REGULAR EXPRESSION :-

The language accepted by finite Automata are easily described in a simple expression is called regular Expression. (Reg. Exp (R.E) → Method of representing language).

Regular Expression over Σ can be defined as

(i) ϕ is a R.E denotes Empty set. i.e., $\phi = \{\}$

(ii) ε is a R.E denotes set $\{\varepsilon\}$ i.e $\varepsilon = \{\varepsilon\}$

(iii) 'a' is a R.E, denotes $\{a\}$

(iv) r and s are R.E and denoting L_1 and L_2 then

$r+s = L_1 \cup L_2$

$rs = L_1 L_2$

$r^* = L_1^*$

Eg

1) R.E for the lang. accepting all combinations of "a" over $\Sigma = \{a\}$

$L = \{\varepsilon, a, aa, aaa, \ldots\}$

$R.E = a^*$

2.) all combinations of "a" except null string

$R.E = a^+$

3.) any combination of a and b, even null string

i.e, $L = \{\varepsilon, a, aa, ab, abb, bab \ldots\}$

$R.E = (a+b)^*$

4) any combination of a and b except null string

R.E = $(a+b)^+$.

5.) Start with 1 and end with 0 over an $\Sigma \{0,1\}$

R.E = $1(0+1)^* 0$

6.) all strings end with 00 over $\Sigma \{0,1\}$

R.E = $(0+1)^* 00$

7.) L = { Language starting and ending with "a" and having any no. of "b's" inbetween }

R.E = $a b^* a$

8.) atleast one "a"

R.E = $a^+ b^+$

9.) all strings which begins or ends with 00 or 11

$L_1 = (00+11)(0+1)^*$

$L_2 = (0+1)^*(00+11)$

10.) Write R.E to denote 3^{rd} character from right end of the string is always "a"

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

3 2 1

11.) Exactly 2b's $\Rightarrow a^* b a^* b a^*$

12.) $\Sigma = \{0\}$ even length of string $\rightarrow$ R.E = $(00)^*$

ie, $L = \{\varepsilon, 00, 0000, 000000, \ldots\}$

13.) odd length of string over $\Sigma = \{1\}$

$L = \{ \varepsilon, 1, 111, 11111, \ldots \}$

$R \cdot E = 1(11)^*$

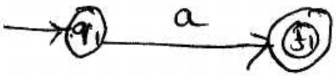
<u>Regular Expression</u> to <u>finite Automata</u> (NFA-ε)

Using Thompson's Construction Method.

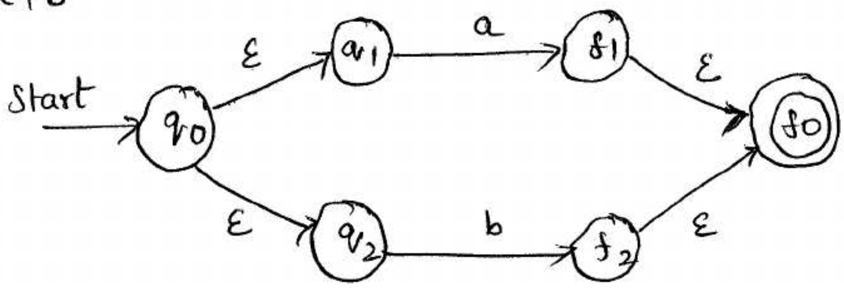
case 1: <u>union</u>

$L_1 \cup L_2 \Rightarrow (a+b)$

$L_1 = \{a\}$ $\quad\quad L_2 = \{b\}$

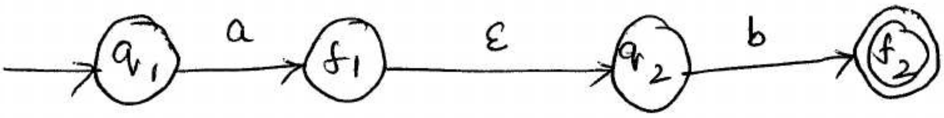


a+b

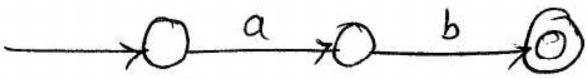


Case 2 :- Concatenation

$L_1 L_2$

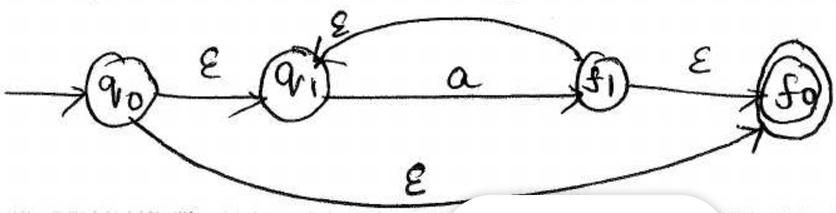


(or)



Case 3 :- closure

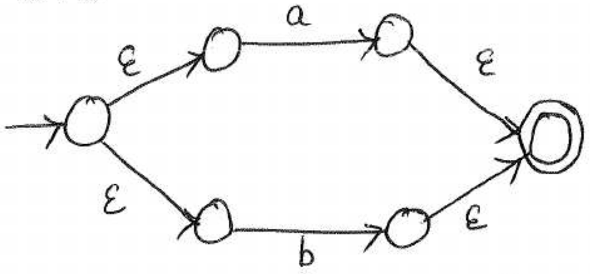
$L_1^* \Rightarrow a^*$ i.e $L = \{ \varepsilon, a, aa, aaa \ldots \}$



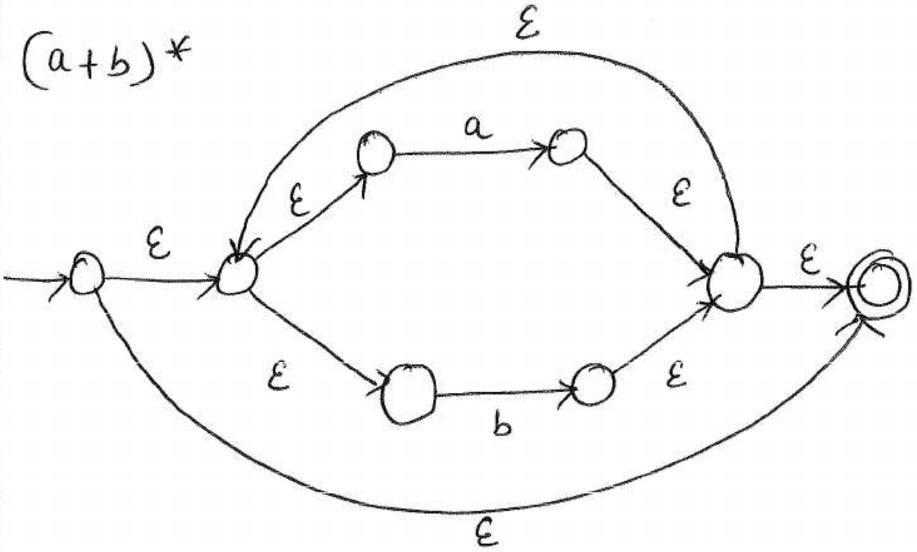
Regular Expression to NFA-ε

Eg-1 $a(a+b)^*a$

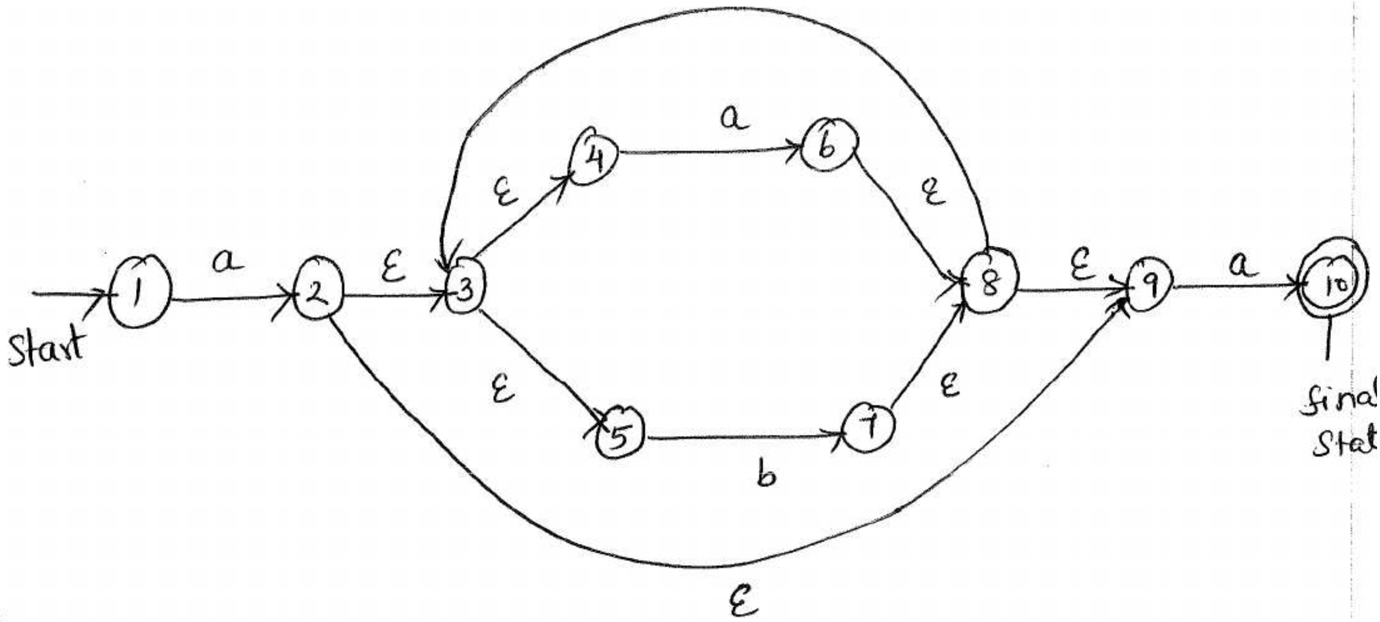
$a+b$



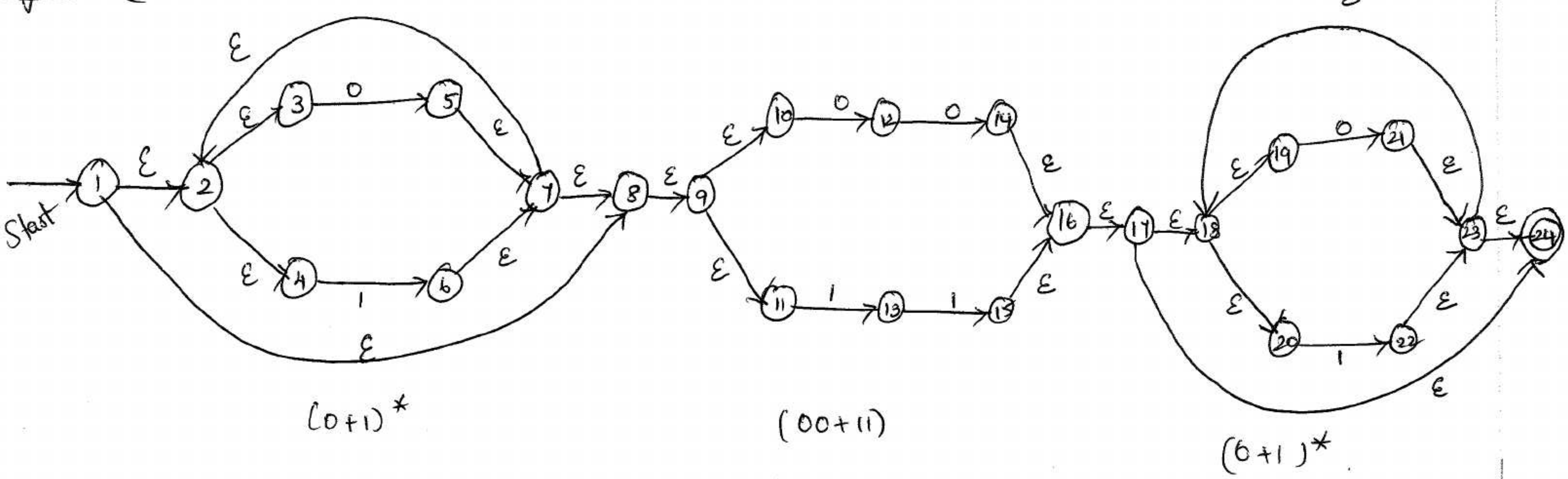
$(a+b)^*$



$a(a+b)^*a$



Eg 2:- $(0+1)^* (00+11) (0+1)^*$



Eg:-

Assignment Qns

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

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

3.) $((10)+(0+1)^*)01$

4.) $((01+001)^* 0^*)^*$

Finite Automaton to Regular Expression.

Arden's Theorem :-

If P and Q are two regular Expression over Σ, if P doesn't contain ε, then the following eqn R is given by $R = Q + RP$ has unique solution $R = QP^*$.

Proof :- Let, P and Q are two regular expressions over the input string Σ.

If P doesnt contain ε then there exists R such that

$R = Q + RP \rightarrow (1)$

Replace $R = QP^*$ in eq (1)

Consider R.H.S of the eqn (1)

$$R = Q + RP$$
$$= Q + QP^* P$$
$$= Q(\underline{\varepsilon + P^* P}) \quad \because \varepsilon + R^* R = R^*$$
$$R = QP^*$$

Thus $R = QP^*$ is proved.

To prove $R = QP^*$ is a <u>unique</u> soln, we will replace L.H.S of eq (1) by $Q + RP$ then

$$R = Q + RP$$
$$= Q + [Q + RP]P$$
$$= Q + QP + RP^2$$
$$= Q + QP + (Q + RP)P^2$$
$$= Q + QP + QP^2 + RP^3$$

Thus if we go on replacing R by $Q + RP$ then we get

$= Q + QP + QP^2 + \cdots QP^n + \underline{R}P^{n+1}$

$= Q + QP + QP^2 + \cdots QP^n + QP^*P^{n+1}$ $\boxed{\because R = QP^*}$

$= Q\underbrace{[\varepsilon + P + P^2 + \cdots P^n + P^*P^{n+1}]}$

$= QP^*$

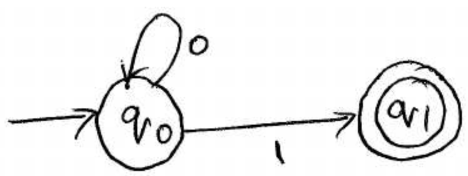
Thus $R = Q + RP$ has a unique soln $R = QP^*$.

Finite Automata to Regular Expression using

Arden's Method :-

1. Finite Automata should not have ε-moves.
2. F.A should have only one start state say q_i
3. Express states interms of transitions (ie transition required to reach q_0 from other states).

Eg 1



Let us write Eqns.

$q_0 = q_0 \cdot 0 + \varepsilon \rightarrow (1)$

(X) since q_0 is a start state ε will be added, & input 0 is coming to q_0 from q_0.

Hence we write

State = Source state of i/p X i/p coming to it

III^ly

$q_1 \Rightarrow q_0 \cdot 1 \rightarrow ②$

Apply Arden's Rule to ①

$$q_0 = q_0 \cdot 0 + \varepsilon$$
$$R \quad\; R \;\; P + Q$$

$$q_0 = \varepsilon 0^*$$

$\because R = RP + Q$

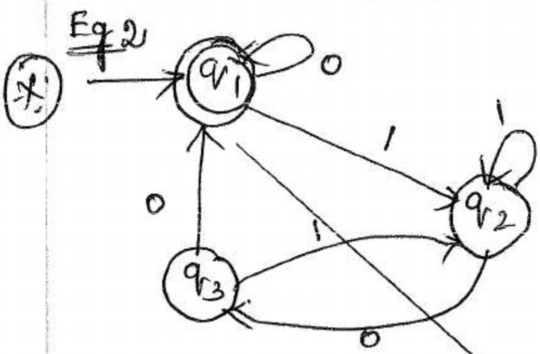
$R = QP^*$

$$\boxed{q_0 = 0^*} \rightarrow ③$$

Sub ③ in ②

$$\boxed{q_1 = 0^* 1}$$

Eg 2



Let us write Eqns

$q_1 = q_1 \cdot 0 + q_3 \cdot 0 + \varepsilon \rightarrow ①$

$q_2 = q_1 \cdot 1 + q_2 \cdot 1 + q_3 \cdot 1 \rightarrow ②$

$q_3 = q_2 \cdot 0 \rightarrow ③$

Sub (3) in (2)

$q_2 = q_1 \cdot 1 + q_2 \cdot 1 + q_2 \cdot 0 \cdot 1$

$\underset{R}{q_2} = \underset{Q}{q_1 \cdot 1} + \underset{R}{q_2} \underset{P}{(1+01)}$

Apply Arden's Rule $R = Q + RP$

$q_2 = q_1 \cdot 1(1+01)^* \rightarrow ④$

Sub(4) in(3)

$q_3 = q_1 \cdot 1(1+01)^* \cdot 0 \rightarrow ⑤$

Sub (5) in (1)

$q_1 = q_1 \cdot 0 + q_1 \cdot 1(1+01)^* \cdot 0 \cdot 0 + \varepsilon$

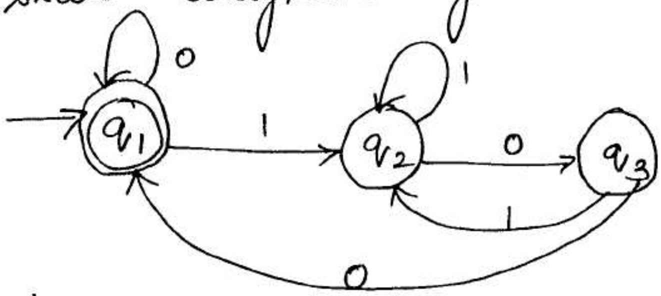
$\underset{R}{q_1} = \underset{R}{q_1} \underbrace{(0 + 1(1+01)^* 00)}_{P} \underset{+Q}{+ \varepsilon}$

Apply Arden's Rule

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

$$\underline{q_1 = (0 + 1(1+01)^* 00)^*}$$

Eg:3 Construct the regular expression corresponding to the state diagram given



<u>using Arden's theorem</u>

$q_1 = q_1 0 + q_3 0 + \varepsilon$ being start state, add ε.

$q_2 = q_1 1 + q_2 1 + q_3 1$

$q_3 = q_2 0$

Considering q_2

$q_2 = q_1 1 + q_2 1 + q_3 1$

$q_2 = q_1 1 + q_2 1 + q_2 01 \quad \because q_3 = q_2 0$

$q_2 = q_2(1+01) + q_1 1$

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

$q_2 = q_1 1 (1+01)^*$

Put value of q_2 in q_3 we get

$q_3 = q_1 1 (1+01)^* 0$

now sub value of q_3 in q_1

$q_1 = q_1 0 + q_3 0 + \varepsilon$

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

$q_1 = q_1 [0 + 1(1+01)^* 00] + \varepsilon$

As $R = Q + RP$ gives $R = QP^*$.

$R = q_1$, $Q = \varepsilon$, $P = 0 + 1(1+01)^* 00$ We get

$q_1 = \varepsilon[0 + 1(1+01)^* 00]^*$

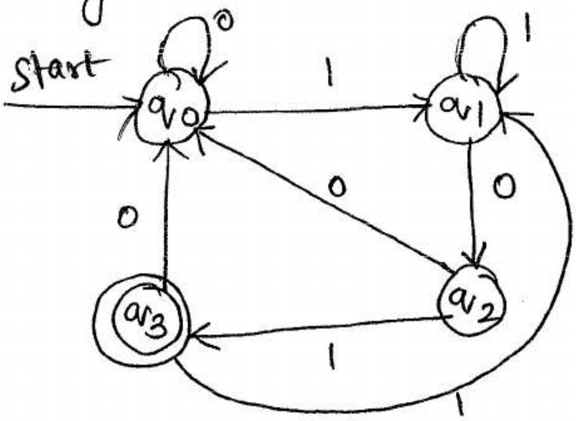
$q_1 = (0 + 1(1+01)^* 00)^*$ $\quad \because \varepsilon R = R$.

State q_1 is a final state, Hence an eqn for final state becomes a regular expression for given state diagram.

$r.e = (0 + 1(1+01)^* 00)^*$.

Eg 4:

Consider the FA and Construct Regex that is accepted by it.



Assume q_0 as initial state

$q_0 = q_0 0 + q_2 0 + q_3 0 + \varepsilon \quad \rightarrow (1)$

$q_1 = q_0 1 + q_1 1 + q_3 1 \quad \rightarrow (2)$

$q_2 = q_1 0 \quad \longrightarrow (3)$

$q_3 = q_2 1 \quad \longrightarrow (4)$

put value of q_2 in eq (4)

$$q_3 = q_1 01 \longrightarrow (5)$$

Put value of eq (5) in (2)

$$q_1 = q_0 1 + q_1 1 + (q_1 01)1$$

$$q_1 = q_0 1 + q_1 1 + q_1 011$$

$$\underbrace{q_1}_{R} = \underbrace{q_1}_{R} \underbrace{(1+011)}_{P} + \underbrace{q_0 1}_{Q}$$

$$\because R = Q + RP$$

$$R = QP^*$$

$$q_1 = q_0 1 (1+011)^*$$

now solving q_0

$$q_0 = q_0 0 + q_2 0 + q_3 0 + \varepsilon$$

$$= q_0 0 + (q_1 0)0 + (q_1 01)0 + \varepsilon$$

$$q_0 = q_0 0 + q_1 (00 + 010) + \varepsilon$$

$$= q_0 0 + q_0 1 (1+011)^* (00+010) + \varepsilon.$$

$$= q_0 (0 + 1(1+011)^* (00+010)) + \varepsilon$$

$$= q_0 = \varepsilon (0 + 1(1+011)^* (00+010))^*$$

Now solve

$$q_3 = q_1 0 1$$

$$= q_0 1 (1+011)^* 01$$

$$q_3 = (0 + 1(1+011)^* (00+010))^* 1 (1+011)^* 01$$

As q_3 is final state, We get the r.e as equation for q_3.

$$r.e = (0 + 1\,(1+011)^*\,(00+010))^*\,1\,(1+011)^*\,01)$$

(4) Prove that

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

$$0^*1\,(0+10^*1)^*.$$

Let us solve L.H.S first.

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

take $(1+00^*1)$ as a common factor

$$1+00^*1\;\underbrace{(\varepsilon + (0+10^*1)^*\,(0+10^*1))}_{(\varepsilon + R^*R)\text{ where } R = (0+10^*1)}$$

We know $(\varepsilon + R^*R) = (\varepsilon + RR^*) = R^*$.

$\therefore\; 1+00^*1\,(0+10^*1)^*$

taking 1 as common factor

$$(\varepsilon + 00^*)\,1\,(0+10^*1)^*$$

Applying $\varepsilon + 00^* = 0^*$

$$0^*1\,(0+10^*1)^*.$$

$= R.H.S.$

Hence the two regular Expressions are equivalent.

Identity Rules :-

Two Regular Expressions P and Q are equivalent (denoted as $P = Q$) if & only if P represents the same set of strings as Q does. For showing this equivalence of regular expressions, need to show some identities of regular Expressions

Let P, Q, and R are regular expressions then the identity rules are given below.

1.) $\varepsilon R = R\varepsilon = R$

2.) $\varepsilon^* = \varepsilon$

3.) $\phi^* = \varepsilon$

4.) $\phi R = R\phi = \phi$

5.) $\phi + R = R$

6.) $R + R = R$

7.) $RR^* = R^* R = R^*$

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

9.) $\varepsilon + RR^* = \varepsilon + R^* R = R^*$

10.) $(P+Q)R = PR + QR$

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

12.) $\varepsilon + R^* = R^*$

13.) $(PQ)^* P = P(QP)^*$

14.) $R^* R + R = R^* R$

15.) $R + RS^* = RS^*$

Conversion of finite Automata to Regular Expression:-

Theorem : The regular expression can also be represented by its equivalent deterministic finite Automata.

Proof: Let, L be the set of the language accepted by the DFA. The DFA can be denoted by the

$M = (\{q_1, q_2, \cdots q_n\}, \Sigma, \delta, q_1, F)$

Let

R_{ij}^{K} denotes the set of the strings x such that $\delta(q_i, x) = q_j$. The q_i and q_j indicates source state to target state respectively. The inputs are going through the states of finite automata means that with some input entering into the states and coming out of it. The value of K is always less than i or j.

The r_{ij}^{k} is denoted by,

$$R_{ij}^{K} = R_{ij}^{K-1} \cup R_{iK}^{K-1} \left(R_{KK}^{(K-1)}\right)^{*} R_{Kj}^{(K-1)}$$

$$R_{ij}^{0} = \{\alpha \mid \delta(q_i, \alpha) = q_j\} \quad \text{if } i \neq j$$

$$\{\alpha \mid \delta(q_i, \alpha) = q_j\} \vee \{\varepsilon\} \quad i = j$$

Show that for each i, j, k there exists a regular expression R_{ij}^{k} denoting the language R_{ij}^{K}. We will put the induction on K basis (K=0)

The R_{ij}^{0} is a set of strings each of which is either ε or a single symbol.

The R_{ij}^{0} is based on $\delta(q_i, a) = q_j$. The R_{ij}^{0} denotes the set of such a's, if there is no such a then it will be taken as ϕ. If $i = j$ the all a's $+ \varepsilon$. will be the set.

<u>Induction</u> :- The formula for getting the language R_{ij}^{k} is given by regular Expression

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

Which completes the induction

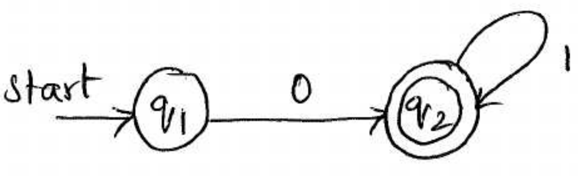
To get the final regular Expression, We have to simply get the language R_{ij}^{n}, where i indicates start state and j indicates final state and n will be Number of states, if there are P number of paths leading to final state

$$R_{ij}^{n} = R_{ij_1}^{n} + R_{ij_2}^{n} + \cdots R_{ij_P}^{n}$$

$i \rightarrow$ start state.
$n \rightarrow$ no. of states.

$$= \bigcup_{j \in F} R_{ij}^{n} \qquad \because j_1, j_2 \ldots j_P \in \text{Final states.}$$

<u>Eg-1</u> Construct regular Expression from given finite automata.



① R_{ij} values indicate the set of all the i/p string from q_i to q_j

② if $i = j$ then add ε with i/p string.

③ if $i \neq j$, there is no path from i to j then add ϕ.

Soln :- $R^{K}_{ij} = R^{K-1}_{ij} + R^{K-1}_{ij} \left(R^{(K-1)}_{KK}\right)^{*} R^{K-1}_{Kj}$

When $K = 0$

$R^{0}_{11} = \varepsilon$ ($i = j$, no i/p from q_1 to q_1, so add ε)

$R^{0}_{12} = 0$ ($i \neq j$, i/p from q_1 to q_2 is 0)

$R^{0}_{21} = \phi$ ($i \neq j$, no i/p from q_2 to q_1, so add ϕ)

$R^{0}_{22} = 1 + \varepsilon$ ($i = j$, i/p from q_2 to q_2, so add ε to i/p

When $K = 1$, $i = 1$, $j = 1$

$$R^{1}_{11} = R^{0}_{11} + R^{0}_{11} \left(R^{0}_{11}\right)^{*} R^{0}_{11}$$
$$= \varepsilon + \varepsilon\, \varepsilon^{*}\, \varepsilon \Rightarrow \varepsilon .$$

$K = 1$, $i = 1$, $j = 2$

$$R^{1}_{12} = R^{0}_{12} + R^{0}_{11} \left(R^{0}_{11}\right)^{*} R^{0}_{12}$$
$$= 0 + \varepsilon \cdot \varepsilon^{*} \cdot 0 \qquad \because \varepsilon R = R \varepsilon = R$$
$$R^{1}_{12} = 0$$

$i = 2$, $j = 1$, $K = 1$

$$R^{1}_{21} = R^{0}_{21} + R^{0}_{21} \left(R^{0}_{11}\right)^{*} R^{0}_{11}$$
$$= \phi + \phi \cdot \varepsilon^{*} \cdot \varepsilon \qquad \because \phi R = \phi$$
$$= \phi$$

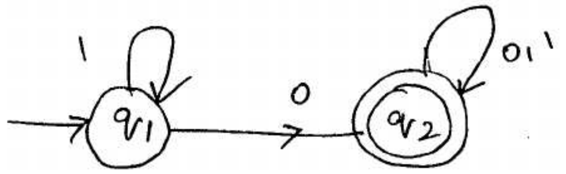
$$R^{1}_{22} = R^{0}_{22} + R^{0}_{21} \left(R^{0}_{11}\right)^{*} R^{0}_{12}$$
$$= (1+\varepsilon) + \phi\,(\varepsilon)^{*} \cdot 0 \qquad \because \phi + R = R .$$
$$= (1+\varepsilon) + \phi$$
$$= (1+\varepsilon)$$

$\therefore$ Calculating R.E, compute for the path from start state to final ie q_1 to q_2

$$R^{2}_{12} = R^{1}_{12} + R^{1}_{12} \left(R^{1}_{22}\right)^{*} R^{1}_{22}$$
$$= 0 + 0 \cdot (1+\varepsilon)^{*} (1+\varepsilon)$$

R^{2} … $+ 01^{*}$

Eg 2/ Obtain the regular expression that denotes the language accepted by following DFA.



By R_{ij} Method,

$$R_{ij}^{(K)} = R_{ij}^{(K-1)} + R_{iK}^{(K-1)} \left(R_{KK}^{(K-1)}\right)^{*} R_{Kj}^{(K-1)}$$

$K=0$

$$R_{11}^{(0)} = \varepsilon + 1$$

$$R_{12}^{(0)} = 0$$

$$R_{21}^{(0)} = \phi$$

$$R_{22}^{(0)} = \varepsilon + 0 + 1.$$

$K=1$

$$R_{ij}^{(K)} = R_{ij}^{(K-1)} + R_{iK}^{(K-1)} \left(R_{KK}^{(K-1)}\right)^{*} R_{Kj}^{(K-1)}$$

$$R_{ij}^{(1)} = R_{ij}^{(0)} + R_{iK}^{(0)} \left(R_{KK}^{(0)}\right)^{*} R_{Kj}^{(0)} \longrightarrow (1)$$

$i=1\ ;\ j=1$

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

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

$$= \varepsilon+1 + \varepsilon+1 \ 1^{*} \ \varepsilon+1 \qquad \because (\varepsilon+R)^{*} = R^{*}.$$

$$= \varepsilon+1 + 1^{*} \varepsilon+1 \ \varepsilon+1 \qquad \because R^{*}(\varepsilon+R) = R^{*}$$

$= \varepsilon+1 + 1^* \varepsilon+1.$

$= \varepsilon+1+1^*$

$= 1^*.$

$$\boxed{R_{11}^{(1)} = 1^*.}$$

$i=2\,;\ j=2$ in (1.)

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

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

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

$$= 0 + 1^* 0 \qquad R + R s^* = R s^*.$$

$$R_{12}^{(1)} = 1^* 0.$$

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

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

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

$$R_{21}^{(1)} = \phi$$

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

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

$$= \varepsilon+0+1+\phi$$

$$R_{22}^{(1)} = \varepsilon+0+1.$$

Now Let us compute for final state which denotes the regular Expression R_{12}^{2} will Computed, because there are total 2 states and final state is q_2 whose start state is q_1

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

$$= 1^*0 + 1^*0\ (\varepsilon+0+1)^* (\varepsilon+0+1) \qquad \because (\varepsilon+R)^* = R^*$$

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

$$= 1^*0 + 1^*0\ (0+1)^* \qquad \because R + RS^* = RS^*$$

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

---

* Show that ϕ^* is ε by Constructing NFA using Thomson's Construction.

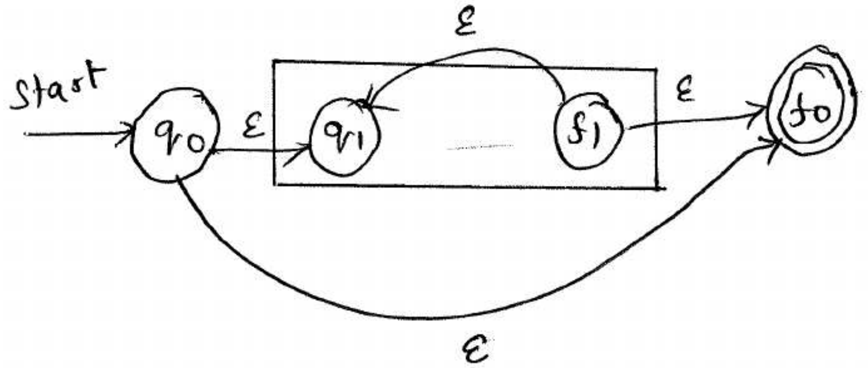
ϕ is represented by q_1 to f_1. We can reach to a final state f_0 from q_0 with i/p ε.

Hence $\phi^* = \varepsilon$

$$\phi^*$$

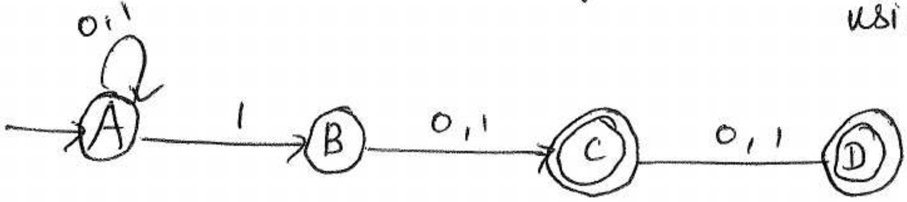
$$\underbrace{\phi^0}_{\varepsilon} \cup \underbrace{\phi^1 \cup \phi^2}_{\text{Empty}}$$

So $\phi^* = \varepsilon$.

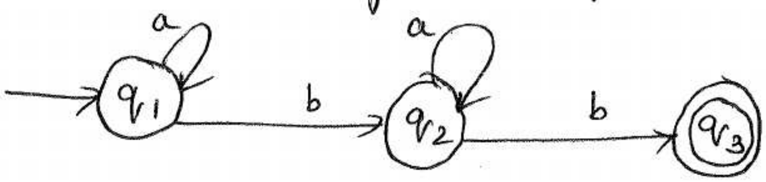


Ans

1.) Convert the following NFA into Regular Expression using Arden's Method.

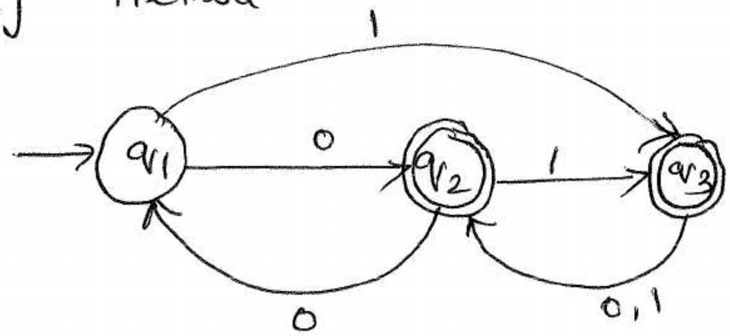


2.) Obtain the Regular Expression for the finite Automata



3.) Prove that $\varepsilon + 1^*(011)^*(1^*(011)^*)^* = (1+011)^*$

4.) obtain the Regular Expression that denotes the language accepted by following Automata. using R_{ij}^{K} Method

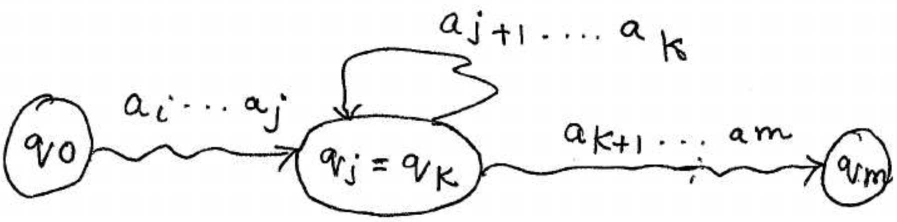


Pumping Lemma for Regular Sets :-

<u>Theorem</u> :- Let L be a regular set. Then there is a constant n such that if z is any word in L and $|z| \geq n$ we can write $z = uvw$ such that $|uv| \leq n$, $|v| \geq 1$ for all $i \geq 0$, uv^iw is in L.

The n should not be greater than the number of states.

<u>Proof</u> :- If the Language L is regular it is accepted by a DFA. $M = (Q, \Sigma, \delta, q_0, F)$ with some particular number of states say, n. Consider the input can be $a_1, a_2, a_3 \ldots a_m$, $m \geq n$. The mapping function δ could be written as $\delta(q_0, q_1, q_2, \ldots q_i) = q_i$



Pumping Lemma.

If q_m is in F i.e, $q_1, q_2, q_3 \ldots q_m$ is in $L(M)$ then $a_1, a_2, \ldots a_j\ a_{k+1}\ a_{k+2} \ldots a_m$ is also in $L(M)$. Since there is path from q_0 to q_m that goes through q_j but not around the loop labelled $a_{j+1} \ldots a_k$.

Thus

$$\delta(q_0, a_1, a_j\, a_{k+1} \cdots a_m) = \delta(\delta(q_0, q_1 \cdots q_j), a_{k+1} \cdots a_m)$$

$$= \delta(q_j, q_{k+1} \cdots q_m)$$

$$= \delta(q_k, q_{k+1} \cdots q_m)$$

$$= q_m.$$

That what we have proved is that given any long string can be accepted by FA. We should be able to find a substring near the beginning of the string that may be pumped ie repeated as many times as we like and resulting string may be accepted by FA.

Advantages :-

The pumping lemma is used to check whether given language is regular or not.

—.

3 0

① Show that $L = \{0^k 1^k \mid k \geq 1\}$ is not regular.

Assume that L is regular,

$$\boxed{\begin{array}{l} Z = UVW \\ |UV| \leq n \\ |V| \geq 1 \\ Z = UV^iW \in L \end{array}}$$ for $i = 1, 2, 3, \ldots$

$Z = 0^k 1^k \quad k \geq 1$

$L = \{01, 0011, \underline{000111}, \ldots\}$

$UV = 0^m \quad m \leq k$

$Z = \underline{UV}\,\underline{W}$ with $UV = 0^3$, $W = 0^3$

$V = 0^j \quad j < m, \; j \geq 1$

$V = 0^1$

$W = 0^{k-m} 1^k$.

$Z = UV^iW$

$\because V^i \Rightarrow V \cdot V^{i-1}$

$= UV\,V^{i-1}\,W$

$= \underline{0^m}\; 0^{j(i-1)}\; 0^{k-m}\; 1^k$

$= 0^{\cancel{m} + j(i-1) + k \cancel{-m}}\; 1^k$

$$UV^iW \Rightarrow 0^{j(i-1)+k} 1^k$$

$i = 1 \Rightarrow 0^{j(1-1)+k} 1^k$

$\Rightarrow 0^k 1^k \in L$.

$i = 2 \Rightarrow 0^{j(2-1)+k} 1^k$

$\Rightarrow 0^{j+k} 1^k \notin L$.

Since given Lang. should contain equal No. of 0's & 1's. Hence given Lang is not regular.

2.) Show that $L = \{ a^p \mid p \text{ is a prime} \}$ is not regular.

Assume L is regular

$$\boxed{\begin{array}{l} Z = UVW \\ |UV| \leq n \\ |V| \geq 1 \\ Z = UV^iW \in L \quad \text{for } i = 1, 2 \dots \end{array}}$$

$Z = UVW$

$L = \{ a^2, a^3, \underline{\underline{a^5}}, a^7, a^{11} \dots \}$

$UV = a^m \quad m < P$

$V = a^j \quad j < m, \; j \geq 1$

$W = a^{P-m}$

$Z = a^5 \quad \because P = 5$

$\underrightarrow{a^2} \cdot \underrightarrow{a^3}$ (UV, W)

$m = 2$

$j = 1$

$Z = UV V^{i-1} W$

$= a^m \; a^{j(i-1)} \; a^{P-m}$

$= a^{\cancel{m} + j(i-1) + P - \cancel{m}}$

$= a^{j(i-1)+P}$

$i = 1$

$\Rightarrow a^{j(1-1)+P}$

$\Rightarrow a^P \in L$

$i = 2$

$\Rightarrow a^{j(2-1)+P}$

$\Rightarrow a^{j+P} \notin L$ is not Regular.

3 0

3.) Prove that $\{a^{p^2} \mid p \geq 1\}$ is not Regular language.

Assume that L is regular,

$$\boxed{\begin{array}{l} Z = UVW \\ |UV| \leq n \\ |V| \geq 1 \\ Z = UV^iW \in L \quad \text{for } i = 1, 2, 3 \ldots \end{array}}$$

Let $a^{p^2} \Rightarrow a^r$

where $r = p^2$

$L = \{a^2, \underline{a^4}, a^9, a^{16} \ldots\}$

$a^r = a^4$

$= \underbrace{a^1 a^1}_{UV} a^2$

$UV = a^m \quad m < r$

$V = a^q \quad q < m, \quad q \geq 1$

$W = a^{r-m}$

$UV^iW = a^m \; a^{q(i-1)} \; a^{r-m}$

$= a^{\cancel{m} + q(i-1) + r - \cancel{m}}$.

$$\boxed{UV^iW = a^{r + q(i-1)}}$$

$i = 1 \Rightarrow a^{r + q(1-1)}$

$\Rightarrow a^r \in L$ ie $r = p^2$

$\Rightarrow a^{p^2} \in L$.

$i = 2 \Rightarrow a^{r+q} \notin L$

ie $a^{p^2+q} \notin L$

Hence proved given Language is not Regular.

4) $\{a^m b^n c^{m+n} \mid m \geq 1, n \geq 1\}$ is not Regular.

$a^m b^n c^\theta$ $\quad \boxed{\theta = m+n}$

$L = \{abcc, \underline{aabbcccc}, \ldots$

$UV = a^p \quad p \leq m$

$Z = \underbrace{a^2}_{UV} / \underbrace{b^2 c^4}_{W}$

$V = a^q \quad q < p, \; q \geq 1$

$m=2$
$p=2$
$q=1$

$W = a^{m-p} b^n c^\theta$

$UV^iW \Rightarrow \underrightarrow{UV}\, V^{i-1} W$

$\Rightarrow a^p \, a^{q(i-1)} \, a^{m-p} b^n c^\theta$

$\Rightarrow a^{\cancel{p} + q(i-1) + m - \cancel{p}} b^n c^\theta$

$$\boxed{UV^iW \Rightarrow a^{q(i-1)+m} b^n c^\theta}$$

When $i=1$ $\quad a^{q(0)+m} b^n c^\theta$

ie $\quad a^m b^n c^\theta \in L$.

$i=2$

$a^{q+m} b^n c^\theta \notin L$

Hence proved Given Language is not Regular.

Ans

1) Show that $L = \{ b^{i^2} \mid i \geq 1 \}$ is not regular.

2.) Show that $L = \{ 0^{2n} \mid n \geq 1 \}$ is not regular

3.) Prove that $L = \{ W \in \{a, b\}^* \mid W = W^R \}$ is not regular.

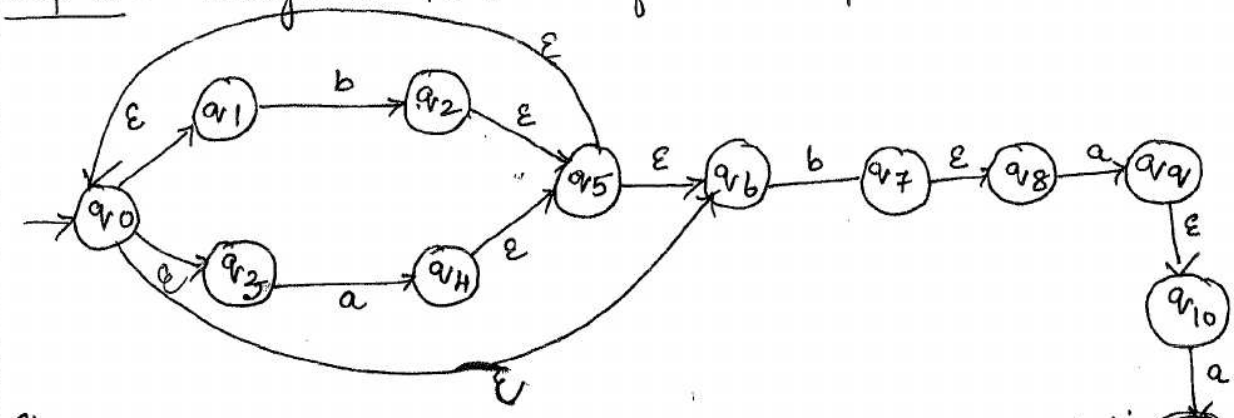
4.) Prove that $L = \{ 1^K \mid K = n^2, n \geq 1 \}$ is not regular.

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

6.) State and prove pumping lemma for regular language.

(iii) Construct the minimal DFA for the regular Expression $(b|a)^* baa$

Step 1 :- design a NFA - ε for the expression $(b|a)^* baa$



Step 2 :- DFA can be obtained by ε-closure computati-on

ε closure $(q_0) = \{q_0, q_1, q_3, q_6\} \rightarrow A$

ε-closure $(q_1) = \{q_1\}$

ε-closure $(q_2) = \{q_0, q_1, q_2, q_3, q_5, q_6\}$

ε-closure $(q_3) = \{q_3\}$

Let, ε-closure $(q_0) = \{q_0, q_1, q_3, q_6\}, a\}$ as state A

Step 3 :- Obtain the input transitions on state A.

$\therefore \delta'(A, a) =$ ε-closure $\{\delta(q_0, q_1, q_3, q_6\}, a\}$

$=$ ε-closure $\{\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_3, a) \cup \delta(q_6, a)\}$

$=$ ε-closure $\{\phi \cup \phi \cup q_4 \cup \phi\}$

$=$ ε-closure $\{q_4\}$

$= \{q_0, q_1, q_3, q_4, q_5, q_6\}$ call it as state B.

$\boxed{\delta'(A, a) = B}$

$\delta'(A, b)$ = ε-closure $\{\delta(q_0, q_1, q_3, q_6), b\}$

= ε-closure $\{\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_3, b) \cup \delta(q_6, b)\}$

= ε-closure $\{\phi \cup q_2 \cup \phi \cup q_7\}$

= ε-closure $\{q_2\} \cup \varepsilon$-closure $\{q_7\}$

= $\{q_0, q_1, q_2, q_3, q_5, q_6\} \cup \{q_7, q_8\}$

= $\{q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_8\}$

= Call it as state C

$\boxed{\therefore\ \delta(A, b) = C}$

$\delta'(B, a)$ = ε-closure $\{\delta(q_0, q_1, q_3, q_4, q_5, q_6), a\}$

= ε-closure $\{\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_3, a) \cup \delta(q_4, a) \cup \delta(q_5, a) \cup \delta(q_6, a)\}$

= ε-closure $\{\phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi\}$

= ε-closure (q_4)

= state B.

$\boxed{\delta'(B, a) = B}$

$\delta'(B, b)$ = ε-closure $\{\delta(q_0, q_1, q_3, q_4, q_5, q_6), b\}$

= εclosure $\{\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_3, b) \cup \delta(q_4, b) \cup \delta(q_5, b) \cup \delta(q_6, b)\}$

= ε closure $\{\phi \cup q_2 \cup \phi \cup \phi \cup \phi \cup q_7\}$

= ε-closure $\{q_2\} \cup \varepsilon$-closure $\{q_7\}$ = C state.

$$\boxed{\therefore \delta'(B,b) = C}$$

$\delta'(C,a) = \varepsilon\text{-closure}\{\delta(q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_8), a)$

$= \varepsilon\text{-closure}\{\delta(q_0,a) \cup \delta(q_1,a) \cup \delta(q_2,a) \cup \delta(q_3,a)$

$\cup \delta(q_5,a) \cup \delta(q_6,a) \cup \delta(q_7,a) \cup \delta(q_8,a)\}$

$= \varepsilon\text{-closure}\{\phi \cup \phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi\ q_9\}$

$= \varepsilon\text{-closure}\{q_4\} \cup \varepsilon\text{-closure}\{q_9\}$

$= \{q_0, q_1, q_3, q_4, q_5, q_6\} \cup \{q_9, q_{10}\}$

$= \{q_0, q_1, q_3, q_4, q_5, q_6, q_9, q_{10}\}$

= Call it as "D" state

$$\boxed{\delta'(C,a) = D}$$

$\delta'(C,b) = \varepsilon\text{-closure}(\delta(q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_8), b)$

$= \varepsilon\text{-closure}\{\delta(q_0,b) \cup \delta(q_1,b) \cup \delta(q_2,b) \cup \delta(q_3,b),$

$\cup \delta(q_5,b) \cup \delta(q_6,b) \cup \delta(q_7,b) \cup \delta(q_8,b)\}$

$= \varepsilon\text{-closure}\{q_2\} \cup \varepsilon\text{-closure}\{q_7\}$

$= C$

$$\boxed{\therefore \delta'(C,b) = C}$$

$\delta'(D,a) = \varepsilon\text{-closure}(\{q_0, q_1, q_3, q_4, q_5, q_6, q_7, q_{10}\}, a\}$

$= \varepsilon\text{-closure}\{\delta(q_0,a) \cup \delta(q_1,a) \cup \delta(q_3,a) \cup \delta(q_4,a) \cup$

$\delta(q_5,a) \cup \delta(q_6,a) \cup \delta(q_9,a) \cup \delta(q_{10},a)\}$

ε- closure $\{\phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi \cup \phi \cup q_{11}\}$

ε- closure$\{q_4\} \cup$ ε- closure$\{q_{11}\}$

$= \{q_0, q_1, q_3, q_4, q_5, q_6, q_{11}\}$

= call its as <u>state E</u>

$\boxed{\delta'(D, a) = E}$

$\delta'(D, b) = \varepsilon$ closure $\{\delta(q_0, q_1, q_3, q_4, q_5, q_6, q_9, q_{10}), b\}$

ε- closure $\{\delta(q_0, b), \delta(q_1, b) \cup \delta(q_3, b) \cup \delta(q_4, b) \cup \delta(q_5, b)$
$\cup\, \delta(q_6, b) \cup \delta(q_9, b) \cup \delta(q_{10}, b)\}$

$= \varepsilon$- closure $\{\phi \cup q_2 \cup \phi \cup \phi \cup \phi \cup q_7 \cup \phi \cup \phi\}$

$= \varepsilon$-closure $\{q_2\} \cup$ ε-closure $\{q_7\}$

= C state .

$\boxed{\delta'(D, b) = C}$

$\delta'(E, a) = \varepsilon$-closure $\{\delta(q_0, q_1, q_3, q_4, q_5, q_6, q_{11}), a\}$

$= \varepsilon$-closure $\{\delta(q_0, a), \delta(q_1, a) \cup \delta(q_3, a)$

$\cup\, \delta(q_4, a) \cup \delta(q_5, a) \cup \delta(q_6, a) \cup \delta(q_{11}, a)\}$

$= \varepsilon$-closure $\{\phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi \cup \phi\}$

$= B$

$\boxed{\therefore\ \delta'(E, a) = B}$

$\therefore \delta'(E, b) = \varepsilon\text{-closure}\{\delta(q_0, q_1, q_3, q_4, q_5, q_6, q_{11}), b\}$

$= \varepsilon\text{-closure}\{\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_3, b) \cup \delta(q_4, b)$
$\cup \delta(q_5, b) \cup \delta(q_6, b) \cup \delta(q_{11}, b)\}$

$= \varepsilon\text{-closure}\{\phi \cup q_2 \cup \phi \cup \phi \cup \phi \cup q_7 \cup \phi\}$

$= \varepsilon\text{-closure}\{q_2\} \cup \varepsilon\text{-closure}\{q_7\}$

$= C$

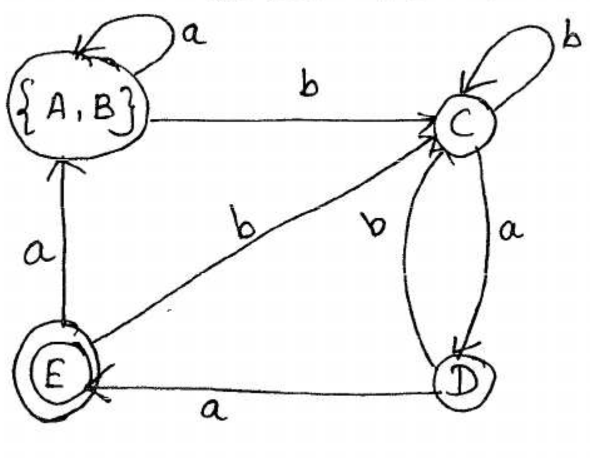
$\boxed{\delta'(E, b) = C}$

The transition table for above computed DFA is

	a	b
A	B	C
B	B	C
C	D	C
D	E	C
E	B	C

state A = B, input transitions for states A and B are same. and both are non-final states.

$\therefore$ Minimized DFA.



-

nvert the regular Expression $a(a+b)^*a$ into

NFA and find the Minimal DFA

Construct a minimized DFA from

the Expression $(x+y)\,x\,(x+y)^*$, Trace for a

String $W = xxyx$.

istinguishable and indistinguishable states

-f for some input string W, $\delta(P,W)$ and $\delta(q,W)$ oth reaches either final states or non final states, then P and q are called equivalent (or) indistinguishable states.

i.e., $\delta(P,W) \in F$, $\delta(q,W) \in F$ (or) $\left.\begin{array}{l}\delta(P,W) \notin F \\ \delta(q,W) \notin F\end{array}\right\}$ P and q are equivalent.

If for some input string W, $\delta(P,W)$ reaches final state and $\delta(q,W)$ reaches non final state then states P and q are called non equivalent (or) distinguishable states.

ie $\left.\begin{array}{l}\delta(P,W) \in F \\ \delta(q,W) \notin F\end{array}\right\}$ distinguishable states.

Minimization of DFA :-

1) State Equivalent Method.

2) Table filling Method (or) Myhill Nerode Theorem.

I. State Equivalent Method :-

Algorithm :-

Step 1 : We will create a set $\Pi_0 = \{Q_1^0, Q_2^0\}$ where Q_1^0 is set of all final states and $Q_2^0 = Q - Q_1^0$ where Q is a set of all states in DFA

Step 2 : Now we will construct Π_{K+1} from Π_K. Let Q_i^K be any subset in Π_K. If q_1 and q_2 are in Q_i^K they are $(K+1)$ equivalent provided $\delta(q_1, a)$ and $\delta(q_2, a)$ are K equivalent. Find out whether $\delta(q_1, a)$ and $\delta(q_2, a)$ are residing in the same equivalence class Π_K. Then it is said that q_1 and q_2 are $(K+1)$ equivalent. Thus Q_i^K is further divided into $(K+1)$ equivalence classes. Repeat step 2 for Q_i^K in Π_K and obtain all elements of Π_{K+1}

Step 3 : Construct Π_n for $n = 1, 2, \ldots\ldots$ until $\Pi_n = \Pi_{n+1}$

Step 4 : Then replace all the equivalent states in one equivalence class by representative state. This helps in minimizing the DFA.

STATE EQUIVALENCE METHOD

steps:

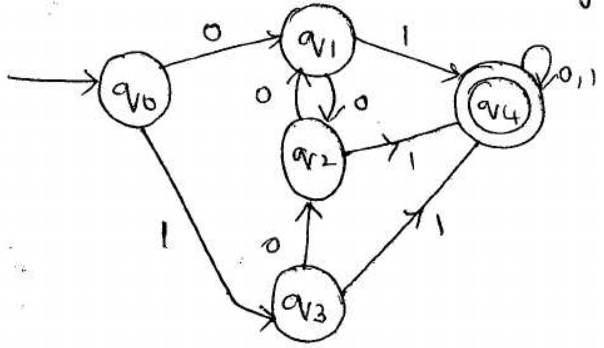
step 1: Draw transition table or transition table for the given question

step 2: Find the final and non-final state

step3: Find equivalence state

step 4: Repeat the process until the result of the last two equivalence state are same.

(1) Find out minimized DFA using state equivalence method



i) Transition table:

δ	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

(ii) Final state: $\{q_4\}$

Non final state: $\{q_0, q_1, q_2, q_3\}$

(iii) Find 0 equivalence states

$\Pi_0(1,2) = [\underbrace{\{q_4\}}_{1}, \underbrace{\{q_0, q_1, q_2, q_3\}}_{2}]$ ⇒ In 0 equivalence state write the final and non-final state

Find 1 equivalence state

	q_0	q_1	q_2	q_3
0	$(q_0,0)=q_1$ q_1 is in 2 2	$(q_1,0)=q_2$ 2	$(q_2,0)=q_1$ 2	$(q_3,0)$ q_2 2
1	$(q_0,1)=q_3$ 2	$(q_1,1)=q_4$ 1	$(q_2,1)=q_4$ 1	$(q_3,1)=q_4$ 1 → gp

$\Pi_1(1,3,4) = [\underbrace{\{q_4\}}_{1}, \underbrace{\{q_0\}}_{3}\underbrace{\{q_1, q_2, q_3\}}_{4}]$ → ①

Find 2 equivalence states

	q_1	q_2	q_3
0	4	4	4
1	1	1	1

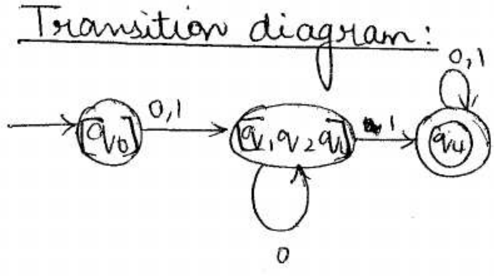
$\Pi_2(1,3,4) = [\{q_4\}, \{q_0\}, \{q_1, q_2, q_3\}]$ → ②

since ① and ② are same then stop the iteration.

Transition Table:

δ	0	1
→[q_0]	[$q_1, q_2 q_3$]	[$q_1, q_2\ q_3$]
[q_1, q_2, q_3]	[$q_1, q_2 q_3$]	[q_4]
*[q_4]	[q_4]	[q_4]

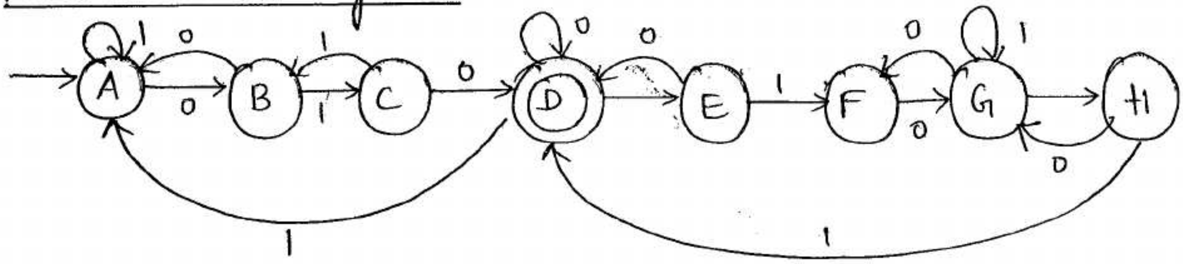
Transition diagram:



② Find minimized DFA using state equivalence method

δ	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

Transition diagram:



Final state = {D}

Non- Final state = {A, B, C, E, F, G, H}

Find 0 equivalence States

$\Pi_0(1,2) = [\underbrace{\{D\}}_{1}, \underbrace{\{A, B, C, E, F, G, H\}}_{2}]$

Find 1 equivalence States

	A	B	C	E	F	G	H
0	2	2	1	1	2	2	2
1	2	2	2	2	2	2	1

$\Pi_1(1,3,4,5) = [\underbrace{\{D\}}_{1}, \underbrace{\{C, E\}}_{3}, \underbrace{\{A, B, F, G\}}_{4}, \underbrace{\{H\}}_{5}]$