Regular Expression:-

Chomsky classifies the languages based on its ~~languag~~ grammer in to 4 languages.

Types of languages
- Types of Language [Phase structured language]
- Type 1 Language [context Sensitive Language]
- Type 2 language [context free Language]
- Type 3 Language [Regular Language].

Regular Language: It is defined as language accepte-d by any finite automata ~~there also~~ they are also language defined by regular expression

*** **Regular Expression:** A regular expression is recursively defined as below.

1> ϕ is a R.E which defines empty language
2) ϵ is a R.E which " " string.
3> $a \in \Sigma$, is a R.E defines the language $\{a\}$

if R is a R.E denoting the language $L(R)$,
and S " " " " " " $L(S)$
then, 1> R+S is a R.E denoting the language $L(R) \cup L(S)$ (union operation)

2> R.S is a R.E denoting the language $L(R) \cdot L(S)$ (concatination operation)

3> R* is a R.E denoting the language $[L(R)]^*$ (closure operation)

All R.E obtained by any of the rules above are R.E.

Write the R.E for the following languages.

R.E	R.L.
ϕ	Empty language
ε	Empty string
a	language containing Exactly one a
a^*	Zero or more number of a's
$a.a^*$	One or more number of a's
$(a+b)$	String consisting of either a or b.
$a^*.b^*$	String consisting of zero or more number of a's followed by zero or more number of b's.
$ab.(a+b)^*$	Set of strings of a's & b's begining with ab
$(a+b)^*ab$	Set of strings of a's & b's ending with ab
$(a+b)^*ab(a+b)^*$	Set of strings of a's and b's containing substrings ab
$(a+b)^*(a+bb)$	Strings of a's & b's ending with aa or bb
$(aa)^*$	Set of ~~sub~~ string consisting of even number of a's
$(aa)^*a$ or $a(aa)^*$	Set of strings consisting of odd numbers of a's
$(aa+ab+ba+bb)$ or $(a+b)(a+b)$	Strings of length 2 over alphabet ab.

$(a+b+\varepsilon)(a+b+\varepsilon)$	All strings of length less than or equal to 2
$[(a+b)(a+b)]^*$ or $(aa+ab+ba+bb)^*$	Accepting all strings of even length over a and b
$[(a+b)(a+b)]^*(a+b)$ or $(aa+ab+ba+bb)(a+b)$	Accepting all string of odd length over a and b
$(\varepsilon+b)(ab)^*(\varepsilon+a)$ or $(\varepsilon+a)(ab)^*(\varepsilon+b)$	String consisting a's & b's with alternate a & b.
$[a(a+b)^*a+b(a+b)^*b]$	String begining or end with same letter, length of the string is atleast 2
$(a+b)^*ba(a+b)(a+b)$	Accept all string whose 3rd symbol from righ is a and 4th symbol is b.
$(a+b)^*a(a+b)^*b(a+b)^*$ $(a+b)^*b(a+b)^*a(a+b)^*$	All string containing atleast 1 a and atleast 1 b.
$(a+b)^*(aa+bb+ba)$	All strings that donot end with ab.

1> $L=\{a^n b^m \mid m+n \text{ is even}\}$

<u>Sol</u>: $(aa)^*(bb)^*$ or

$(aa)^*a(bb)^*b$

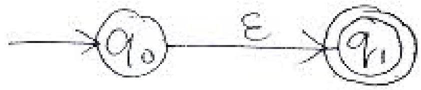
2> $L=\{a^n b^m \mid \text{both } m,n \geq 1, nm \geq 3\}$

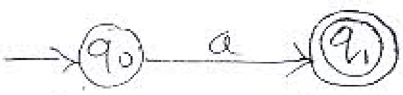
Obtaining ε-NFA from given Regular Expression

*** Theorem

- PT there exist an finite automata to accept a regular language $L(R)$ corresponding to R
- Let R be a regular expression then PT there exist a FA, $M = \{Q, \Sigma, \delta, q_0, F\}$ which accepts the language $L(R)$

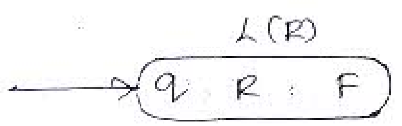
<u>Proof</u>: By definition of R.E ϕ, ε & $a \in \Sigma$ are regular expressions the corresponding automaton can be constructed as below case





Thus we can say that all basic regular expression can be represented using finite automata.

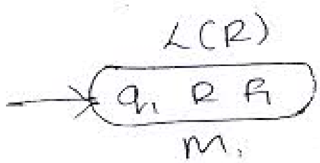
The Schematic representation of finite automata to accept Language $L(R)$ corresponding to R.E R with start state q & final state F is as given below

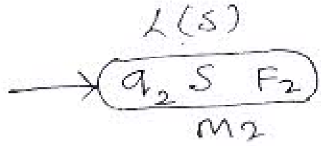


By defn of R.E by applying the operators +, · & * ~~and~~ on another R.E leads to R.E an equivalent automaton can be constructed as below.

Let R & S be two R.E, Let $M_1 = \{Q_1, \Sigma, \delta_1, q_1, F_1\}$ & $M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2\}$

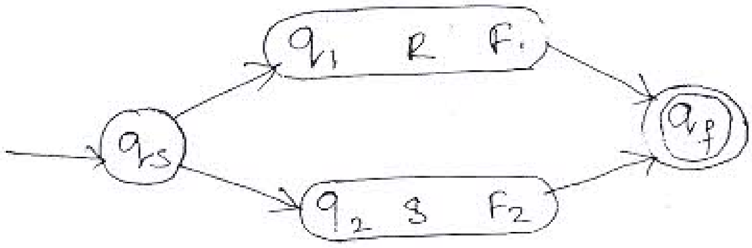
gives the FA's which accepts the language L(R) & L(S) respectively which can be diagrammatically represented as below





case 1> union (+) operation :- A machine to accept the language $L(R) \cup L(S)$ or (R+S) can be constructed by adding 2 new states & 4 ε transitions. make one of the new states a combined start state and ~~one~~ other a combined final state. Add ε transitn from a new start state to old start state. and old final state to one ~~new~~ new final state.

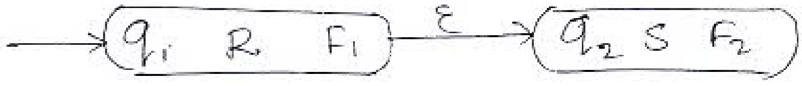
$L(R) \cup L(S) = L(R+S)$



$L(R) \cup L(S)$

case 2> L(R).L(S) concatination operation : We can construct ε NFA which accepts the language L(R) followed by L(S) by adding 1 new ε transition b/n the final state of the first automata and ~~final~~ start state of the second remove final and Start state status of first and Second FA respectively.

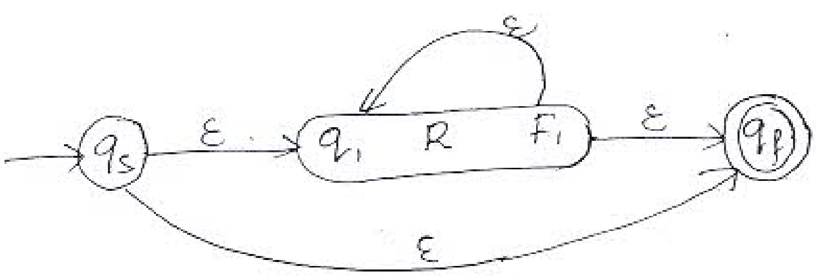
$L(R) \cdot L(S) = L(R \cdot S)$



$L(R \cdot S)$

Case 3) Closure operation $L(R)^*$

we can construct ε-NFA to accept language $L(R)^*$ by adding two new states and four ε-transitions make one of the new state as the start state and other as the final state. Add ε-transitions from 1) New start state to old start state 2) New start state to new final state 3) Old final state to new final state 4) Old final state to old start start state.



$L(R)^*$

Thus we can write the F(A) for all basic R.E and all its possible operators.

Hence we can say that there exist a finite automata for every R.E.

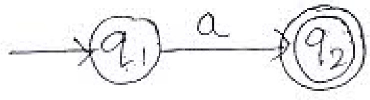
Problems:

1) Obtain ε NFA for following regular expression

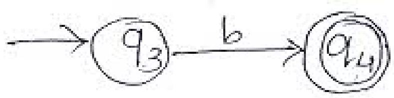
$ab(a+b)^*$

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

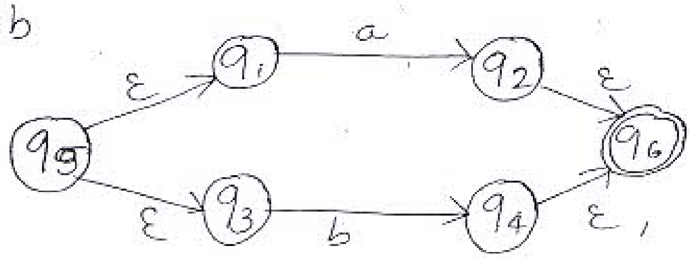
$L = a$



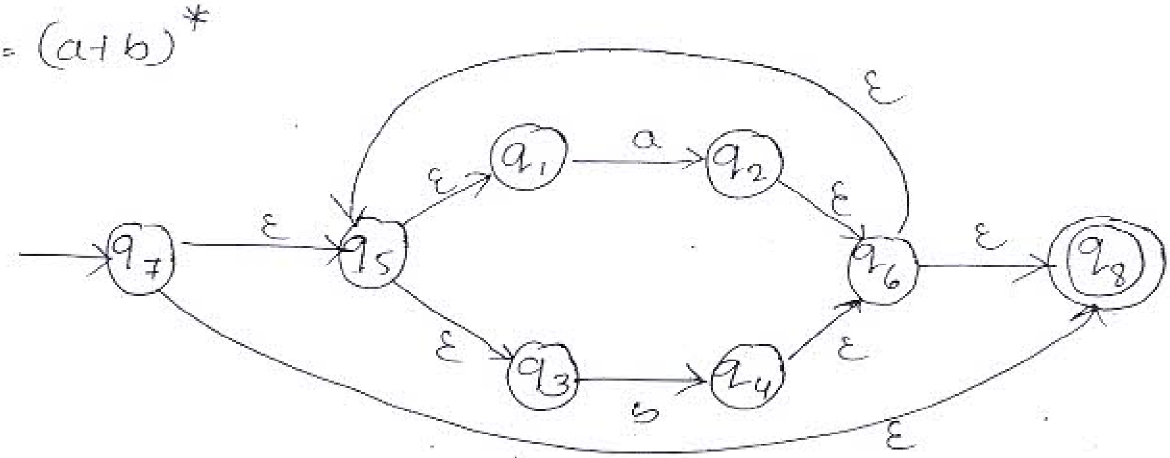
$L = b$



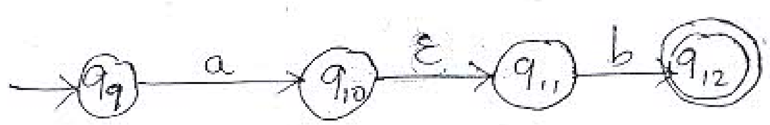
$L = a+b$



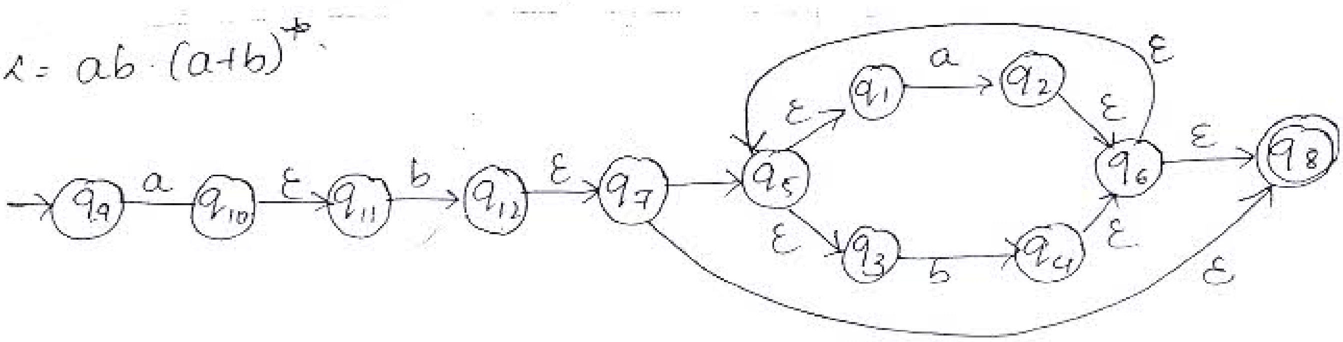
$L = (a+b)^*$



$L = ab$

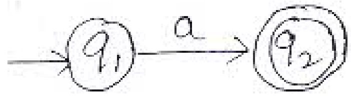


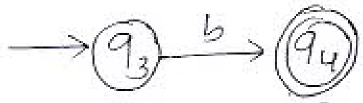
$L = ab \cdot (a+b)^*$



2) $a^* + b^* + c^*$

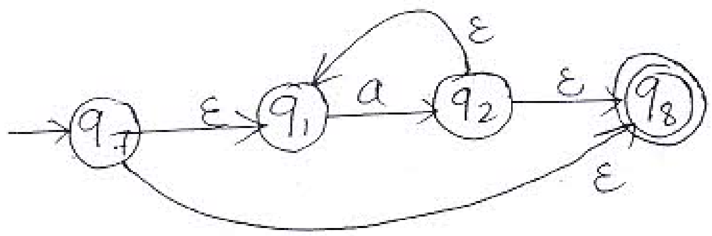
Sol:

$L = a$ 

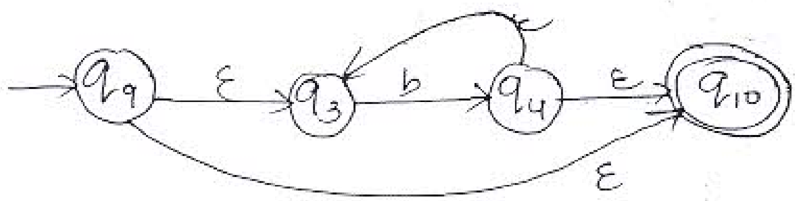
$L = b$ 

$L = c$ 

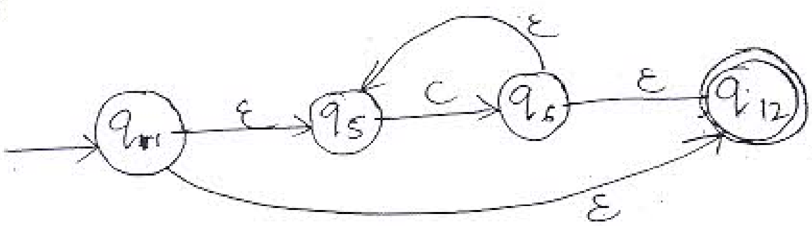
$L = a^*$



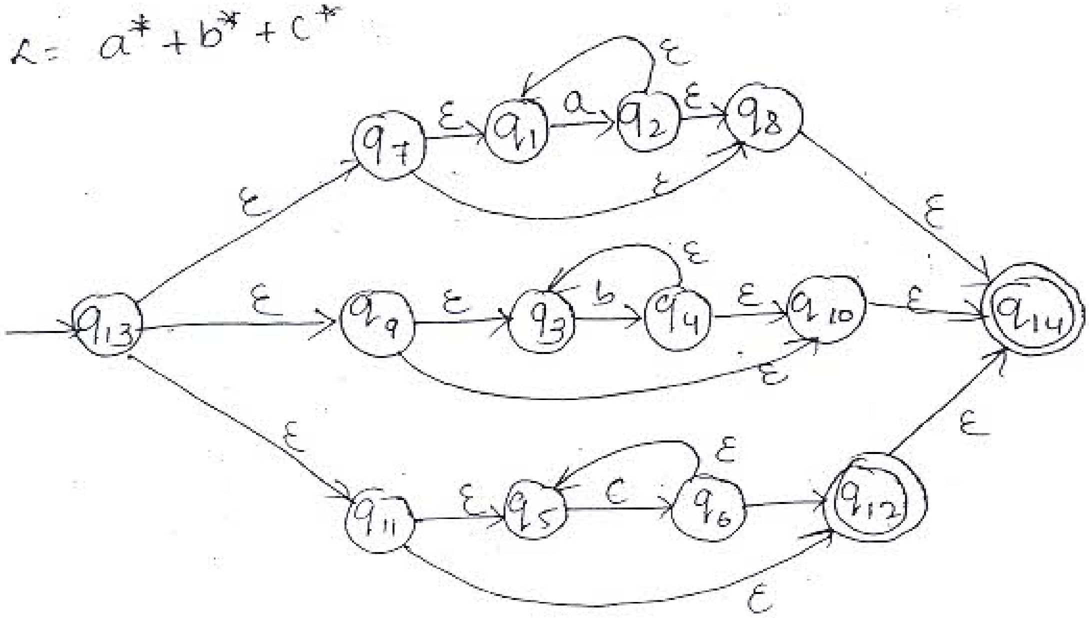
$L = b^*$



$L = c^*$



$L = a^* + b^* + c^*$



3)

Sol:

L

(a

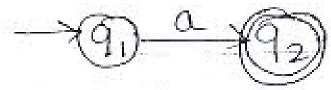
(a

a

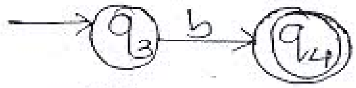
FLAT

3) $(a+bb)^*(ba^*+\lambda)$

Sol: $\lambda = a$

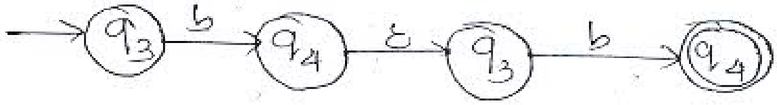


$\lambda = b$

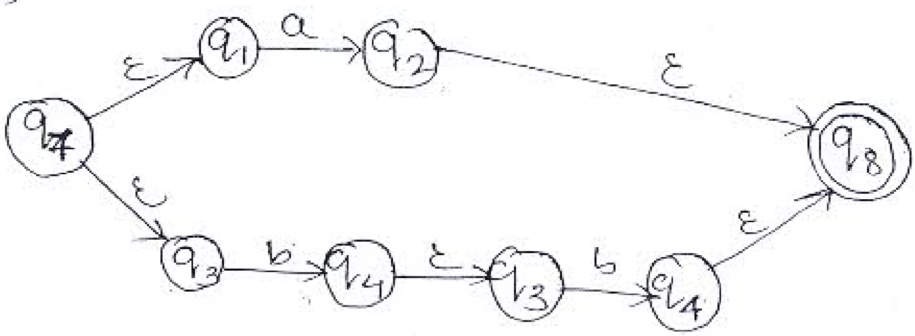


$\lambda = c$

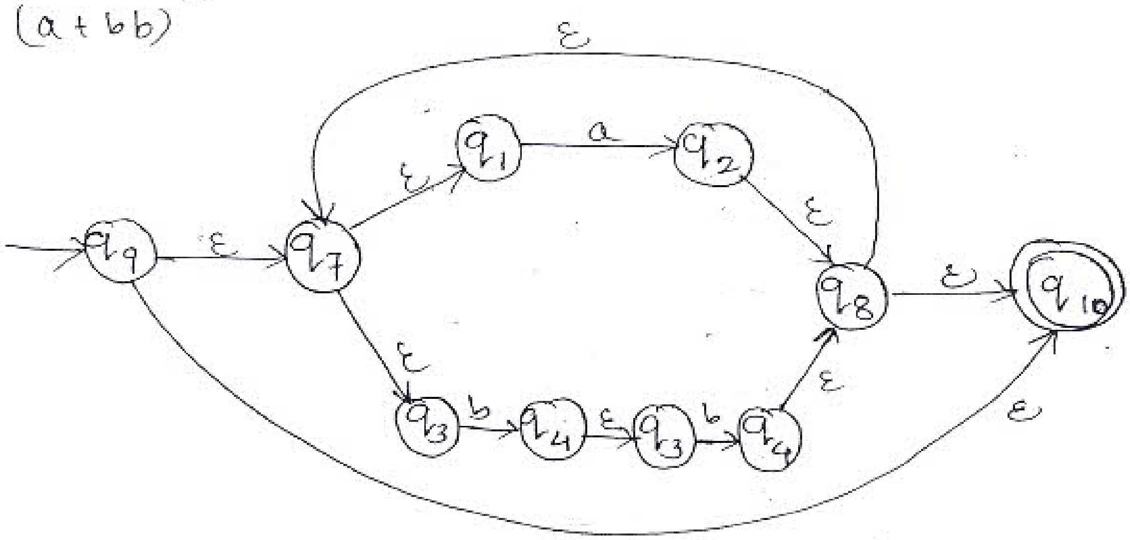
$\lambda = bb$



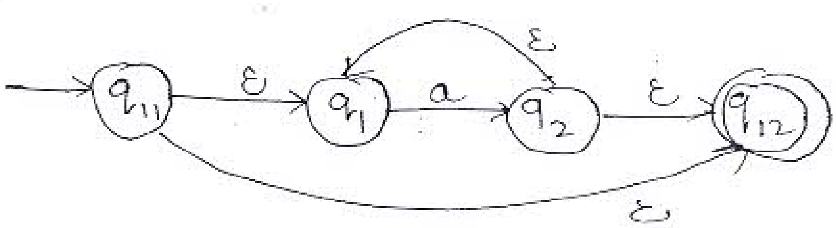
$(a+bb)$



$(a+bb)^*$

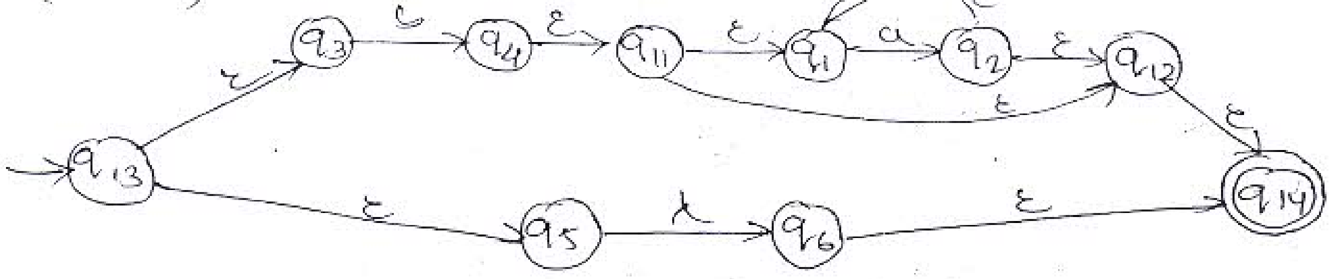


$a^* \Rightarrow$

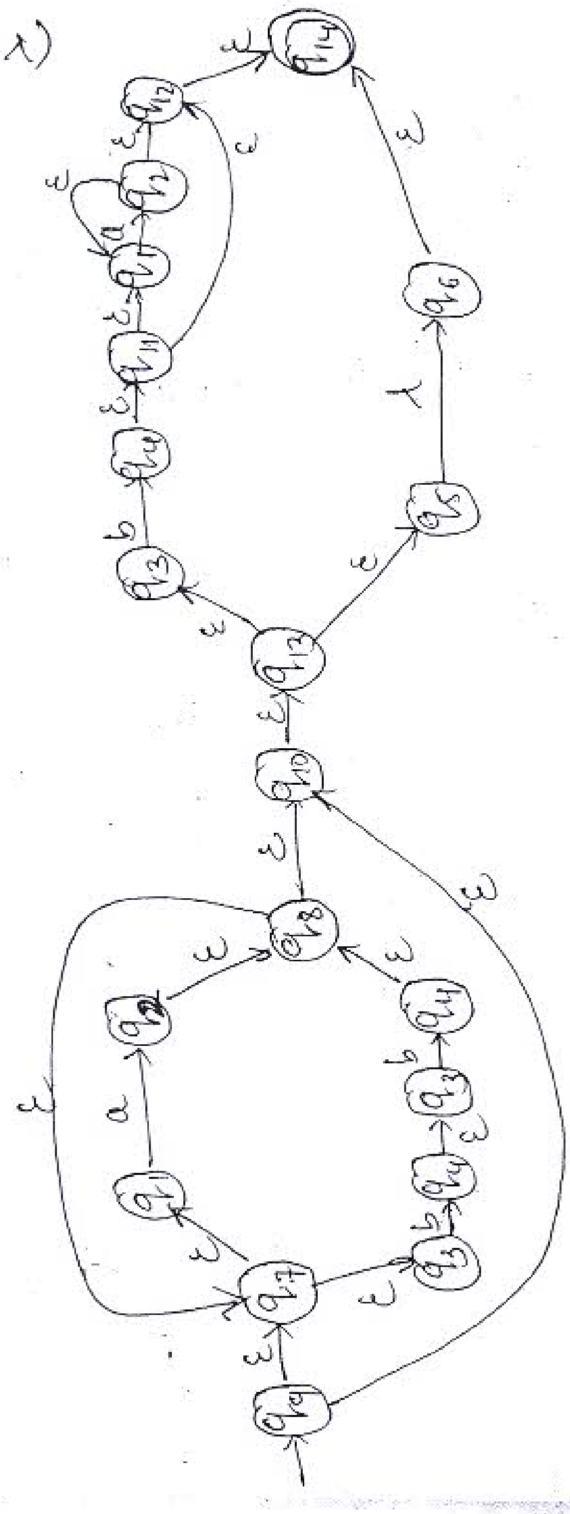


$ba^* \Rightarrow$

$(ba^*+\lambda)$



$\therefore$ $(a+bb)^*(ba^*+\lambda)$



* Algebraic laws of R.E

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

2) ϕ. anything $= \phi$

3) Commutative law on union, $L+M = M+L$

4) Associative law on union, $(L+M)+N = L+(M+N)$

5) Associative law for concatination, $L.(M.N) = (L.M).N$

6) Identifiers and annhilators:

$$\left.\begin{aligned} \phi + L &= L \\ \epsilon . L &= L \end{aligned}\right\} \text{identifier}$$

$\phi . L = \phi$, annhilator.

7) Distributive law, $L(M+N) = LM + LN$

$(L+M)N = LN + LM$

8) Idempotent law, $L+L = L$

9) Law's of closure

$(L^*)^* = L^*$

$(\phi)^* = \epsilon$

$L^+ = L.L^*$

$L^* = L^* + \epsilon$

$L? = L+\epsilon$

* Obtaining R.E from FA

R.E for an FA can be obtained using 2 diff method

1) Kleen's theorem

2) State Elimination Method

1) Kleens theorem :- (X)

P.T. if $L = L(M)$ for some $M = \{Q, \Sigma, \delta, q_0, F\}$ then there exist R.E Such that $L(M) = L(R)$

in other words,

If Language L is accepted by DFA then proove that there exist a regular expression R which accept the same language

Proof: Let $Q = \{q_1, q_2, q_3 \ldots q_n\}$ are the states of automata N where N is the no of state path from state i to state j through an intermediate state whose no is not greater than K is given by $R_{ij}^{(K)}$. ~~where i>j~~

A string w can be written as,

$w = xy$, $\hat{\delta}(i, x) \longrightarrow K$

$\hat{\delta}(K, y) \longrightarrow j$

Proof by Induction.

Basis part when $K = 0$, indicates that there is no intermediate state and the path from state i to state j is given by 2 conditions

Case 1) $i != j$, results in 3 cases

case 1) No input symbol between i and j

$$\boxed{R_{ij}^{(0)} = \phi}$$

case 2) One input symbol say 'a' on transition from i to j $\boxed{R_{ij}^{(0)} = a}$

case 3) There are multiple input symbols say 'a_1' 'a_2' 'a_3' ... a_m

$$\boxed{R_{ij}^{(0)} = a_1 + a_2 + a_3 + \cdots + a_m}$$

Case 2) $i = j$,

case 1> No transition from i to j

$$\boxed{R_{ij}^{(0)} = \varepsilon}$$

case 2> Input symbol say 'a'

$$\boxed{R_{ij}^{(0)} = \varepsilon + a}$$

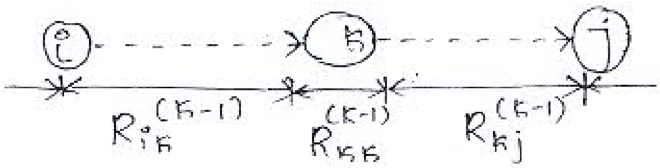
case 3> multiple i/p symbols

$$\boxed{R_{ij}^{(0)} = \varepsilon + a_1 + a_2 + a_3 + \cdots + a_m}$$

Induction part, when $k > 0$,
If there exist path from state i to j through a state which is not higher than k can be two cases

Case 1: There exist a path from state i to j which does not go through k and hence the language accepted $R_{ij}^{(k-1)}$

Case 2: There exist a path from state i to j which passes through k ~~and~~ as below.



The regular expression for the path from i to j wch passes through no state higher than k is given by,

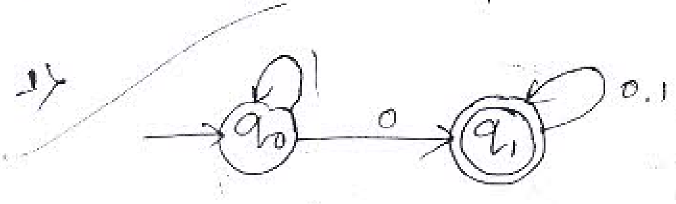
$$\boxed{R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \cdot \left(R_{kk}^{(k-1)}\right)^{*} \cdot R_{kj}^{(k-1)}}$$

By constructing the R.E in increasing order of subscript k i.e every $R_{ij}^{(k)}$ depends on the expression obtained for the smaller subscript values of k we can obtain the total R.E when we will have R_{ij}^{N} for all i & j where i is the start state & j is the final state

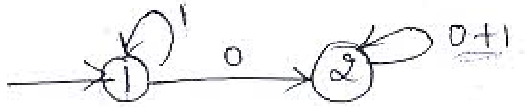
Problems:

Obtain the R.E for the following FA. Kleens

1)



	0	1
$\rightarrow q_0$	q_1	q_0
$* q_1$	q_1	q_1



Sol) $R_{ij}^{(k)}$, $k=0 \Rightarrow R_{ij}^{(0)}$ 30

	$k=0$	$k=1$
$R_{11}^{(0)}$	$\epsilon+1$	1^*
$R_{12}^{(0)}$	0	1^*0
$R_{21}^{(0)}$	ϕ	ϕ
$R_{22}^{(0)}$	$\epsilon+0+1$	$\epsilon+0+1$

S.P

$$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)} \cdot (R_{11}^{(0)})^* R_{11}^{(0)}$$
$$= (\epsilon+1) + (\epsilon+1)(\epsilon+1)^* (\epsilon+1)$$
$$= (\epsilon+1) + 1^*$$
$$= 1^*$$

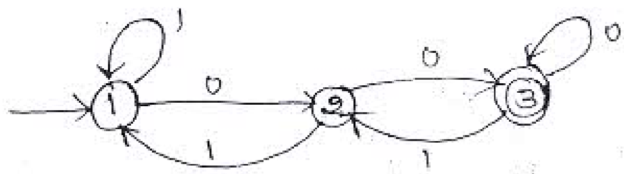
$$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$$
$$= 0 + (\epsilon+1)(\epsilon+1)^* (0)$$
$$= 0 + 1^*0$$
$$= 1^*0$$

$$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)} \cdot \left(R_{11}^{(1)}\right)^* \cdot R_{11}^{0}$$
$$= \phi + \phi \cdot (1^*)^* \cdot (\varepsilon+1)$$
$$= \phi$$

$$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{21}^{(0)}$$
$$= \varepsilon + 0 + 1 + \phi$$
$$= \varepsilon + 0 + 1$$

$$R_{22}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} \left(R_{22}^{(1)}\right)^* R_{22}^{(1)}$$
$$= 1^*0 + 1^*0 \, (\varepsilon+0+1)^* (\varepsilon+0+1)$$
$$= 1^*0 + 1^*0 \, (0+1)^*$$
$$= 1^*0 \, (0+1)^*$$

2) Convert the following FA to RE using Kleens theorem



S.P:

	K=0	K=1	K=2
$R_{11}^{(0)}$	$\varepsilon+1$	1^*	
$R_{12}^{(0)}$	0		
$R_{13}^{(0)}$	ϕ		
$R_{21}^{(0)}$	1		
$R_{22}^{(0)}$	ε		
$R_{23}^{(0)}$	0		
$R_{31}^{(0)}$	ϕ		
$R_{32}^{(0)}$	1		
$R_{33}^{(0)}$	$\varepsilon+0$		

$$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$$
$$= (\varepsilon+1) + (\varepsilon+1) \cdot (\varepsilon+1)^* (\varepsilon+1)$$

$(\varepsilon+1)+1$

$R_{11}^{(1)} = 1^*$

$$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{12}^{(0)}$$

$$= 0 + (\varepsilon+1)(\varepsilon+1)^* 0$$

$$= 1^* 0$$

$$R_{13}^{(1)} = R_{13}^{(0)} + R_{11}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{13}^{(0)}$$

$$= \phi + (\varepsilon+1)(\varepsilon+1)^* \cdot \phi$$

$$= \phi$$

$$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{11}^{(0)}$$

$$= 1 + 1(\varepsilon+1)^* (\varepsilon+1)^*$$

$$= 1 + 11^* \quad = 11^*$$

$$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* \cdot R_{12}^{(0)}$$

$$= \varepsilon + (1)(\varepsilon+1)^* (0)$$

$$= \varepsilon + 11^* 0$$

$$R_{23}^{(1)} = R_{23}^{(0)} + R_{21}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{13}^{(0)}$$

$$= 0 + 1 \cdot (\varepsilon+1)^* \cdot \phi$$

$$= 0 + \phi = 0$$

$$R_{31}^{(1)} = R_{31}^{(0)} + R_{31}^{(0)} \cdot \left(R_{(11)}^{(0)}\right)^* \cdot R_{11}^{(0)}$$

$$= \phi + \phi \cdot (\varepsilon+1)^* \cdot (\varepsilon+1)$$

$$= \phi$$

$$R_{32}^{(1)} = R_{32}^{(0)} + R_{31}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{12}^{(0)}$$
$$= 1 + \phi \cdot (\varepsilon+1)^* \cdot 0$$
$$= 1 + \phi \Rightarrow 1$$

$$R_{33}^{(1)} = R_{33}^{(0)} + R_{31}^{(0)} \cdot \left(R_{11}^{(0)}\right)^* \cdot R_{13}^{(0)}$$
$$= (\varepsilon+0) + \phi \cdot (\varepsilon+1)^* \cdot \phi$$
$$= (\varepsilon+0) + \phi$$
$$R_{33}^{(1)} = \varepsilon+0$$

$$R_{11}^{(2)} = R_{11}^{(1)} + R_{12}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{21}^{(1)}$$
$$= 1^* + (1^*0)(\varepsilon+11^*0)^* \cdot (11^*)$$
$$= 1^* + (1^*0)^* \cdot 1^* = (1^*0)\,1^*$$

$$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{22}^{(1)}$$
$$= 1^*0 + (1^*0)\,(\varepsilon+11^*0)^* \cdot (\varepsilon+11^*0)$$
$$= 1^*0 + (1^*0)\,(11^*0)^* \rightarrow$$

$$R_{13}^{(2)} = R_{13}^{(1)} + R_{12}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{23}^{(1)}$$
$$= \phi + (1^*0) \cdot (\varepsilon+11^*0) \cdot 0$$
$$= 1^*0(11^*0)0.$$

$$R_{21}^{(2)} = R_{21}^{(1)} + R_{22}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{21}^{(1)}$$
$$= 11^* + (\varepsilon+11^*0) \cdot (\varepsilon+11^*0)^* \cdot (11^*)$$
$$= (11^*0)^*\,11^*$$

$$R_{22}^{(2)} = R_{22}^{(1)} + R_{22}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{22}^{(1)}$$

$$= (\varepsilon + 11^*0) + (\varepsilon + 11^*0)(\varepsilon + 11^*0)^* \cdot (\varepsilon + 11^*0)$$

$$= (11^*0)^*$$

$$R_{23}^{(2)} = R_{23}^{(1)} + R_{22}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{23}^{(1)}$$

$$= \cancel{11^* + (11^*) \cdot (\varepsilon + 11^*0)^* \cdot (11^*)}$$

$$= 0 + (\varepsilon + 11^*0) \cdot (\varepsilon + 11^*0)^* \cdot 0$$

$$= (11^*0)^*0$$

$$R_{31}^{(2)} = R_{31}^{(1)} + R_{32}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{21}^{(1)}$$

$$= \phi + (1) \cdot (\varepsilon + 11^*0)^* \cdot (11^*).$$

$$= 1(11^*0)^* \cdot (11^*)$$

$$R_{32}^{(2)} = R_{32}^{(1)} + R_{32}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{22}^{(1)}$$

$$= 1 + (1)(\varepsilon + 11^*0)^* \cdot (\varepsilon + 11^*0)$$

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

$$= 1(11^*0)^*$$

$$R_{33}^{(2)} = R_{33}^{(1)} + R_{32}^{(1)} \cdot \left(R_{22}^{(1)}\right)^* \cdot R_{23}^{(1)}$$

$$= (\varepsilon + 0) + (1)(\varepsilon + 11^*0)^* (0)$$

$$= \varepsilon + 0 + 1(11^*0)^*0$$

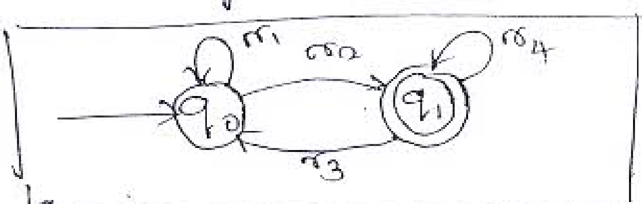
$$R_{13}^{(3)} = R_{13}^{(2)} + R_{13}^{(2)} (R_{33}^{(2)})^* \cdot R_{33}^{(2)}$$

$$= 1^*0(11^*0)^*0 + 1^*0(11^*0)^*0(0+1(11^*0)^*0)^* \cdot (0+1(11^*0)^*0)$$

$$= \underline{1^*0(11^*0)^*0(0+1(11^*0)^*0)^* 1^*0}.$$

③ Conversion of FA to RE using State elemination method

Let $M = \{Q, \Sigma, \delta, q_0, F\}$ be a dfa recognizing the language L equivalent to a regular expression R such that $L = L(R)$. We can obtain the R.E. R by eliminating the states of FA to the generalized transaction graph as below.

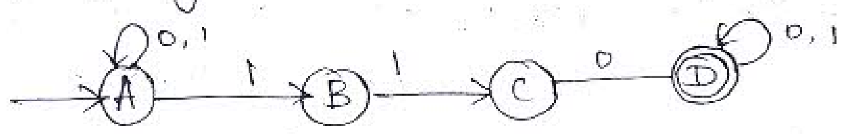


The R.E. for the above generalized automata is given as $\boxed{R = r_1^* r_2 (r_4^* + r_3 r_1^* r_2)^*}$

For each final state apply the reduction procedure saperately and bring the graph to the generalized form. union of such R.E obtained is the language accepted by the automata. if start state is also an accepting state then the eliminam process has to be performed such that all state except start state is eliminated and final generalized form is as follows.

$\boxed{R = r^*}$

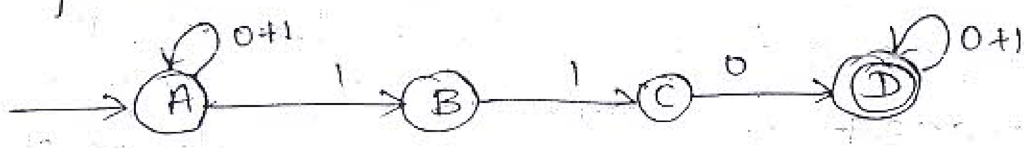
Problem 1) obtain R.E for following automata using state elemination method



Sol) 1) Remove all the trap state if any.

No trap state in the given automata

2) There are multiple input symbol in transition represent them as R.E



3) Eliminate intermediate state 1 by 1 but ensure the regular expression which can be obtained passing through the intermediate state as it is

eliminating the state B.

$A \xrightarrow{B} C = 11$

$A \xrightarrow{B} A = \phi$

$C \xrightarrow{B} C = \phi$

$C \xrightarrow{B} A = \phi$

automata after removing state B

Eliminate the state C.

$A \xrightarrow{C} D = 110$

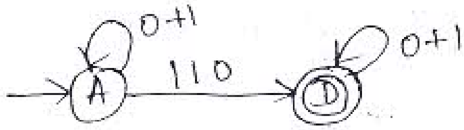
$A \xrightarrow{C} A = \phi$

$D \xrightarrow{C} A = \phi$

$D \xrightarrow{C} D = \phi$

FLAT

ition



is the generalized automata

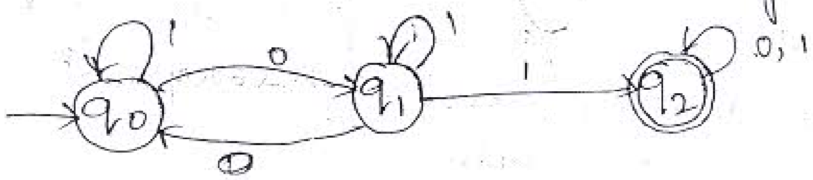
$r_1 = 0+1$, $r_2 = 110$, $r_3 = \phi$, $r_4 = 0+1$

$$R = r_1^* r_2 (r_4^* + r_3 r_1^* r_2)^*$$
$$= (0+1)^* 110 ((0+1)^* + \phi \cdot (0+1)^* \cdot 110)^*$$
$$= (0+1)^* 110 ((0+1)^* + \phi)^*$$
$$= (0+1)^* 110 ((0+1)^*)^*$$
$$= (0+1)^* 110 (0+1)^*$$

se

21) Obtain the R.E for the following automata

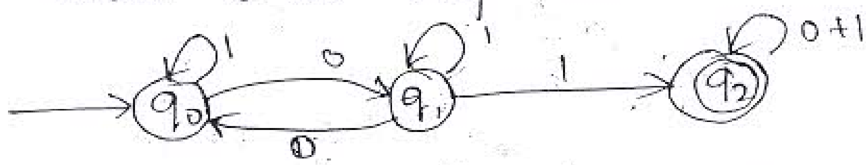
Sol:



state

Sol: Remove all the trap state if any

There is no trap state in this automata



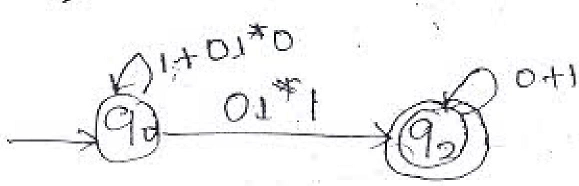
Eleminate the intermediate q_1

$q_0 \xrightarrow{q_1} q_2 = 01^*1$

$q_0 \xrightarrow{q_1} q_0 = 01^*0$

$q_2 \xrightarrow{q_1} q_2 = \phi$

$q_2 \xrightarrow{q_1} q_0 = \phi$



$r_1 = 1 + 01^*0$

$r_2 = 01^*1$

$r_3 = \phi$

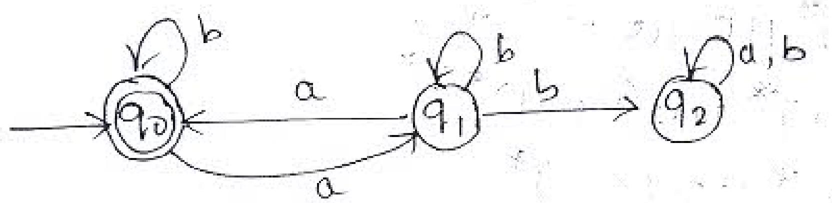
$r_4 = 0+1$

$R = (1+01^*0)^* 01^*1 \left((0+1)^* + \phi \cdot 0+01^*0 \cdot 01^*1\right)^*$

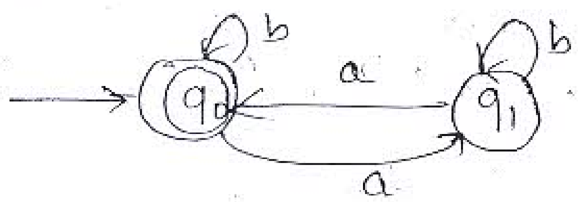
$= (1+01^*0)^* 01^*1 \left((0+1)^* + \phi\right)^*$

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

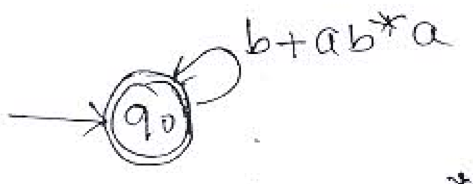
$= (1+01^*0)^* 01^*1 (0+1)^*$

3)  (state eleminam method)

Sol) remove the trap state if any
there q_2 is the trap state bcz it has no state going through it & it is not final state

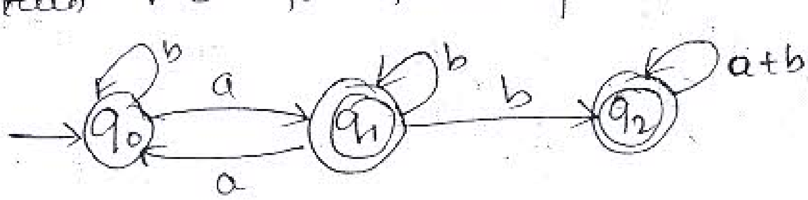


$q_0 \xrightarrow{q_1} q_0 = ab^*a$ [if start state & final state are same



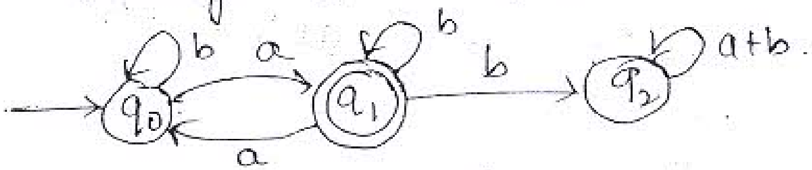
$R = (b+ab^*a)^*$

4) obtain R.E for following FA.



Sol^n: Case 1

Taking q_1 as final state



remove the q_2 which is a trap state

$r_1 = b \quad r_2 = a$

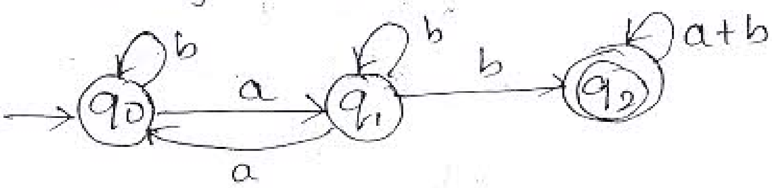
$r_3 = a \quad r_4 = b$

$$q_0 \xrightarrow{q_1} q_0 = b^*a(b^*+ab^*a)^*$$

$$R_1 = b^*a(b^*+ab^*a)^*$$

Case 2

Taking q_2 as final state



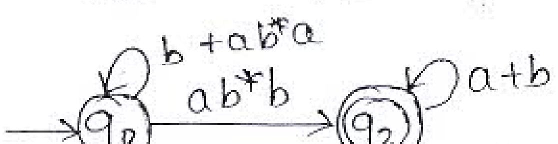
remove state q_1.

$$q_0 \xrightarrow{q_1} q_2 = ab^*b$$

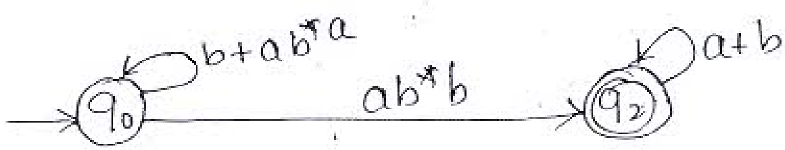
$$q_0 \xrightarrow{q_1} q_0 = ab^*a$$

$$q_2 \xrightarrow{q_1} q_0 = \phi$$

$$q_2 \xrightarrow{q_1} q_2 = \phi$$



=>

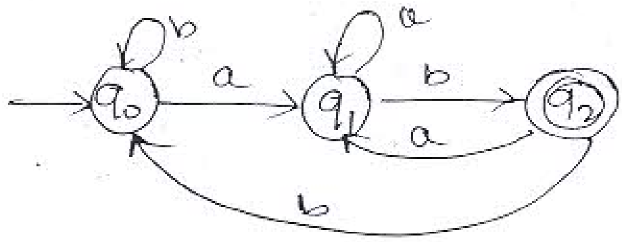


$r_1 = b+ab^*a \quad r_2 = ab^*b \quad r_3 = \phi \quad r_4 = a+b$

$$R_2 = (b+ab^*a)^* \; ab^*b(a+b)^*$$

$r = r_1 . r_2$

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

5) 

obtain the R.E.

Soln: There is no ~~trap~~ state here.

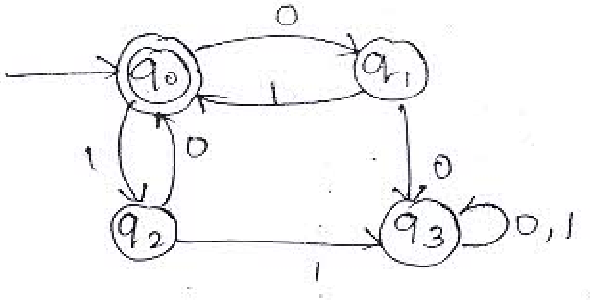
remove q_1

$$q_0 \xrightarrow{q_1} q_2 = aa^*b$$
$$q_0 \xrightarrow{q_1} q_0 = \phi$$
$$q_2 \xrightarrow{q_1} q_0 = \cancel{aa^*a} \ \phi$$
$$q_2 \xrightarrow{q_1} q_2 = \cancel{ab^*b} \ aa^*b$$

$r_1 = b \qquad r_2 = aa^*b \qquad r_3 = \phi b, \ r_4 = aa^*b.$

$$R = b^*aa^*b((aa^*b)^* + \phi b(b)^* aa^*b)^*$$
$$= b^*aa^*b((aa^*b)^* + \phi b)^*$$
$$= b^*aa^*b((aa^*b)^* + b)^*$$
$$= b^*aa^*b(aa^*b + b)^*$$
$$= b^*aa^*b(aa^*b + b)^*$$

6) obtain R.E for the following.



Sol:

Properties of regular Language

Regular languages are language which are excepted by any of the automata which can be described using regular expression.

The important classes of regular languages are plumbing lema: This property is used to prove certain languages are not regular

Closure property: These property are used to build complex RL. applying certain operation of regular language

Decision property: It is used to decide whether 2 automata define the same lang. by minimising them.

1. Decision property: If two automatas except the same languages. minimization of same automata will result in automata with same no of states and exactly the same transition In such case FA's are said to be equivalent.

Distiquishable and Indistinguishable:

Two states p and q of a dfA are Indistinquicable if and only if both are final state or both are non final state.

$\hat{\delta}(Q, w) \in F$ & $\hat{\delta}(P, w) \in F$

or

$\hat{\delta}(Q, w) \notin F$ & $\hat{\delta}(P, w) \notin F$

Two state are said to be distiquisable

if one of them is a final state and other one is non final state.

$\hat{\delta}(q,w) \in F$ & $\hat{\delta}(p,w) \notin F$

$\hat{\delta}(Q,w) \in F$ & $\hat{\delta}(p,w) \in F$

minimization of DFA with table filling algorithm or mark filling algorithm

ies to Imp* 4

Table filling algorithm it is used to find the set of state that are indistinguishable and State which are distinguishable which can be used further, to minimize the given DFA as few state as possible

Step 1: Identify the initial marking

for each pair of state (p, q) where $p, q \in Q$ if $p \in F$ & $q \notin F$ or versa the pair (p, q) is distinguishable and hence mark the pair p, q

Step 2: identify Subsequent marking

for each unmarked pair p, q & each input $a \in \Sigma$ find $\delta(p,a) \to r$ and $\delta(q,a) \to s$

and if pair (r, s) is already marked and mark the pair p, q also

- Repeat the step 2 untill no previously unmarked pair is marked

Step 3: identifying distinguishable and Indistinguishable states

All the pair of state which are un marked after step 1 & 2 are indistinguishable step & remaining are individual distinguishable state

Step 4: ~~identifying distinguishable~~

Computing transition table.

If group {p1, p2 ... pk} is a indistinguishable group of state and if $\delta(\{p_1, p_2 \ldots p_k\}, a) = \{r_1, r_2 \ldots r_m\}$ then place an edge a from $(p_1, p_2 \ldots p_k)$ to $(r_1, r_2 \ldots r_m)$.

Follows the same for each groups of indistinguishable state or individual distinguishable state w.r.t. each i/p symbol or Σ

Step 5: Identify the state and that state if one of the component in a groups say $p_1, p_2 \ldots p_k$ is the start state given DFA, then the group itself is the ~~state~~ start state & same is applicable for final state also.

Problem

1. Obtain distinguishable table for given automata then minimize no of state of DFA

Sol:

δ	a	b
→A	B	F
B	G	C
C*	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

Sol:

	A	B	C	D	E	F	G
B	X						
C*	X	X					
D	X	X	X				
E		X	X	X			
F	X	X	X		X		
G	X	X	X	X	X	X	
H	X		X	X	X	X	X

final state

* final state

	a	b
(A, B)	(B, G)	(F, C)
(A, D)	(B, C)	—
(A, E)	(B, H)	(F, F)
(A, F)	(B, C)	
(A, G)	(B, G)	(F, E)
(A, H)	(B, G)	(F, C)
(B, D)	(G, C)	—
(B, E)	(G, H)	(C, F)
(B, F)	(G, C)	—
(B, G)	(G, G)	(C, E)
(B, H)	(G, G)	(C, C)

	a	b
(D, E)	(C, H)	—
(D, F)	(C, C)	(G, G)
(D, G)	(C, G)	—
(D, H)	(C, G)	—
(E, F)	(H, C)	—
(E, G)	(H, G)	(F, E) -
(E, H)	(H, G)	(F, C)
(E, G)	(C, G)	—
(F, H)	(C, G)	—
(G, H)	(G, G)	(E, C)

(2)

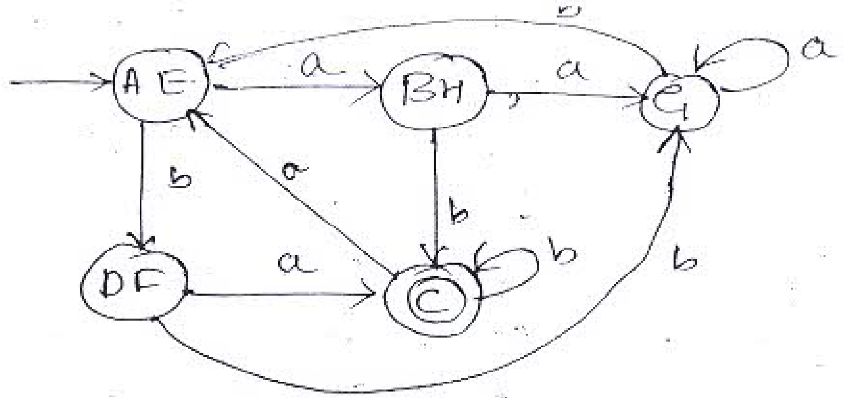
	a	b
(A, E)	(B, H)	(F, F)
(A, G)	(B, G)	(F, E)
(B, H)	(G, G)	(C, C)
(D, F)	(C, C)	(G, G)

(3)

	a	b
(A, E)	(B, H)	(F, F)
(B, H)	(G, G)	(C, C)
(D, F)	(C, C)	(G, G)

The pair (A, E) (B, H) (D, F) are indistinguishable pair of states and C, G are distinguishable state

	a	b
→ (A E)	(B H)	(D F)
(B, H)	G	C
(D, F)	C	G
* C	(A E)	C
G	G	(A E)



Q1) Minimize the following DFA using table filling algorithim

Sol:

δ	0	1
$\rightarrow$ A	B	E
B	C	F
C*	D	H
D	E	H
E	F	I
F*	G	B
G	H	B
H	I	C
I*	A	E

Sol:

B	X							
* C	X	X						
D		X	X					
E	X		X	X				
* F	X	X		X	X			
G		X	X		X	X		
H	X		X	X		X	X	
* I	X	X		X	X		X	X
	A	B	C *	D	E	F *	G	H

①

	0	1
(A B)	BC	—
(A D)	BE	EH
(A E)	BF	—
(A G)	BH	EB
(A H)	BI	—
(B D)	CE	—
(B E)	CF	FI
(B G)	CH	—
(B H)	CI	FC
(C F)	DG	HB
(C I)	DA	HE
(D E)	EF	—
(D G)	EH	HB
(D H)	EI	—
(E G)	FH	—
(E H)	FI	IC
(F I)	GA	BE
(G H)	HI	—

②

	0	1
(A D)	BE	EH
(A G)	BH	EB
(BE)	CF	FI
(BH)	CI	FC
(CF)	DG	HB
(CI)	DA	HE
(DG)	EH	HB
(EH)	FI	IC
(FI)	GA	BE

③ (AD) (AG) (DG) = (ADG)

(BE) (BH) (EH) = (BEH)

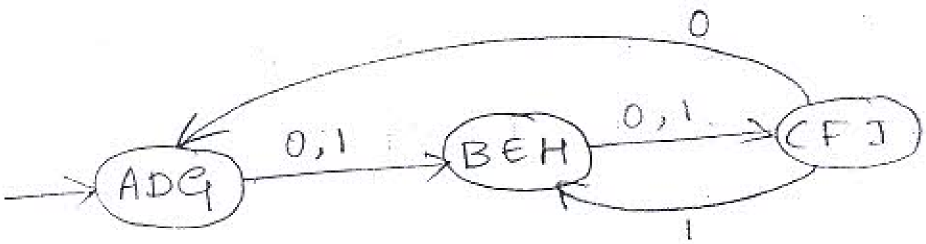
(CF) (CI) (FI) = (CFI)

There are no distinguishable state in this DFA & there are three indistinguishable state in it.

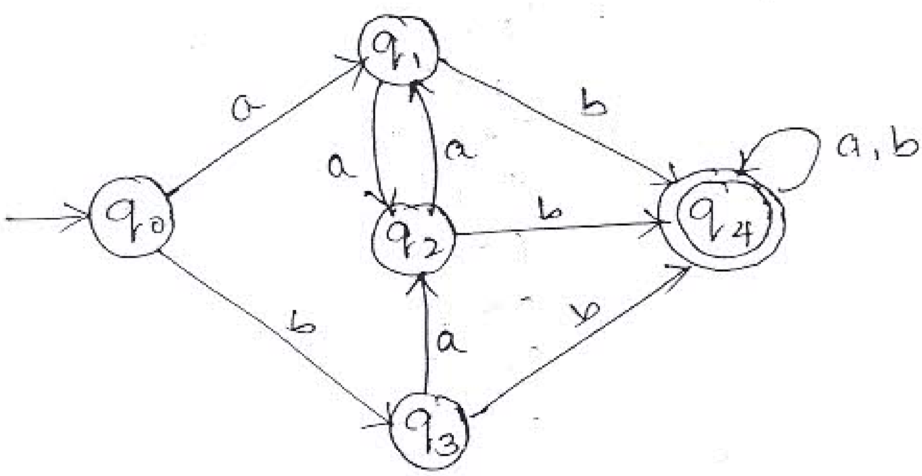
DFA i.e (ADG) ...

The transition for these states can be as below.

	0	1
→ ADG	BEH	EHB
BEH	CFI	FIC
* CFI	DGA	HBE



3) Minimize the following DFA.



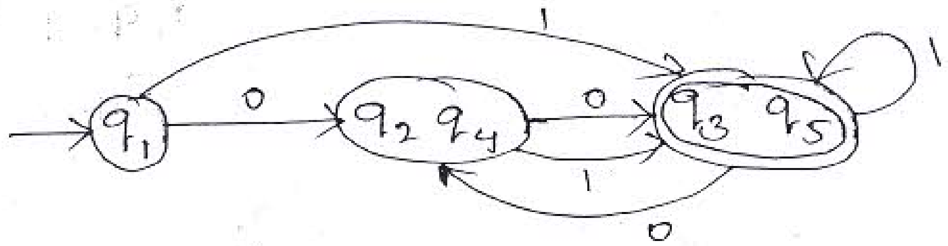
Sol:

	a	b
→ 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	1
(q_1, q_2)	$q_2 q_3$	—
(q_1, q_4)	$q_2 q_3$	—
$(q_2 q_4)$	$q_3 q_3$	$q_5 q_5$
$(q_3 q_5)$ *	$q_4 q_2$	$q_3 q_5$

	0	1
$q_2 q_4$	$q_3 q_3$	$q_5 q_5$
* $q_3 q_5$	$q_4 q_2$	$q_3 q_5$
→ (q_1)		

∴ $(q_2 q_4), (q_3 q_5), q_1$



5)

δ	a	b
→ A	B	A
B	A	C
C	D	B
D *	D	A
E	D	F
F	G	E
G	F	G
H	G	D

q_1	X			
q_2	X			
q_3	X			
* q_4	X	X	X	X
	q_0	q_1	q_2	q_3

①

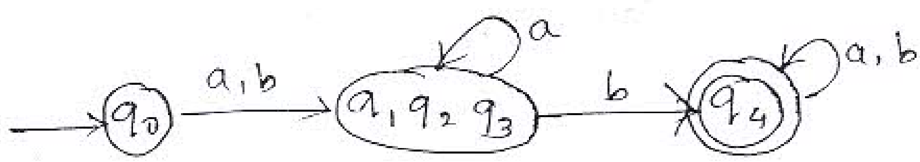
	a	b
$q_0\ q_1$	$q_1 q_2$	$q_3 q_4$
$q_0\ q_2$	$q_1 q_1$	$q_3 q_4$
$q_0\ q_3$	$q_1 q_2$	$q_3 q_4$
$q_1\ q_2$	$q_2 q_1$	$q_4 q_4$
$q_1\ q_3$	$q_2 q_2$	$q_4 q_4$
$q_2\ q_3$	$q_1 q_2$	$q_4 q_4$

②

	a	b
$q_1\ q_2$	$q_2 q_1$	$q_4 q_4$
$q_1\ q_3$	$q_2 q_2$	$q_4 q_4$
$q_2\ q_3$	$q_1 q_2$	$q_4 q_4$

③

$$(q_1 q_2)(q_1 q_3)(q_2 q_3) = (q_1 q_2 q_3), q_0, q_4.$$



4)

δ	0	1
→ q_1	q_2	q_3
q_2	q_3	q_5
q_3 *	q_4	q_3
q_4	q_3	q_5
q_5 *	q_2	q_5

Sol^n

q_2	X			
* q_3	X	X		
q_4	X		X	
* q_5	X	X		X
	q_1	q_2	q_3 *	q_4

B	X						
C	X	X					
D	X	X	X				
E	X	X		X			
F	X		X	X	X		
G		X	X	X	X	X	
H	X	X	X	X	X	X	X
	A	B	C	D	E	F	G

①

	a	b
AB	BA	AC
~~AC~~	BD	—
~~AE~~	BD	—
AF	BG	AE
AG	BF	AG
~~AH~~	BG	AD
~~BC~~	AD	—
~~BE~~	AD	—
BF	AG	CE
BG	AF	CG
~~BH~~	AG	CD
CE	DD	BF
~~CF~~	DG	—
~~CG~~	DF	—
~~CH~~	DG	—
~~EF~~	DG	—
~~EG~~	DF	—
~~EH~~	DG	—
FG	GF	EG
~~G~~FH	GG	ED
GH	FG	GD

②

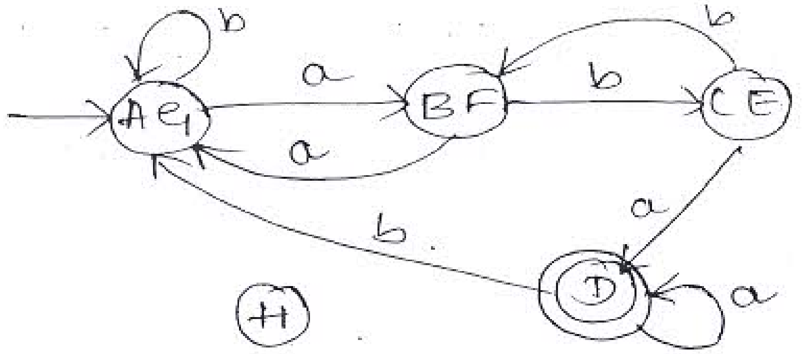
δ	a	b
~~AB~~	BA	AC
~~AF~~	BG	AE
AG	BF	AG
BF	AG	CE
~~BG~~	AF	CG
CE	DD	BF
~~FG~~	GH	EG
~~FH~~	GG	ED
~~GH~~	FG	GD

③

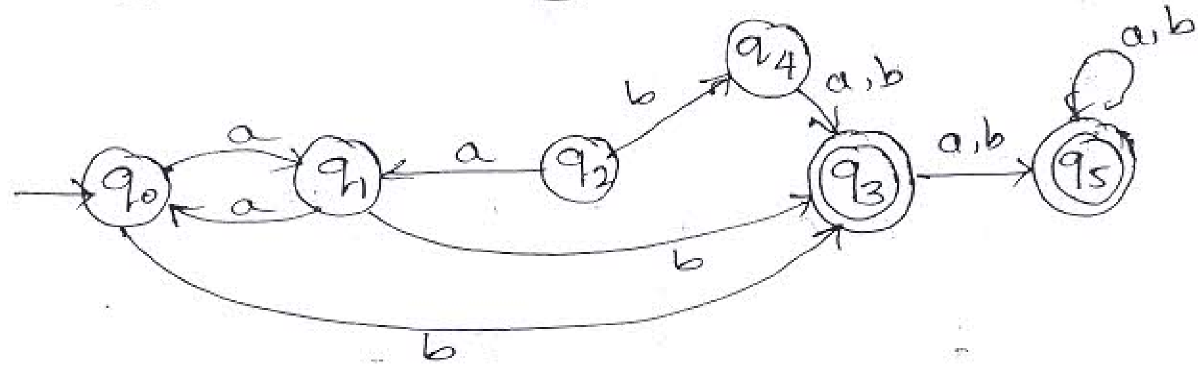
δ	a	b
AG	BF	AG
BF	AG	CE
CE	DD	BF

∴ (AG), (BF), (CE), D, H.

(H)



(G)



Sol:

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3
q_2	q_1	q_4
q_3^*	q_5	q_5
q_4	q_3	q_3
q_5^*	q_5	q_5

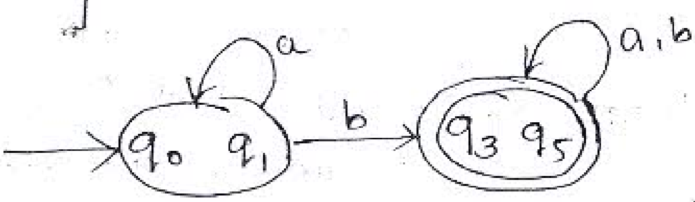
q_1					
q_2	X	X			
* q_3	X	X	X		
q_4	X	X	X	X	
* q_5	X	X	X		X
	q_0	q_1	q_2	q_3 *	q_4

16'

a,b

	a	b
$q_0 q_1$	$q_1 q_0$	$q_3 q_3$
~~q_0~~ q_2	$q_1 q_1$	$q_3 q_4$
~~q_0~~ q_4	$q_1 q_3$	—
~~q_1~~ q_2	$q_0 q_1$	$q_3 q_4$
~~q_1~~ q_4	$q_0 q_3$	—
~~q_2~~ q_4	$q_1 q_3$	—
$q_3 q_5$	$q_5 q_5$	$q_5 q_5$

	a	b
$\rightarrow q_0 q_1$	$q_1 q_0$	$q_3 q_3$
$* q_3 q_5$	$q_5 q_5$	$q_5 q_5$
q_2	q_1	q_4
q_4	q_3	q_3



Closure properties of Regular language

Operations using which new regular languages can be constructed using the existing ones such as union, intersection etc. these operations which can be applied on regular languages which results in another regular language are known as closure properties of regular language.

Various properties of R.E are below.

1. Union
2. Concatination
3. Closure
4. Intersection
5. Complement
6. Difference
7. Reversal
8. Homomorphism
9. Inverse homomorphism.

Theorem 1

If L_1 and L_2 are the regular languages we can proove that $L_1 \cup L_2$, $L_1 \cdot L_2$ & L_1^* are also r.L.

or

P.T regular language are closed under union, concatination and closure operation.

Proof: It is given that L_1 and L_2 are the R.L. and hence there exist regular expressions R and S Such that $L_1 = L(R)$ and $L_2 = L(S)$

By the defn of RE if R and S are two regular expressions then $R+S$, $R \cdot S$ & R^* are also R.E. which denotes a language $L(R) \cup L(S)$, $L(R) \cdot L(S)$ & $L(R)^*$ respectively. and hence the proof.

Theorem 2

Closure under Compliment

If L is regular language then P.T $\bar{L}$ is also regular language.

Proof: It is given that language L is regular and hence Construct an FA $M = \{Q, \Sigma, \delta, q_0, F\}$ which accepts the language L

Construct another FA

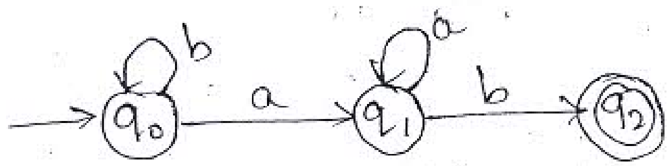
$M' = \{Q, \Sigma, \delta, q_0, Q-F\}$ which accepts all strings rejected by M. ie $\bar{L}$

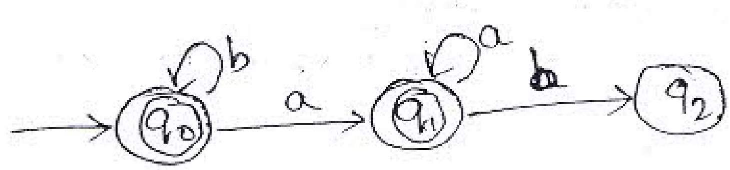
Since we can construct and FA for L It is a R.E.

Hence the proof

Ex: Construct an FA for accepting all the sub string which has ab in it.

& construct an FA for accepting all the sub string which donot have ab in it.





Closure under reversal

Theorem: (9)

If L is a regular language then L^R is also a regular language

PT. RL are closed under reversal operation

Proof: Let $M = \{Q, \Sigma, \delta, q_0, F\}$ which accept the language L since it is regular language we can construct another FA $M' = \{Q, \Sigma, \delta', q_0^f, F\}$ by reversing all the arcs of the DFA M thus construct an FA M' accepts all the strings of language L in reverse i.e it accepts the language $L(R)$ and hence $L(R)$ is also regular.

Ex: divisible 5 & reverse.

Closure under intersection

Theorem:

Show that if L_1 and L_2 are regular language hence $L_1 \cap L_2$ is a regular language

Proof: Let $M_1 = \{Q_1, \Sigma, \delta_1, q_1, F_1\}$ &

$M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2\}$ are the dfa accepting the language L_1 and L_2 respectively with same set of alphabet Σ

we can construct another FA

$M = \{Q, \Sigma, \delta, q_0, F\}$

where state of the machine M are the pairs (p, q) where $p \in Q_1$
$q \in Q_2$

The transition for automata M from state pq on input symbol A can be defined as below.

$$\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$$
$$= (r, s)$$

By this kind of transition function the FA, M stimulates the affect of both the languag L_1 & L_2 & it will accept all strings accepted by M_1 & M_2

The final DFA constructed will be as below

Ex:

$M = \{Q, \Sigma, \delta, q_0, F\}$

where,

$Q = Q_1 \times Q_2$

$\Sigma = \Sigma$

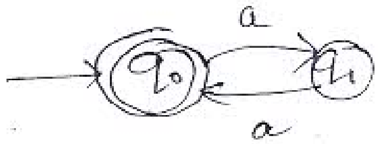
$\delta = \delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$

$q_0 = (q_1, q_2)$

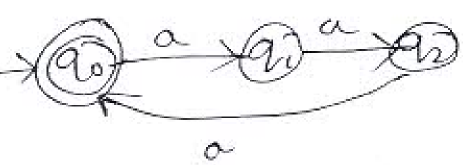
$F = \{(p, q) \in F$ iff $p \in F_1$ & $q \in F_2$

Ex: Construct a dfa to accept the language L_1 & L_2. regular languages are closed under intersection

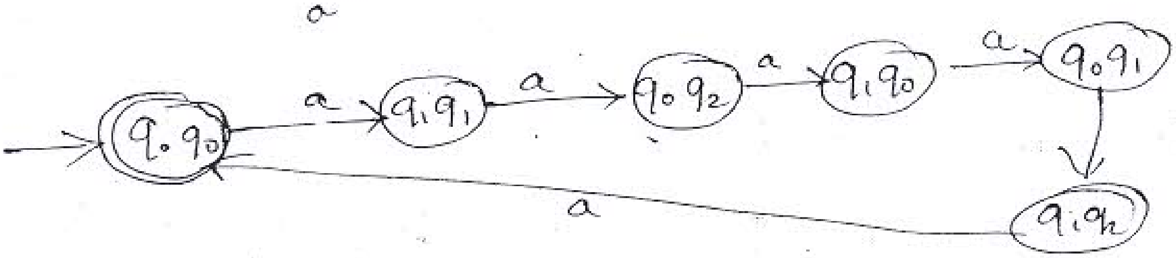
$L = \{w \mid n_a(w) \bmod 2 = n_a(w) \bmod 3 = 0\}$ $\Sigma = \{a\}$



$n_a(w) \bmod 2 = 0$



$n_a(w) \bmod 3 = 0$



② Difference

Theorem: Show that if L_1 and L_2 are two regular languages, the $L_1 - L_2$ is also regular

Proof:

we can construct FA $M_1 = \{Q_1, \Sigma, \delta_1, q_1, F_1\}$ & $M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2\}$ representing the language L_1 and L_2 respectively.

we can also define an FA $M = \{Q, \Sigma, \delta, q_0, F\}$

where $Q = Q_1 \times Q_2$

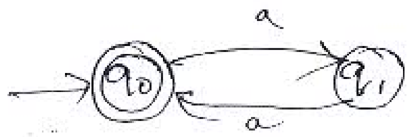
$\Sigma = \Sigma$

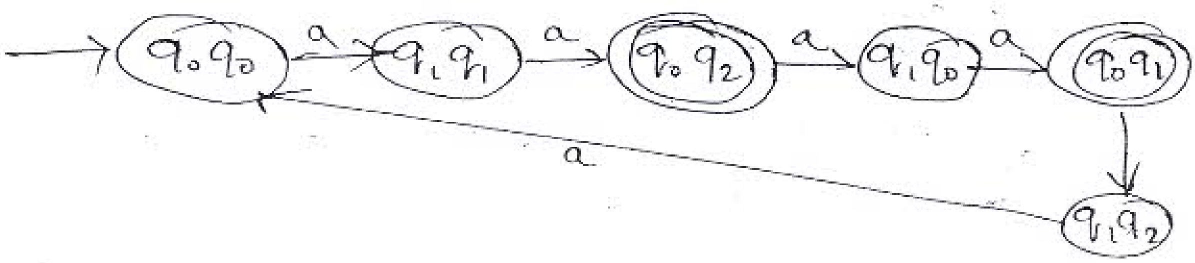
$\delta = (\delta(p,q), a) = (\delta_1(p,a), \delta_2(q,a))$

$q_0 = \{q_1, q_2\}$

$F = \{(p,q) \ \forall \ p \in F_1, q \notin F_2$

Ex: $L = \{w \mid n_a(w) \bmod 2 = n_a(w) \bmod 3 = 0 \quad \Sigma = \{a\}\}$





Homomorphism:

Defn: Let Σ and T are the two set of alphabets with the homomorphic function h

$h: \Sigma \to T^*$

ie a single letter from Σ is substituted by string of T

If $w = a_1 a_2 \ldots a_n$ then,

$h(w) = h(a_1) \cdot h(a_2) \ldots h(a_n)$

If a language L is made of alphabet from Σ then $h(L) = \{h(w) \mid w \in L\}$

is called as homomorphic image of language L

Ex: Let $\Sigma = \{0,1\}$

$T = \{a, b, c\}$ and $h(0) = aba$

$h(1) = bc$

what is the value of $h(010)$

$\Rightarrow$ $h(010) = h(0)\,h(1)\,h(0)$

$= aba\,bc\,aba$

2) $h(01^*(0+1)^*)$

Theorem: PT Language L is regular over an alphabet Σ & h is a homomorphic function on Σ then $h(L)$ is also regular.

Or PT regular languages are closed under homomorphism.

Proof: Let E be a regular expression with symbols in Σ and $h(E)$ is the expression obtained by replacing each symbol 'a' of Σ in E by $h(a)$

proof by inductn

Basic part: If $E = \phi$ or ϵ

then $h(E)$ is defined as below.

$h(\phi) = \phi$

$h(\epsilon) = \epsilon$

i.e homomorphic fn doesnot effect ϕ & ϵ

thus $L(h(E)) = h(L(E))$

If $E = a$ for some symbol a in Σ i.e.

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

$h(L(E)) = \{h(a)\}$

$h(E)$ is a regular expression obtained by applying homomorphism on $h(a)$ Since,

$E = a$ and thus,

$L(h(E)) = \{h(a)\}$ thus,

$L(h(E)) = h(L(E))$

Induction part: Let $E = F + G$ where E, F, G are regular expressions then, ~~h(E) =~~

$h(E) = h(F+G) = h(F) + h(G)$

we also know that

$L(E) = L(F) \cup L(G)$

hence, $L(h(E)) = \{L(h(F)) + L(h(G))\}$

$= L(h(F) + h(G))$

$L(h(F)) \cup L(h(G))$

By the defn of union operation +

$h(L(E)) = h(L(F) \cup L(G))$ — ①

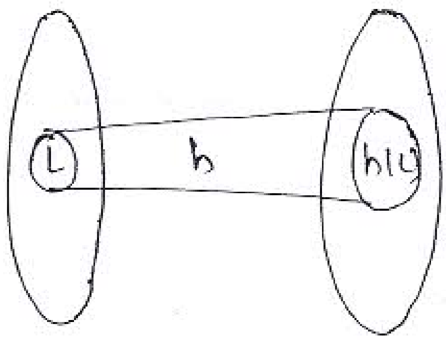
$= h(L(F)) \cup h(L(G))$ — ②

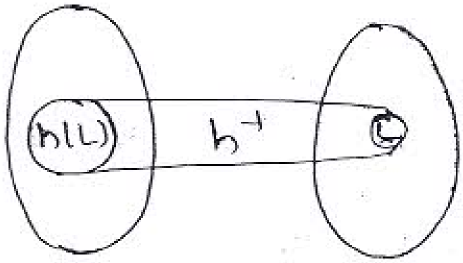
where homomorphic function h is applied to a language by applying it to each of its string individually and hence $L(h(E)) = h(L(E))$ & same is applicable for concatination and closure operation.

<u>Inverse homomorphism:</u>

It is a homomorphism applied backwords.

This can be shown symbolically as below





The homomorphic language is obtained by applying homomorphic fn on a regular language. doing an inverse homomorphism on a homomorphic language to give back a original regular language and hence RL is also closed under inverse homomorphism

Imp. *** (V) Pumping Lemma of Regular languages:

Pegion hole principle says that if we have more pegions than the number of pegion holes then there must be at least one pegion hole which has more than one pegion.

principle theorem (pumping lemma)

Let $m = \{Q, \Sigma, \delta, q_0, F\}$ be an FA & has n number of states and let L be the regular language accepted by M. then for every string $w \in L$ there exist a 'n'

45

Such that the length of the $w >= n$,

you can break the w into 3 substrings

$w = xyz$ such that

$|y| > 0$

$|xy| \leq n$

then, for all value i,

$i \geq 0$, $xy^iz \in L$

<u>Proof:</u> Let $M = \{Q, \Sigma, \delta, q_0, F\}$ be the FA given accepting the language L.

Let n be the number of states of given FA

Consider the string w ; $w = \{a_1 a_2 a_3 \dots a_k\}$

where $k \geq n$

for each value $i = 0, 1, 2 \dots n$

define state p_i such that

$$\boxed{\hat{\delta}(p_0, a_1 a_2 a_3 \dots a_i) \rightarrow p_i}$$

where δ is transition function of given automata

and p_0 is the start state of the given automata

p_i is the state the automata M is in after reading first i symbols of w

By the pigeon hole principle it is not possible for the $n+1$ different p_i for $i = 0, 1, 2, \dots, n$ to be distinct.

Since there are only n distinct state

thus we can find 2 diff integers i & j

where i & j value lies b/n 0 to n

$(0 \leq i < j \leq n)$

$p_i = p_j$

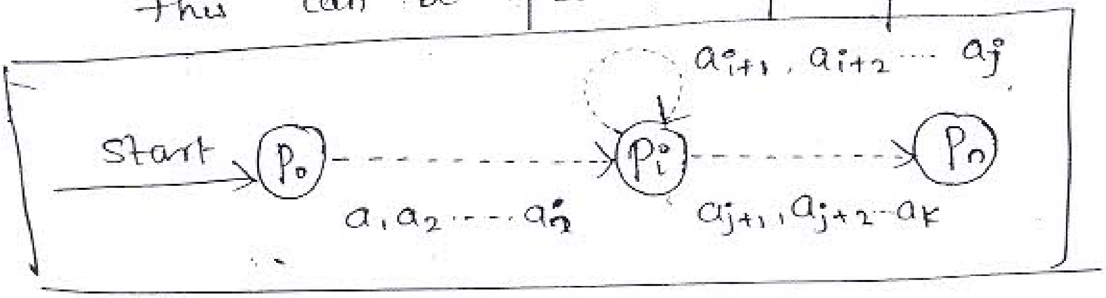
Now we can break ~~str~~ string w into 3 Substrings

$w = \{x, y, z\}$ 1. $x = a_1 a_2 \ldots a_i$

$y = a_{i+1}, a_{i+2} \ldots a_j$

$z = a_{j+1}, a_{j+2} \ldots a_k$

this can be pectorically represented as



i.e the string 'x' takes us to state p_i, and 'y' takes the state from p_i to p_i itself and z being the balance of the w takes to the ~~fini~~ final state p_n

The automata can accept the input xy^iz for any value $i \geq 0$ ie if $i=0$ the automata starts at p_0 on input x goes to p_i and on z it goes to state p_n since 'y' is ϵ when $i>0$, the x part takes us from p_0 to p_i and on y part it circles from p_i to p_i i number of times then on z it goes to state p_n accepting the string thus, for any value $i \geq 0$ $xy^iz \in L$

21/9/16 STEPS TO PROVE LANGUAGE IS NOT REGULAR USING PUMPING LEMMA

S1 : Assume language is regular and n is a number of State of FA accepting the given language

S2: Select a string $w \in L$ such that the length of the w is $\geq n$ and ~~bratte~~ break it into substrings $w = xyz$, $|y| > 0$ & $|xy| \leq n$

S3: Find any value of i such that $xy^iz \notin L$ which contradicts the pumping lemma of R.L that prooves that given L is not R.

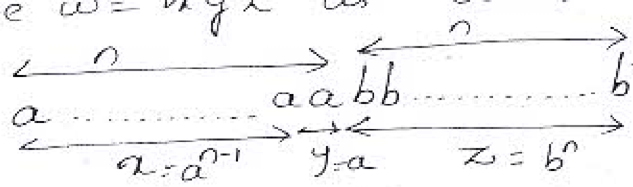
Problem:

1) ST $L = \{a^n b^n \mid n \geq 0\}$ is not regular

Sol: S1 : Assume L is regular, 'n' is the no of states in FA of language L

S2 : Considers a string $w = a^n b^n \in L$, $|w| = 2n \geq n$

S3 : Divide the string w in to three parts i.e $w = xyz$ as below



it satisfies the condition

$|y| = 1 > 0$

$|xy| = n \leq n.$

S4: According to pumping lemma of regular languages $xy^iz \in L$, $i \geq 0$

Let us take $i = 0$

ie xy^0z

$(a^{n-1})\,(a)^0\,(b^n)$

$\Rightarrow a^{n-1} b^n \notin L$

Here number of a is less than the number of b.
This contradicts the pumping lemma of R.L and hence the language is not regular.

Q) PT the language $L = \{ww^R \mid w \in (a+b)^*\}$ is not regular

(3a)

Sol: S1: Assume L is regular 'R' is the number of states in FA of language L.

S2: Consider a string

$w = a^n b^n b^n a^n$

S3: Divide the string w into three parts.

i.e $w = xyz$ as below

$$\overbrace{aa\ldots\ldots aa}^{n}\overbrace{b\ldots\ldots bb}^{n}\overbrace{bb\ldots\ldots b}^{n}\overbrace{aa\ldots\ldots aa}^{n}$$

$$\underbrace{a^{n-1}}_{x}\ \underbrace{a}_{y}\ \underbrace{b^n b^n a^n}_{z}$$

$|y| = 1 > 0$

$|xy| = n \leq n$

According to pumping lemma of regular language

$xy^iz \in L,\ i \geq 0$

Let us take $i=0$

i.e xy^0z

$= a^{n-1}(a)^0\, b^n b^n a^n$

$\Rightarrow a^{n-1} b^n b^n a^n \notin L$

This ~~is~~ contradicts the pumping lemma of RL and hence the language is not regular

3) ST $L = \{a^l b^k \mid l \neq k, l, k >= 0\}$ is not regular.

Sol:- S1: Assume L is regular, 'n' is the no of states of Language.

5,

S2: $w = a^n b^{n-1}$ $|w| = 2n-1 > n$.

S3:

$$\overbrace{aa\ldots\ldots aa}^{n}\overbrace{b\ldots\ldots bb}^{n-1}$$

$$\underbrace{aa\ldots\ldots a}_{x = a^{n-1}}\ \underbrace{a}_{y = a}\ \underbrace{b\ldots\ldots bb}_{z = b^{n-1}}$$

S4: According to pumping lemma

$xy^iz \in L$, $i \geq 0$.

$i = 0$, $(a^{n-1})(a)^0(b^{n-1})$

$\Rightarrow a^{n-1} b^{n-1} \notin L$ because l & k is not equal

it contradicts the ~~Language~~ pumping lemma hence it is not regular

4) ST $\{a^l b^k \mid l > k, (k \geq 0\}$ is not regular

Ass 6

Sol: S1: Assum L is regular & 'n' is the no of State of Language

S2: $w = a^n b^{n-1}$ $|w| = 2n-1 > n$.

S3:

$$\overbrace{aa\ldots\ldots aa}^{n}\overbrace{b\ldots\ldots bb}^{n-1}$$

$$\underbrace{aa\ldots\ldots a}_{x = a^{n-1}}\ \underbrace{a}_{y = a}\ \underbrace{b\ldots\ldots bb}_{z = b^{n-1}}$$

7 **

S4: According to pumping lemma

$xy^iz \in L$, $i \geq 0$.

$i = 0$, $(a^{n-1})(a)^0(b^{n-1})$

$\Rightarrow a^{n-1} b^{n-1} \notin L$ because ~~$l > k$~~ l is not $> k$

it contradicts the pumping lemma
hence it is not regular.

5) ST $L = \{ w \mid n_a(w) < n_b(w),\ w \in (a+b)^* \}$ is not regular.

Sol^n: S1 Assume L is regular n is no of states in FA of language L

S2 : $w = a^{n-1} b^n$ $\quad |w| = 2n-1 > n$

S3 : $\underbrace{aa\ldots\ldots aa}_{x}\ \underbrace{b}_{y}\ \underbrace{b\ldots\ldots bb}_{z}$

$\quad a^{n-1} \qquad b \qquad b^{n-1}$

According to the pumbing ~~too~~ lemma
$xy^iz \in L \quad i \geq 0.$
$i = 0.\ (a^{n-1})(b)^0(b^{n-1})$
$\Rightarrow a^{n-1} b^{n-1} \notin L$ because L & K is not equal in L

it contradicts the pumping lemma
hence it is not regular

Ass.
6) ST the following languages are not regular
i) $L = \{ 0^n 1^{2n} \mid n \geq 0 \}$
ii) $L = \{ 0^n 1^m 2^n \mid n \geq 0 \}$
iii) $L = \{ 0^n 1 0^n \mid n \geq 0 \}$

7) PT. the following language are not regular
***
$L = \{ a^n \mid n \text{ is prime} \}$

Sol^n: 1) Assume L is regular, 'n' no of states in FA.

2) $w = a^n \quad |w| = n$, n is a prime number.

3) Split the string w into 3 part.

$w = a^i a^j a^{n-i-j}$, $|y| = i > 0$
$|xy| = i + j \leq n$.

pump y into i number of times

Let $i=0$ $\quad x(y)^0 z = a^i (a^j)^0 a^{n-i-j}$

$= a^{n-j} \notin L$ for all value $j>0$

It is considered that n is prime number & value of j is greater than zero. Subtraction of j from n doesn't always yields another prime number and hence a^{n-j} doesn't belongs to L for all values of $j>0$

This contradicts the pumping lemma theorem hence language is not regular

8) P.T $L = \{a^n \mid n$ is perfect square$\}$ is not regular

Sol: Assume L is regular, 'n' no of states in FA

2) $w = a^n \in L$ $\quad |w| = n$, n is a ~~prime~~ number perfect square

3) $xy = a^i a^j a^{n-i-j}$

$|y| = i > 0$

$|xy| = i+j \leq n$

4) pump y, into i number of times

Let $i=0$ $\quad x(y)^0 z = a^i (a^j)^0 a^{n-i-j}$

$= a^{n-j} \notin L$ for all value $j>0$

It is given that n is ~~prime~~ square number & value of j is greater than zero. Subtraction of j from n doesn't always yields another ~~prime~~ perfect square number & hence a^{n-j} doesnot belongs to L for all values of $j>0$

This contradicts the pumping lemma theorem

hence language is not regular

9) ST $L = \{a^{n!} \mid n \geq 0\}$ is not regular

36

Sol: 1) L is regular. 'n' no of states in FA

2) $w = a^{n!}$ $\quad |w| = n! \geq n$

3) $w = a^{i} a^{j} a^{n!-i-j}$

$|y| = j > 0$

$|xy| = i+j \leq n$

4) pump y 'i' no of times

Let $i=0$ $\quad x(y)^0 z = a^{i} (a^{j})^{0} a^{n!-i-j}$

$= a^{n!-j} \notin L$ for all values of $j > 0$

Ass.

10) PT following languages are not regular

i) $L = \{0^n 1 0^n \mid n \geq 1\}$

ii) $L = \{0^n 1^m 2^n \mid m, n \geq 1\}$

** iii) $L = \{a^n \mid n$ is a power of $2\}$

** iv) $L = \{a^n b^m \mid n \leq m\}$

i) $L = \{0^n 1 0^n \mid n \geq 1\}$

Sol: 1) Assume L is regular & 'n' no of states in FA

2) $w = 0^n 1 0^n$

3) $\underbrace{0 \ldots\ldots 0}_{\substack{x \\ 0^{n-1}}} \underbrace{0}_{\substack{y \\ 0}} \underbrace{1 0 \ldots\ldots 0 0}_{\substack{z \\ 1 0^n}}$

$|y| = 1 > 0$

$|xy| = n \leq n$

4) According to pumping lemma theorem

$xy^{i}z \in L, \quad i \geq 0$

Let $i = 0 \Rightarrow xy^{0}z = 0^{n-1} (0)^{0} \cdot 1 0^{n}$

$= 0^{n-1} 1 0^{n} \notin L$

ii) $L = \{0^n 1^m 2^n \mid m, n \geq 1\}$

S.D Assume L is regular with 'n' no of states in FA

Consider the string

$w = 0^n 1 2^n$

$$\underbrace{0 \ldots\ldots 0}_{0^{n-1}}\ \underbrace{0}_{\substack{y \\ 0}}\ \underbrace{1 2 \ldots\ldots 2 2}_{\substack{z \\ 12^n}}$$

$|y| = 1 > 0$

$|xy| = n \geq n$

By the pumping lemma theorem

$xy^iz \notin L \Rightarrow i = 0.$

$xy^0z = 0^{n-1}(0)^0(12)^n$

$\Rightarrow 0^{n-1}(12)^n \notin L$

Hence it contradicts the pumping lemma theorem & it is not regular

iii) $L = \{a^n \mid n \text{ is a power of } 2\}$

S.D: $\Rightarrow L = \{a^n \mid n \text{ is a } n^2\}$

Assume L is a regular with 'n' no of states in FA

$w = a^{2n}$

tes

Questions:

UNIT - 2

1) Proove that if $L = L(A)$ for some dfA then there is a regular expression R such that $L = L(R)$. (Kleens) 6M.

2) For the following DFA obtain R.E, $R_{ij}^{(0)}$ & $R_{ij}^{(1)}$

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

3) Construct NFA for regular expression $(01+10)^+$

$\Rightarrow (01+10)(01+10)^*$

4) Define R.E and write the RE for the following languages

i) $L = \{a^{2m} b^{2n} \mid m, n \geq 0\}$

ii) $\Sigma = \{0,1\}$ not containing 00

5) Convert the RE to ε-NFA $(0+1)^* 1 (0+1)$

6) PT for every RE there exist a finite automata which accept the same language accepted by regular expression

is FA

7) Give R.E for the following languages

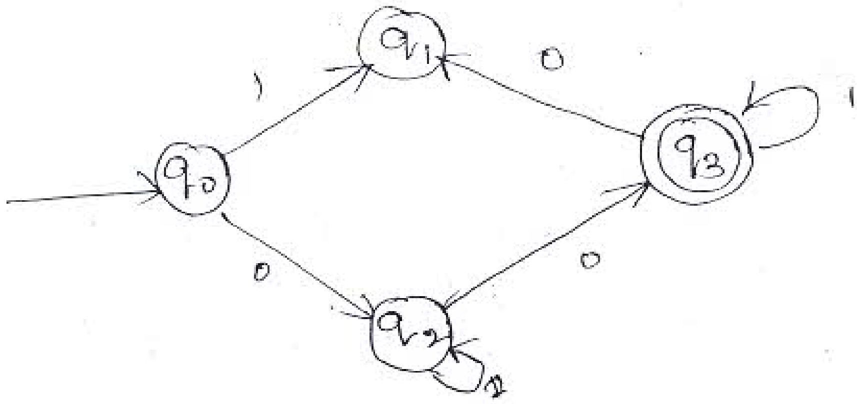
1) $L = \{w \mid w \in (a+b)^* \; |w| \bmod 3 = 0\}$

2) $L = \{w$ is a string even number of 0's followed by odd number of 1's.

$L = \{w \mid w \in (a,b)^*$, has exactly one pair of consequitive a's $\}$

8) Obtain the RE for following FA using state elemination technique

(55)

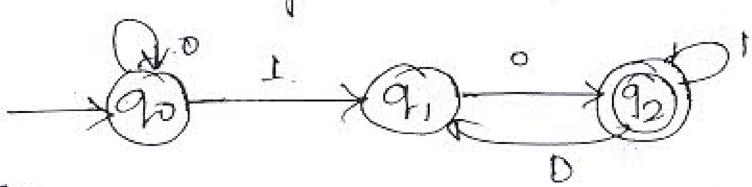


9) $L = \{a^n b^m \mid n \leq 4 \quad m \geq 2\}$

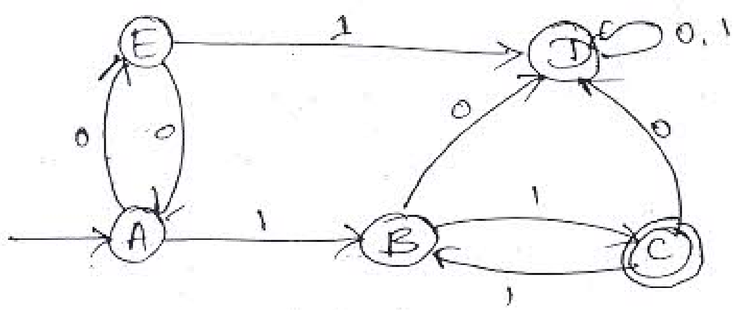
10) Mention the application of R.E

11) Solve using state elimination method

i)

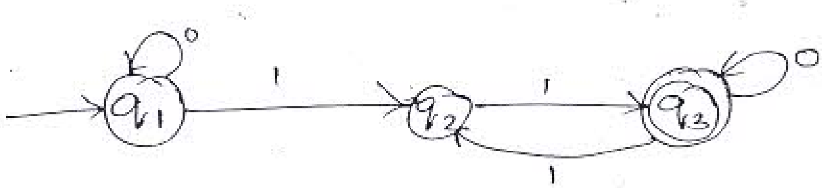


ii)



$\rightarrow (00)^* 1(1+11)^*$

12) Convert the following DFA to RE using Kleen's theorem



13) Construct the NFA for RE $a^* + b^* + c^*$

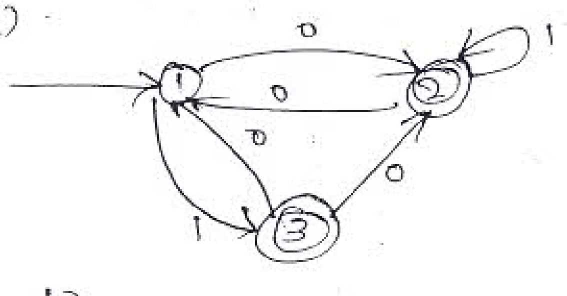
14) Define R.E & RL.

15) Find NFA which accepts $L(r)$

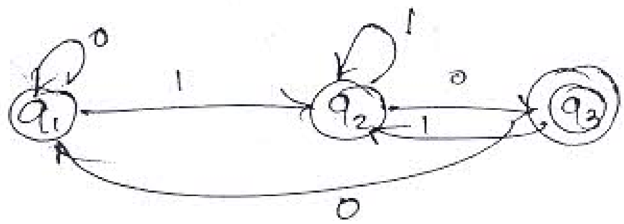
~~L(r)~~ $r = (a+bb)^* (ba^* + \lambda)$ λ is ε

16) Convert the following DFA to RE by state elemination method

a)



b)



UNIT 3

1) State & Proove pumping lemma theorem for R.L.

2) S.T. $L = \{A^{n!} \mid n \geq 0\}$

3) Construct the minimum automata equavalent to given automata. (Or minimize the following automata whose transition table is given below

	0	1
q_0	q_0	q_3
q_1	q_2	q_5
q_2	q_3	q_4.
q_3	q_0	q_5
q_4	q_0	q_6
q_5	q_1	q_1
* q_6	q_1	q_3

4) ST ... is not regular

5) what is homomorphism explain with an ex

6) Consider the transition table given below
   ① draw the table of distinguishability
   ② Construct the equivalent minimize ~~DFA~~ DFA

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

7) Proove that ~~Rh~~ RL are closed under homomorphism

8) Proove that the language $L = \{ww^R \mid w \in (a,b)^*\}$ is not regular

9) Write note on table filling method
   when twostates are said to be equivalent or distinguishable

10) ST that $L = \{w \mid w \in (a,b)^* \; n_a(w) < n_b(w)\}$ is not regular

11) Consider the DFA given below
   i) Draw table of distinguishability
   ii) minimize DFA
   iii) write the L accepted by DFA

a)

	0	1
→ 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

FLAT