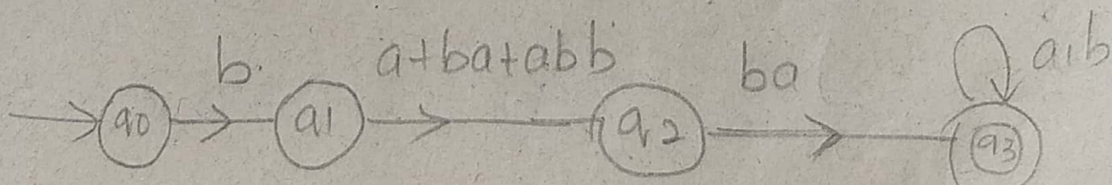
4) $\left(a+ba(a+b)\right)^* a(ba)^* b^*$

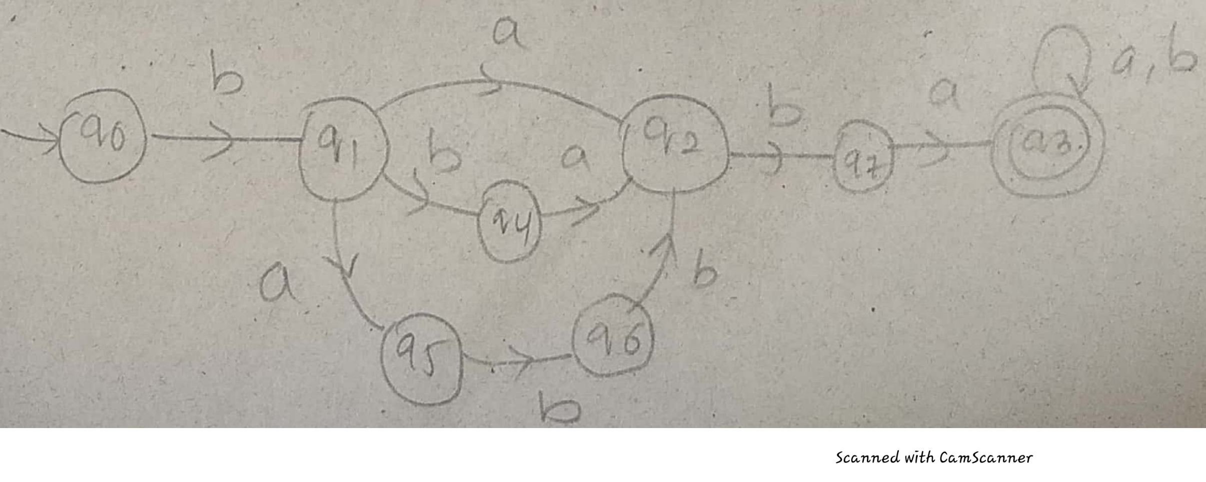


a(ba+a)



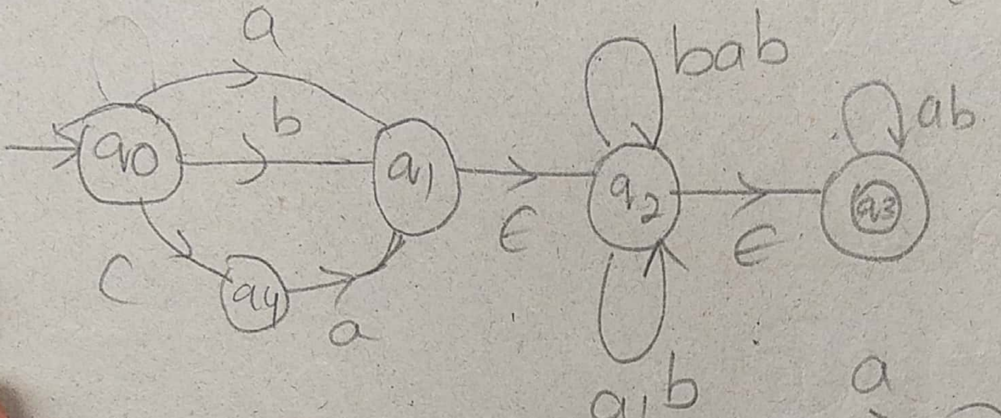
5) $b(a+ba+abb)\left(ba(a+b)^*\right)$

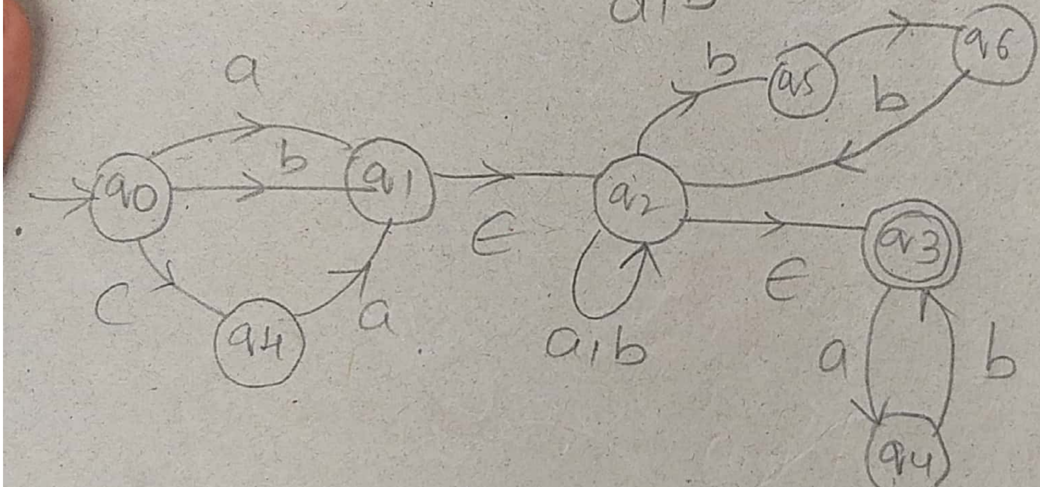




6) $(a+b+ca)\left[(bab)^* + (a+b)^*\right]^* (ab)^*$

$\left[\because (p^* + Q^*)^* = (p+Q)^*\right.$

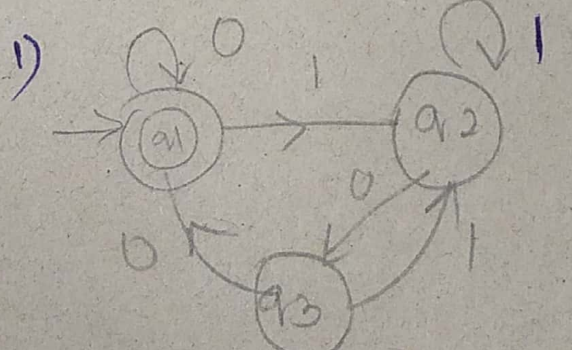




27|1|20

Conversion ~~from~~ FA to RE

1)



$q_1 = q_1 0 + q_3 0 + \epsilon \rightarrow (1)$

$q_2 = q_1 1 + q_2 0 + q_3 1 \rightarrow (2)$

$q_3 = q_2 0$

$$q_2 = q_1 1 + q_2 1 + q_2 01$$

$$\underbrace{q_2}_{R} = \underbrace{q_1 1}_{Q} + \underbrace{q_2}_{R}\underbrace{(1+01)}_{P}$$

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

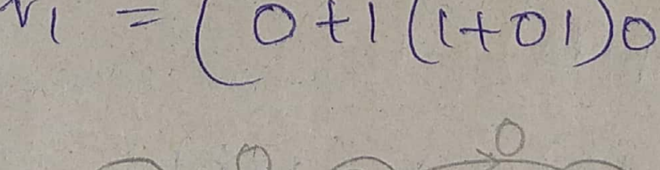
$$q_2 = q_1 1 (1+01)^*$$
$$q_3 = q_1 1 (1+01)^* 0$$
$$q_1 = q_1 0 + q_1 1 (1+01)^* 00 + \epsilon$$

$$q_1 = \epsilon + q_1 0 + q_1 1 (1+01)^* 00$$

$$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R}\underbrace{\left(0+1(1+01)^*00\right)}_{P}$$

$$q_1 = \epsilon\left(0+1(1+01)^*00\right)^*$$

$$q_1 = \left(0+1(1+01)^*00\right)^*$$

2)


$$q_1 = \epsilon$$
$$q_2 = q_1 0 + q_3 0$$
$$q_3 = q_2 0$$

[NOTE: add ϵ in the initial state equations]

$q_2 = \epsilon.0 + q_3 0$

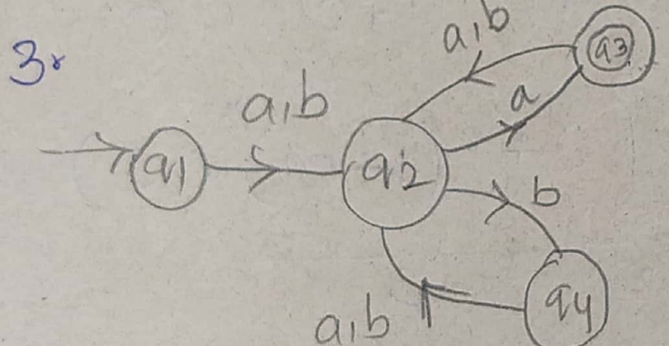
$q_2 = 0 + q_3 0 \Rightarrow q_2 = 0 + q_2 00$

$q_2 = 0(00)^*$

$\cancel{q_3 = (0 + q_3 0)0}$

$\cancel{q_3 = 0 + q_3 0}$

3.



$q_1 = \epsilon$

$q_2 = q_1 a + q_1 b + q_3 a + q_3 b + q_4 a + q_4 b$

$= q_1(a+b) + q_3(a+b) + q_4(a+b)$

$q_3 = q_2 a$

$q_4 = q_2 b$

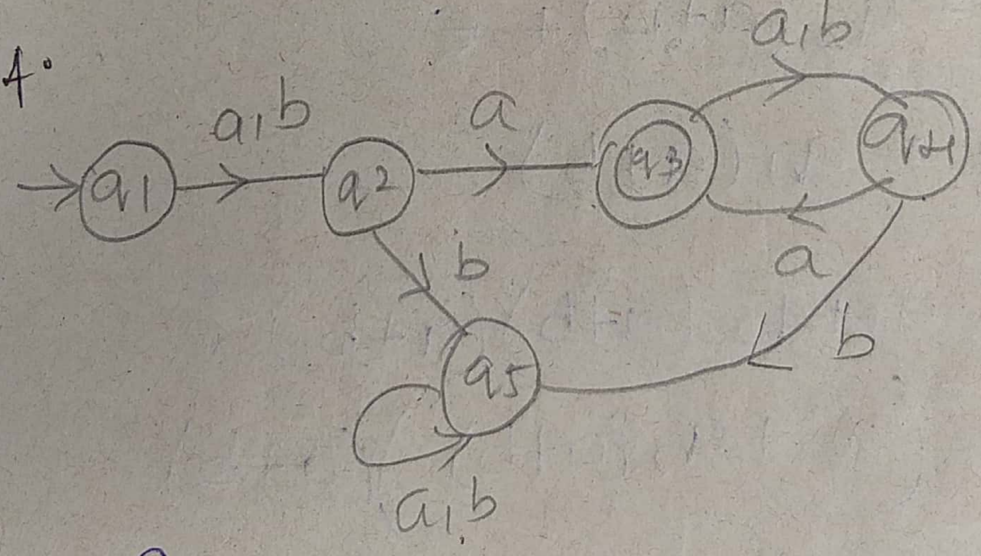
$$q_2 = \epsilon(a+b) + q_2 a\,(a+b) + q_2 b\,(a+b)$$

$$q_2 = (a+b) + q_2\left[a(a+b) + b(a+b)\right]$$

$$q_2 = (a+b)\left[a(a+b) + b(a+b)\right]^*$$

$$q_3 = (a+b)\left[a(a+b) + b(a+b)\right]^* a.$$

4.



$q_1 = \epsilon$

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

$q_3 = q_2 a + q_4 a$

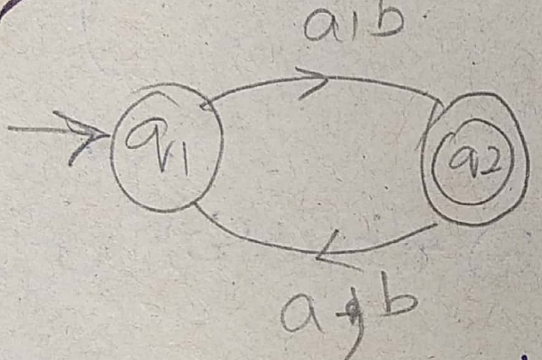
$q_4 = q_3(a+b)$

$q_5 = q_2 b + q_4 b + q_5(a+b)$

$$q_3 = \underbrace{q_3}_{R} = \underbrace{q_2 a}_{Q} + \underbrace{q_3}_{R}\,\underbrace{(a+b)a}_{P}$$

$$q_3 = (a+b)a\,\big(a(a+b)\big)^*$$

$\rightarrow$



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

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

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

$$q_2 = \cancel{\epsilon + q_2(a+b)}(a+b)$$

$$q_2 = \big(\epsilon + q_2(a+b)\big)(a+b)$$

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

$$q_2 = \big[(a+b)(a+b)\big]^*$$

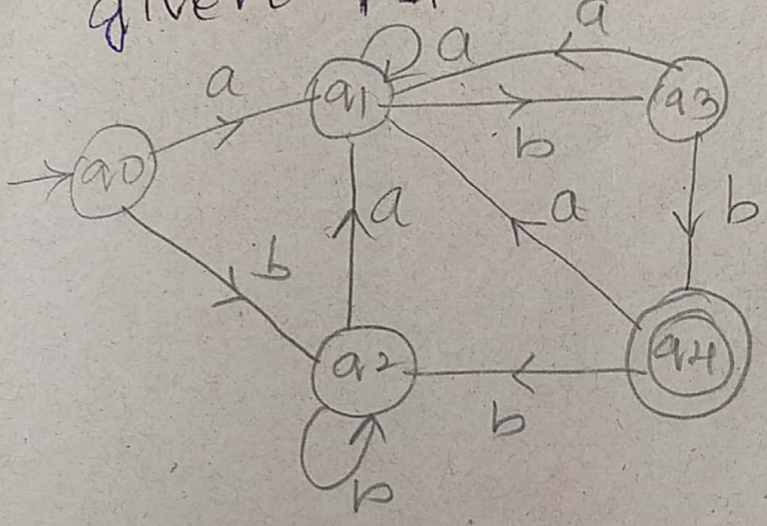
Minimization of FA

two ways.

way 1:- partition

way 2 :- Table filling.

1) construct a minimal DFA for the given FA



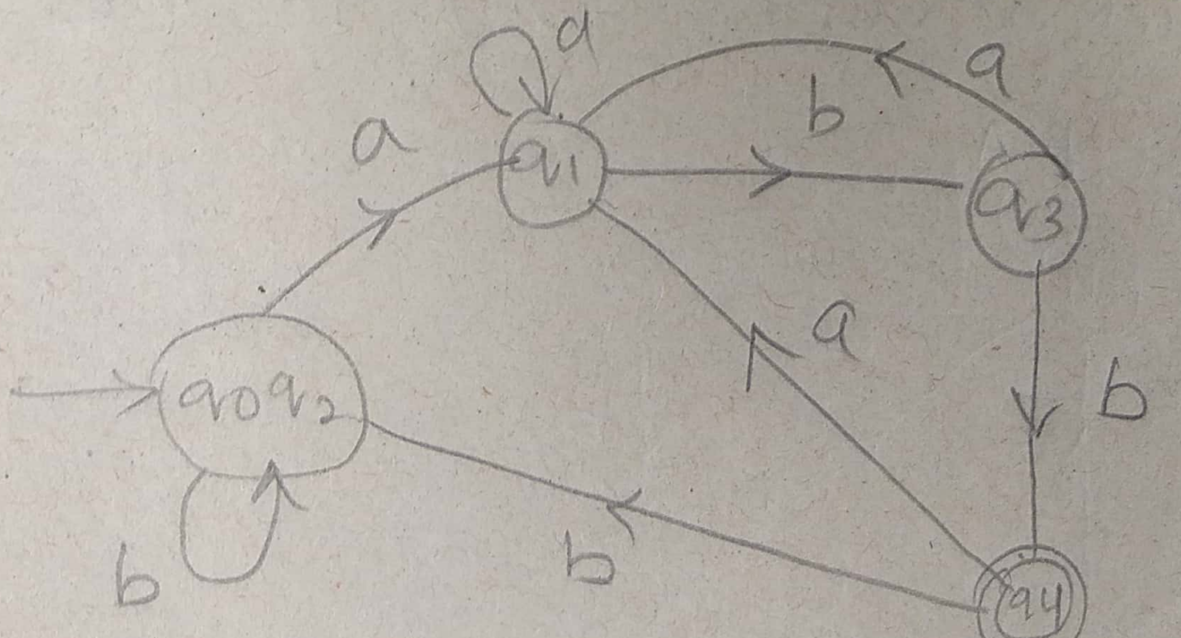
c	a	b
→ q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
* q4	q1	q2

step 1:- check all the states which are not reachable from the initial state then eliminate the particular state

0-equi $[q_0 q_1 q_2 q_3]$ $[q_4]$

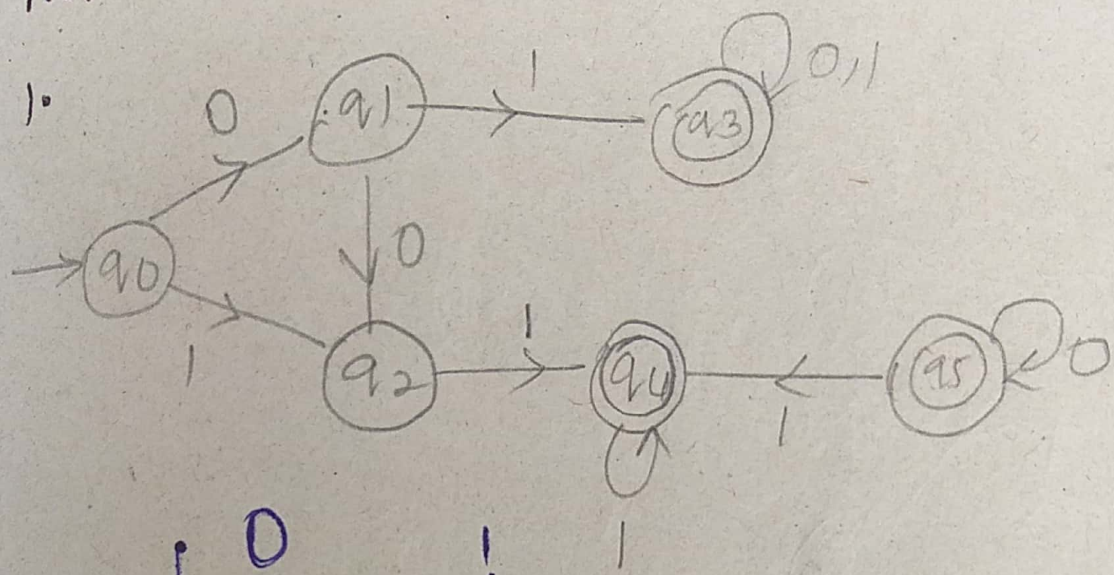
1-equi $[q_0 q_1 q_2]$ $[q_3]$ $[q_4]$

2-equi $[q_0 q_2]$ $[q_1]$ $[q_3]$ $[q_4]$



Minimization of DFA

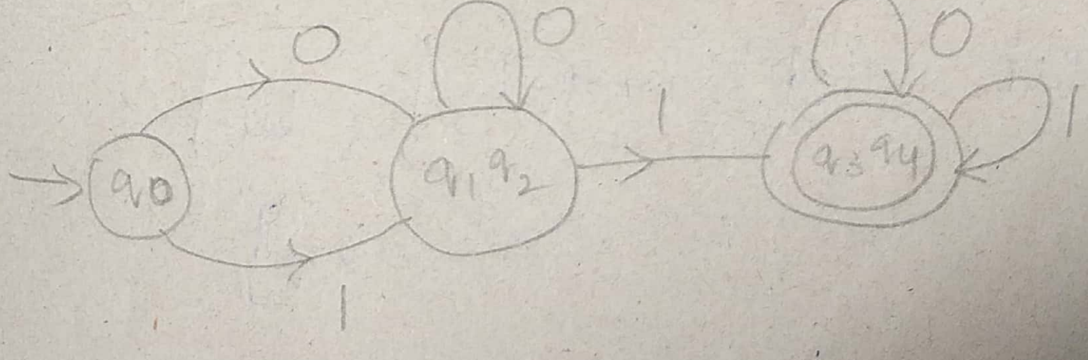
1.



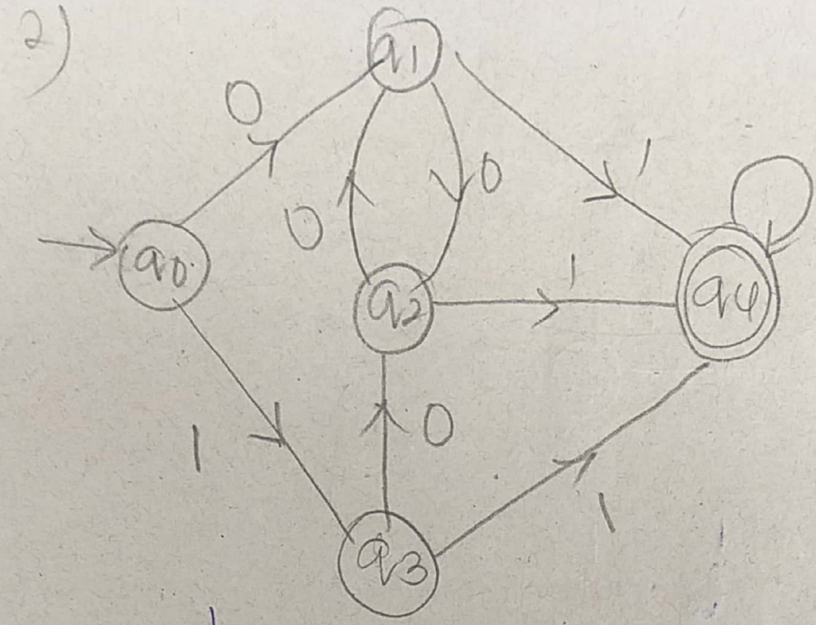
	0 a	1 b
→q0	q1	q2
q1	q2	q3
q2	φ	q4
q3*	q3	q3
q4*	φ	q4

0-equi [q0 q1 q2] [q3 q4]

1-equi [q0] [q1 q2] [q3 q4]



2)

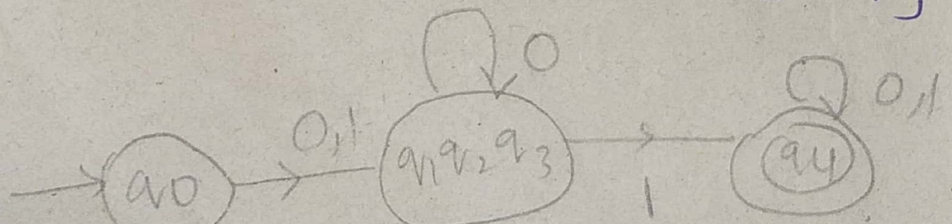


	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
q_4^*	q_4	q_4

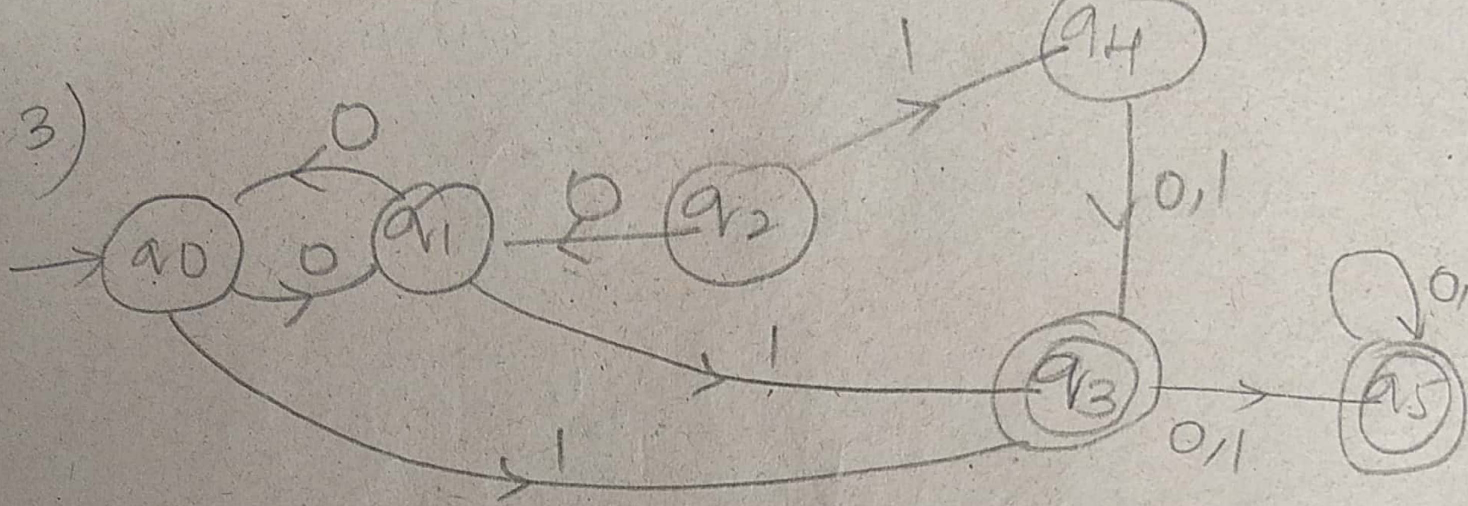
0-equi $[q_0 q_1 q_2 q_3]$ $[q_4]$

~~1-equi $[q_0 q_2]$ $[q_1 q_3]$ $[q_4]$~~

1-eq $[q_0]$ $[q_1 q_2 q_3]$ $[q_4]$



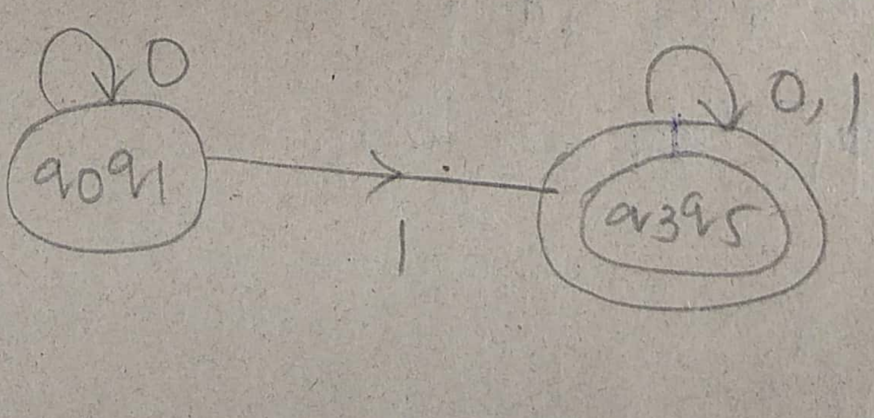
3)



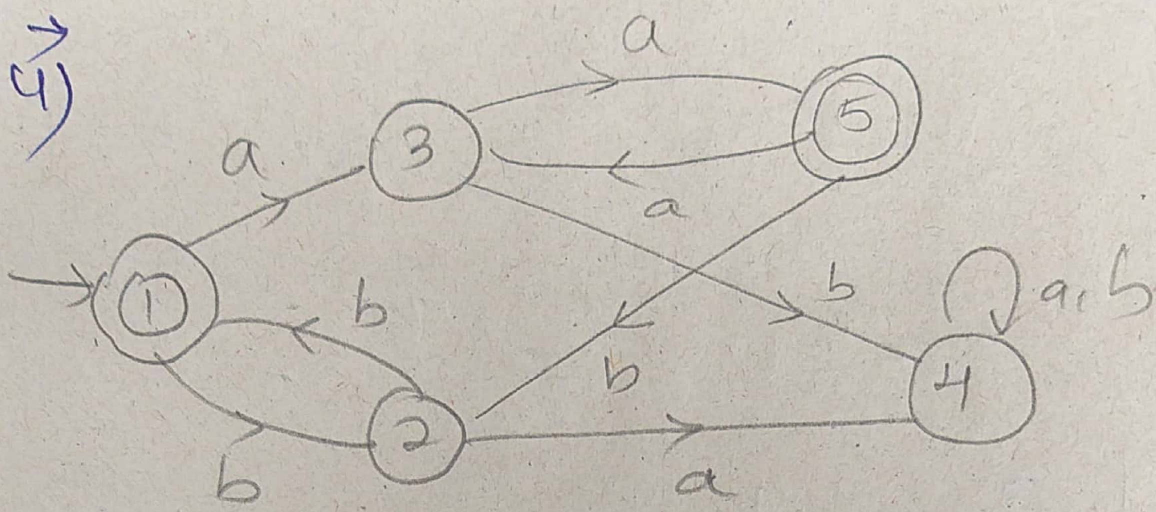
	0	1
→ q_0	q_1	q_3
q_1	q_0	q_3
q_3*	q_5	q_5
q_5*	q_5	q_5

0-equi $[q_0\, q_1]$ $[q_3\, q_5]$

1-equi $[q_0\, q_1]$ $[q_3\, q_5]$



4)

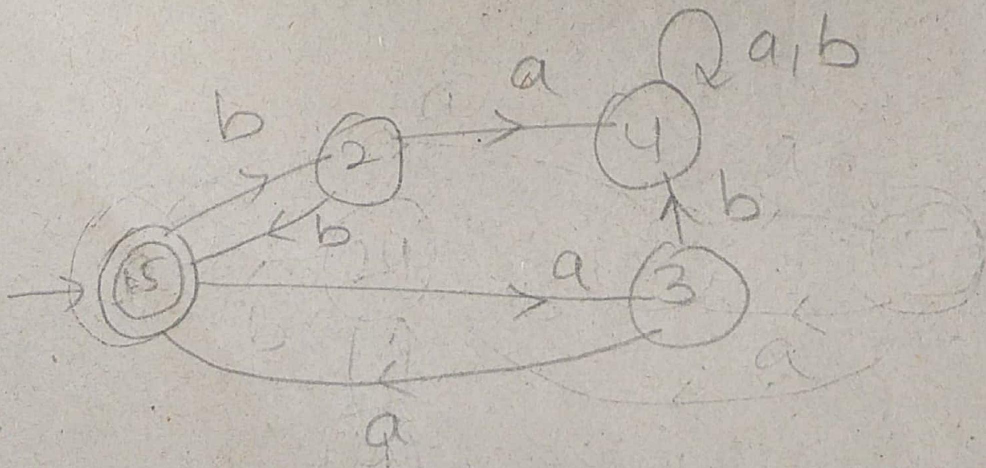


	a	b
→ 1*	3	2
2	4	1
3	5	4
4	4	4
5*	3	2

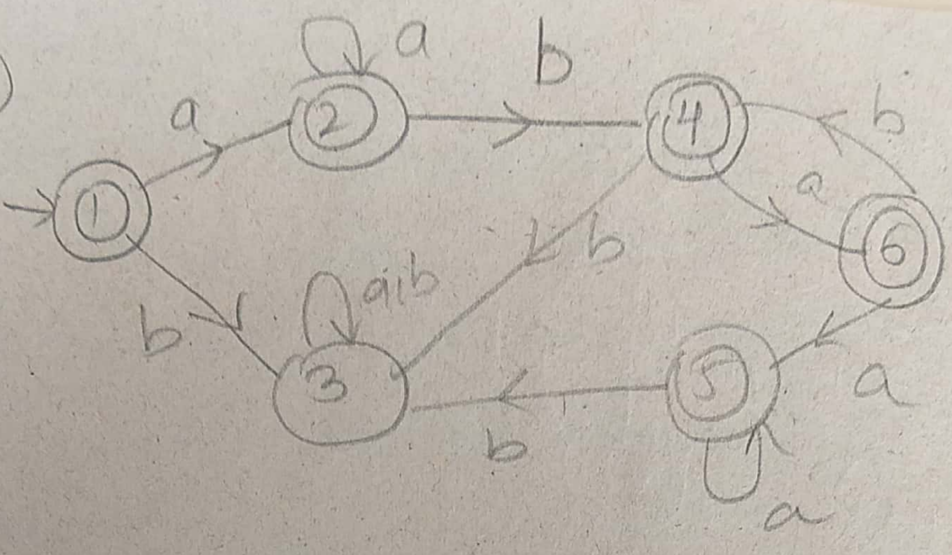
0-equi [~~1~~ 2 3 4] [1,5]

1-equi- [~~1~~ 2 4] [3] [1,5]

2-equi [2] [4] [~~4~~] [3] [1,5]



5)



	a	b
→1*	2	3
2*	2	4
3	3	3
4*	6	3
5*	5	3
6*	5	4

[3] [1 2 4 5 6]

[3] [1 4 5] [2,6]

[3] [

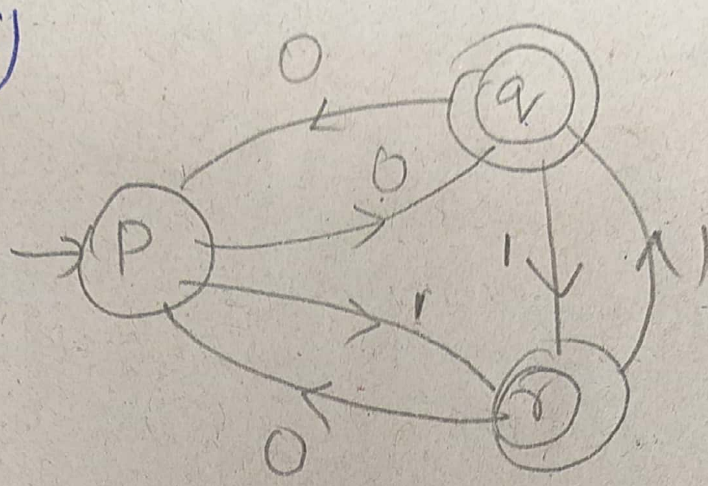
0-equi [3] [1 2 4 5 6]

1-equi [3] [1 4 5 6] [2]

2-equi [3] [2] [1 5 6] [4] [~~2~~]

[3] [2]

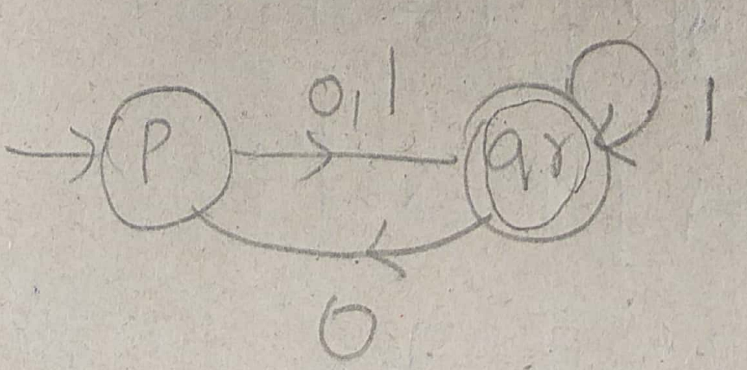
6)



	0	1
→p	q	r
q*	p	r
r*	p	q

0-equi [p] [q r]

1-equi [p] [q r]



28/1/20

pumping Lemma for Regular Language.

→ pumping lemma is used to prove that the given language is not a regular.

→ It cannot be used to prove the Language is regular.

→ Whenever the given language is regular then we are able to construct DFA, NFA, ϵ-NFA automatas to accept the given language

Formal definition of pumping lemma:

→ let 'L' be a Regular Language then there exists a constant 'p' such that for every string 'W' in L.

→ Where the length of W i.e $|W| \geq P$

→ W can be divided into three parts i.e $W = xyz$ such that the following conditions must be true.

i) $|y| \geq 0$

ii) $|xy| \leq P$

iii) for all $k \geq 0$ then the string xy^kz is also in 'L'.

Applications:-

→ pumping lemma is to be applied to show that certain languages are not regular

→ It should never be used to show the language is Regular.

- If L is regular it satisfies pumping Lemma
- If L doesn't satisfy pumping Lemma then it is not a regular.

→ To prove that a Language is not regular using pumping Lemma.

Step 1:- We prove using contradiction

→ Assume that L is regular.

Step 2: It has a pumping length i.e P.

All strings longer than P, then only can be pumped.

→ Now find a string W in L such that $|W| \geq P$

→ divide W into xyz.

→ Show that none of these three ($|y| \geq 0$, $|xy| \leq P$, for all $k \geq 0$ then $xy^kz \in L$)

conditions can satisfy.

→ There W cannot be pumped.

1) show that the given language $L=\{0^n 1^n / n\geq 1\}$ is not a regular.

$L=\{01, 0011, 000111, ---\}$

let the set L is defined by $L=\{01, 0011, --\}$.

choose a string from the defined language and apply pumping lemma

let $W = 0011$

here, $x=0$ $y=01$ $z=1$

for $k\geq 0$ then $xy^kz \notin L$

here $k=2$ then

$\Rightarrow 0(01)^2 1$

$\Rightarrow 001011 \notin L$

then the given language is not a regular language.

2) show that the set $L=\{0^{i^2} / i\geq 1\}$ is not a regular.

$L=\{0, 0000, 000000000, ----\}$

let $W = 0000$

let $x=00$ $y=0$ $z=0$

for $K \geq 0$ then $xy^Kz \notin L$

for $K=2$

$\Rightarrow 00(0)^2 0$

$\Rightarrow 00000 \notin L$

$\therefore$ The given language is not Regular

**

3) Show that the given Language $L = \{a^P / P$ is prime$\}$ is not regular

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

let $W = aaa$

let $x = a \quad y = a \quad z = a$

for $K \geq 0$ then $xy^Kz \notin L$

for $K=2$

$a(a)^2 a$

$\Rightarrow aaaa \notin L$

$\therefore$ The given language is not regular

(or) $W = \underbrace{aa}_{x}\underbrace{aa}_{y}\underbrace{a}_{z}$

$xy^Kz \notin L$

if $K=3$ $\quad aa(aa)^3 a \Rightarrow aaaaaaaaa \notin L$

4) show that the following language is not a regular $L = \{a^n b a^n \mid n \geq 0\}$

$L = \{b, aba, aabaa, aaabaaa, --\}$

let $W = aabaa$

let $x = aa$ $y = b$ $z = aa$

for $k \geq 0$ $xy^kz \notin L$

$k = 2$

$\Rightarrow aa(b)^2aa$

$\Rightarrow aabbaa \notin L$

$\therefore$ It is not a regular Language

5) show that the given Language is not regular where $L = \{a^n b^{2n} \mid n > 0\}$

$L = \{abb, aabbbb, aaabbbbbb---\}$

let $W = abb$

let $x = a$ $y = b$ $z = b$

for all $k \geq 0$ $xy^kz \notin L$

let $k = 2$

$\Rightarrow a(b)^2b$

$\Rightarrow abbb \notin L$

$\therefore$ It is not a regular language.

Scanned with CamScanner

6) Show that the given language is not regular $L = \{ww \mid w \text{ in } (a,b)^*\}$

$L = \{abab, baba, aabbaabb \text{---}\}$

let $W = abab$

$x = ab \quad y = a \quad z = b$

for all $k \geq 0 \quad xy^kz \notin L$

$k = 2$

$\Rightarrow ab(a)^2b$

$\Rightarrow aba\overset{a}{a}b \notin L$

$\therefore$ It is not a regular Language

7) $L = \{WW^R / w \text{ in } (a,b)^*$ $R \Rightarrow$ reverse

$L = \{abba, baab, bbaabbb, \text{---}$
aabbaa, - -

$w = aabbaa$

$x = aab \quad y = ba \quad z = a$

for all $k \geq 0, \; xy^k z \notin L$.

for $k=2$ $\; aab(ba)^2 a$

$\Rightarrow aabbabaa \notin L$

$\therefore$ It is not a regular

Closure properties of Regular Language:-
(Assignment)

Decision properties:-

i) Membership $\rightarrow$
ii) Equivalence
iii) Emptyness
iv) Containment:- $(L_1 - L_2)$ is also regular