MODULE-2

14/10/2024

Regular Expressions and Finite Automata:

*) Regular ~~express~~ language :- It is the language accepted by finite automata like DFA, NFA and ε-NFA.

*) Regular expression :- It is the representation of a regular language.

Regular expression is defined as below:-

a) 'ϕ' is the regular expression denoting an empty language.

b) 'ε' is regular expression denoting empty string [zero length string]

c) 'a' is regular expression denoting language containing only 'a'. i.e., $L = \{a\}$.

d) say R_1 is a regular expression representing L_1 and R_2 is the regular expression ~~denoting~~ representing L_2, then

→ $R_1 + R_2$ is a regular expression denoting $L_1 \cup L_2$.

→ $R_1 . R_2$ is a regular expression denoting $L_1 . L_2$

(. operator → concatenation).

e) R_1^* is regular expression denoting L_1^*

f) R_1^+ is regular expression denoting L_1^+

Ex: Write R.E for language containing 0 (or) more a's over $\Sigma = \{a\}$.

Soln: $L = \{\varepsilon, a, aa, aaa, ---\}$.

$$\boxed{R.E = a^*}$$

R.E for finite language:-

1). Write a R.E for empty language.

Soln:- $\boxed{R.E = \phi}$

2) Write a R.E for language having zero length string.

Soln:- $\boxed{R.E = \varepsilon}$

3) Write R.E for language having string of length 1 over $\Sigma = \{a,b\}$.

Soln:- $L = \{a,b\}$.

$\boxed{R.E = a+b}$

4) R.E for language having string of length 2 over $\Sigma = \{a,b\}$.

Soln:- $L = \{aa, ab, ba, bb\}$.

$R.E = aa + ab + ba + bb \Rightarrow a(a+b) + b(a+b)$.

$\therefore \boxed{R.E = (a+b).(a+b)} \rightarrow$ {this is the standard representation}.

5) R.E for language having string length atmost 2 over $\Sigma = \{0,1\}$.

Soln:- $L = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$

$R.E = (0+1)(0+1)$ ← Exactly of length 2

$(\varepsilon+0+1).(0+1)$ ← String of length 1 & 2.

$\therefore$ $\boxed{\text{required } R.E = (\varepsilon+0+1).(\varepsilon+0+1)}$

Note:- If language is finite, the regular expression will generally be made up using '+' (or) '.' operators.

R.E for infinite language:-

). For infinite lang, most of the regular expression would contain '' (or) '+' operators.

1). Write the regular expression for language having strings which contains single 'b' over {a, b}.

Soln:- $L = \{b, ab, aab, aaab, \ldots\}$

Regular expression (R.E) = _ b _

more no of a's.

$\therefore \boxed{R.E = a^* b^*}$.

---

2) R.E for lang, containing all strings having atleast one 'b' over {a, b}.

Soln:- $L = \{b, ab, abaa, abbbb, \ldots\}$.

R.E = _ b _

any no of a's (or) b's => $(a+b)^*$

$\therefore \boxed{R.E = (a+b)^* b (a+b)^*}$.

---

3). R.E for lang containing strings made up of a's and b's of any length.

Soln:- $\boxed{R.E = (a+b)^*}$.

4) R.E for lang, containing "bbbb" as sub-string.

Soln:- $\boxed{R.E = (a+b)^* bbbb (a+b)^*}$ (or) $\boxed{R.E = (a+b)^* b^4 (a+b)^*}$.

5) R.E for lang containing strings that ends with "ab".

Soln:- R.E = $\underset{\downarrow}{-}$ab

any combination of a's (or) b's

∴ $\boxed{R.E = (a+b)^* ab}$

6). R.E for lang containing strings that begins and ends with 'a'.

Soln:- R.E = a $\underset{\downarrow}{-}$ a

any combination of a's (or) b's.

∴ $\boxed{R.E = a(a+b)^* a}$

7) R.E for lang containing strings that begins and ends with different symbols.

Soln:- case1:- All strings that begins with 'a' and ends with 'b'. (R1).

case2:- All strings that begins with 'b' and ends with 'a'. (R2)

R1 : $a(a+b)^* b$

R2 : $b(a+b)^* a$

∴ Required R.E = R1 + R2

$\boxed{R.E = (a(a+b)^* b) + (b(a+b)^* a)}$

8) R.E for the lang. in which 3rd symbol in the strings from right is a.

Soln: R.E = $\underbrace{}_{\text{any combination of a's (or) b}}$ a $\overbrace{\quad}^{\text{a (or) b}}$ → 3rd symbol from right.

any combination of a's (or) b.

a (or) b.

$\therefore \boxed{R.E = (a+b)^* a (a+b)(a+b)}$

9) R.E for the lang. containing even length string over {a,b}.

Soln: $\boxed{R.E = [(a+b)(a+b)]^*}$

$\rightarrow [(a+b)(a+b)]^0 \cup [(a+b)(a+b)]^1 \cup [(a+b)(a+b)]^2 \cup \ldots$

'ε'

aa
bb
ab
ba

aaaa
aabb
abab
⋮

10) R.E for the lang. containing odd length string over {a,b}.

Soln: $\boxed{R.E = [(a+b)(a+b)]^* (a+b)}$

11) R.E for the lang. containing strings that has exactly two a's.

Soln: R.E = $\_\, a \,\_\, a \,\_$

any number of b's.

$\therefore \boxed{R.E = b^* a b^* a b^*}$

11) R.E for lang containing strings that have atmost 2 a's.

Soln:- At most 2 a's ⇒ 0 : $L_1 = b^*$

1 : $L_2 = b^* a b^*$

2 : $L_3 = b^* a b^* a b^*$

∴ Required R.E = $L_1 + L_2 + L_3$

$$\boxed{R.E = b^* + (b^* a b^*) + (b^* a b^* a b^*)}$$

12) Given, $L = \{w : n_0(w) \bmod 2 = 0\}$ over $\Sigma = \{0,1\}$

Soln:- $n_0(w) \bmod 2 = 0$ ⇒ no of 0's should be even.

$L = \{\varepsilon, 00, 1010, 0000, 110000, \ldots\}$.

$(00)^*$ represents even no of 0's.

$$\boxed{R.E = 1^* + (1^* 0 1^* 0 1^*)^*}$$

13) Given, $L = \{w : n_0(w) \bmod 2 = 1\}$ over $\Sigma = \{0,1\}$.

Soln:-

$$\boxed{R.E = 1^* 0 1^* (1^* 0 1^* 0 1^*)^*}$$

14) Construct R.E for lang containing any combination of 0's and 1's except null string.

Soln:-

$$\boxed{R.E = [0+1]^+}$$

15). Given, $L = \{a^{2n} b^{2m+1} : m \geq 0, n \geq 0\}$.

Soln:- $\boxed{R.E = (aa)^* (bb)^* b}$

16). Given, $L = \{a^m b^m : (m+n) \text{ is even}\}$.

Soln:- case 1:- m & n are odd $(2k+1) = (aa)^* a (bb)^* b$

case 2:- m & n are even $= (aa)^* (bb)^*$

$\therefore \boxed{R.E = (aa)^* a (bb)^* b + (aa)^* (bb)^*}$

17) Given, $L = \{a^n b^m : n \geq 4 \text{ and } m \leq 3\}$

Soln:- $a^n, n \geq 4 \Rightarrow (aaaa) a^*$

$b^m, m \leq 3 \Rightarrow \varepsilon + b + bb + bbb$.

$\therefore \boxed{R.E = (aaaa)a^* + (aaaa)a^* b + (aaaa)a^* bb + (aaaa)a^* bbb}$

(or)

$\boxed{R.E = (aaaa)a^* (\varepsilon + b + bb + bbb)}$

18) R.E for language whose strings contain exactly one '1' and even no of 0's.

Soln:- ~~Re~~ 1

(00)*

$\therefore \boxed{R.E = (00)^* 1 (00)^*}$

19) Language containing strings in which 10th symbol from right is '1'.

Soln:- $\boxed{R.E = (0+1)^* 1 (0+1)^9}$

Describing the regular expression:-

*) ϕ → this R.E represents an empty language.

*) ϵ → this R.E represents language containing empty string.

*) $a+b$ → this R.E represents language containing either a (or) b.

*) ab → this regular expression represents language containing the string a followed by b.

*) $a(a+b)^*b$ → this R.E represents language containing string that starts with 'a' and ends with 'b'.

*) $a(a+b)^*ab$ → this R.E represents lang containing strings that start with 'a' and ends with 'ab'.

*) $(1^*01^*0^*)^*$ → this R.E represents lang containing strings with even no of 0's.

*) a^*b+b^*a → this R.E represents lang containing string with 0 (or) more no of 'a's' followed by single 'b' (or) 0 (or) more no of 'b's' followed by single 'a'.

*) $(a+b)^*a(a+b)^7$ → this R.E represents lang containing string with a as the 8^{th} character from right.

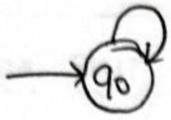
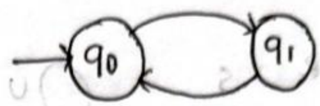
*). $(a+b)^*aa(bb)^*aa(a+b)^*$ → this R.E represents lang in which every string should have even no of b's b/w any two occurances of a's.

15/10/2024

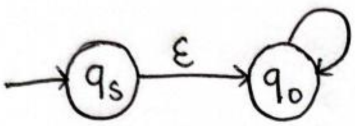
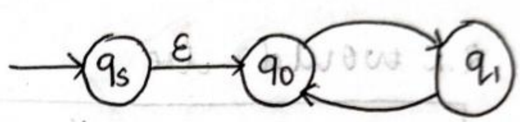
Conversion of FA into R.E using state Elemination Method:-

Basic step:- If there are any dead state $\longrightarrow$ remove it.

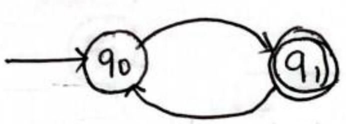
S1:- Check whether the start state has an incoming edge.

ie.,  (or) , if there,

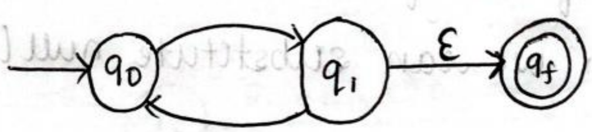
then consider a new start state

 (or) 

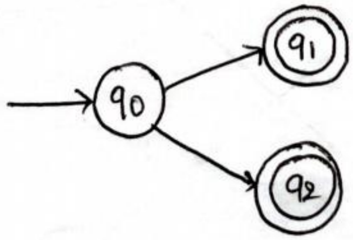
S2:- Check whether the final state has an outgoing edge,

ie., , if there,

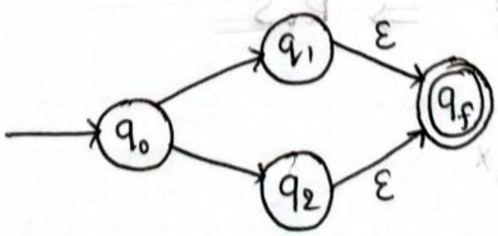
then consider a new final state.



S3:- Check whether FA is having multiple final states,

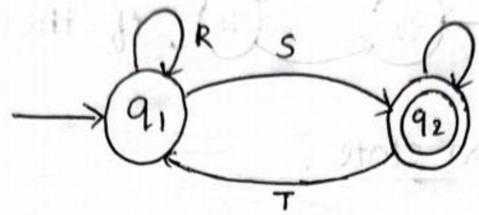
ie., , if there,

then combine the final states and make it has single final state.

ie., 

S4:- Eliminate all states one by one except initial and final state.

Notes:- After Eliminating all states, let us consider the finite automata looks as below.

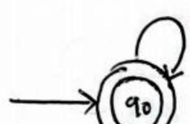


then, R.E would be

$$\boxed{R.E = (R + SU^*T)^* SU^*}$$

here R, S, U, T are all Regular expressions.

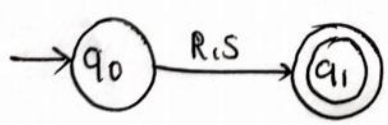
Notes:-



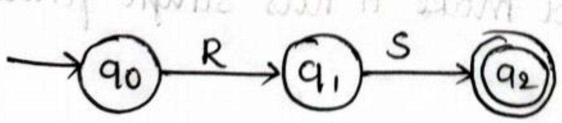
$\Rightarrow R^*$ · {usefull notation}

* while solving problem, if the final automata does not contain this (R, S, U, T), then we can substitute null ($\emptyset$) in its place.

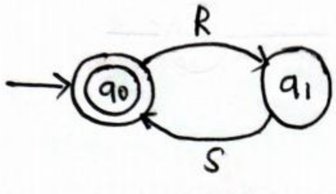
Usefull notations { conti :-



$\Rightarrow R + S$



$\Rightarrow R.S$



$\Rightarrow (RS)^*$

Note:-

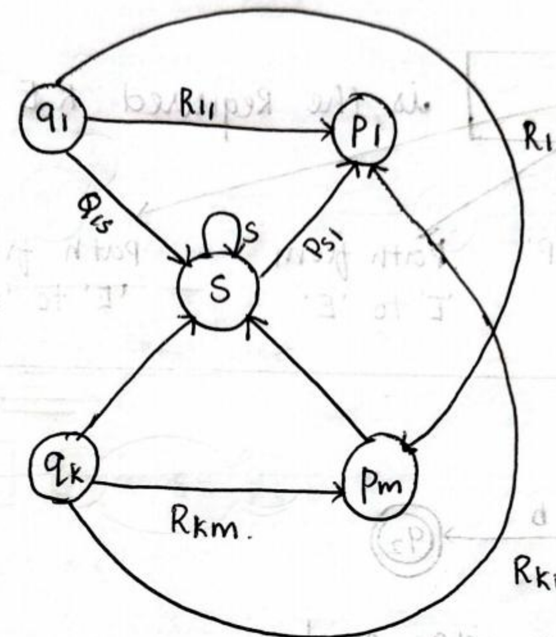
$\phi . R = \phi$

$\phi + R = R$

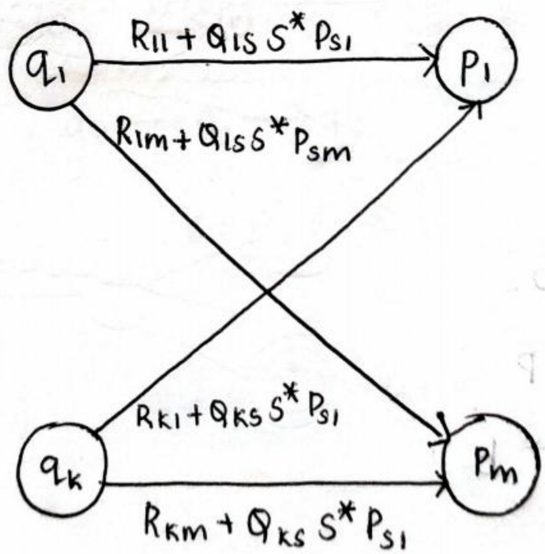
$(\phi)^* = \varepsilon$.

Method for Eliminating state:-

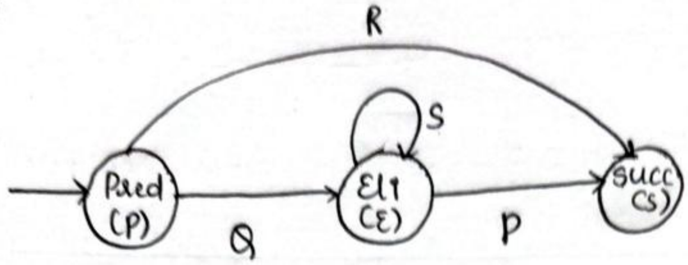
Consider,



$\Rightarrow$



Note:-



$\boxed{R + QS^* P}$ is the Required R.E.

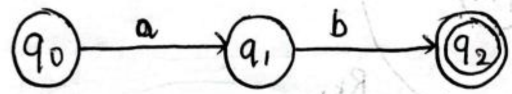
Direct path from 'P' to 'S'

Path from 'P' to 'E'

Path from 'E' to 'E'

Path from 'E' to 'S'

1). Given,



Soln:- 'q_0' has no incoming edge and 'q_2' has no outgoing edge

∴ Eliminate q_1

Consider $q_0 \rightarrow q_1 \rightarrow q_2$

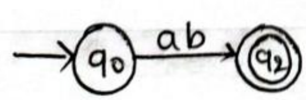
$R = \phi$, $Q = a$, $S = \phi$, $P = b$.

we have $R.E = R + QS^*P$

$= \phi + a.\phi^*.b$

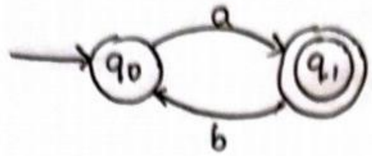
$= a.\varepsilon.b$

$= ab$.



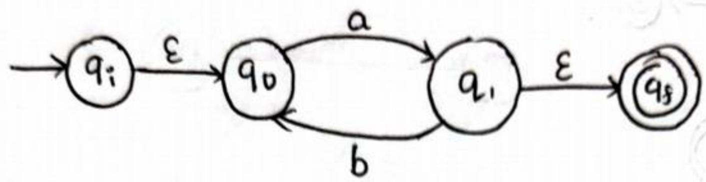
∴ Required $\boxed{\text{R.E is ab.}}$

2). Given,



Soln: ~~required R.E as~~

~~R.E =~~ 'q0' has incoming edge & 'q1' has outgoing edge. so we consider new initial and final states.

$\Rightarrow$ 

Eleminating q_1 :-

i) consider $q_0 \rightarrow q_1 \rightarrow q_f$.

$R = \phi,\ Q = a,\ S = \phi,\ P = \varepsilon$.

$R.E = R + QS^*P$

$= \phi + a(\phi)^*\varepsilon$.

$= a.\varepsilon.\varepsilon$

$= a. \longrightarrow (1)$.

ii) Consider, $q_0 \rightarrow q_1 \rightarrow q_0$.

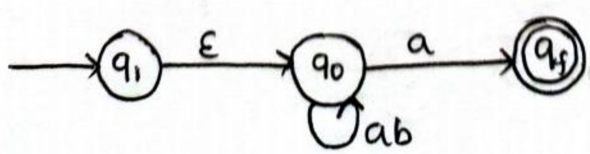
$R = \phi,\ Q = a,\ S = \phi,\ P = b$.

$R.E = R + QS^*P$

$= \phi + a(\phi)^* b$.

$= a.\varepsilon.b$

$= ab \longrightarrow (2)$.

$\therefore$ 

Eleminating q_0:-

Consider $q_i \longrightarrow q_0 \longrightarrow q_f$.

$R=\phi,\ Q=\varepsilon,\ S=ab,\ P=\cancel{\phi}\,a$

$\therefore$ $R.E = R + QS^*P$

$= \phi + \varepsilon.(ab)^*a$

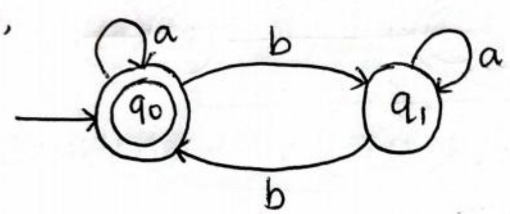
$= (ab^*)^*.a \longrightarrow$ 3 .

$\therefore \longrightarrow q_i \xrightarrow{(ab)^*a} q_f$

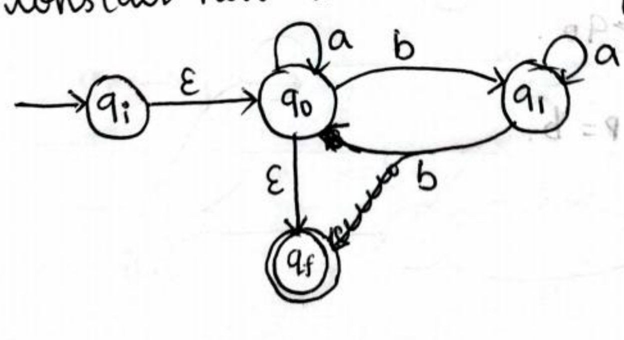
$\therefore$ required Regular expression is $\boxed{(ab)^* a.}$

$\boxed{16/10/2024}$

3). Given,



Soln:- 'q0' has an incoming edge and an outgoing edge, so consider new initial and final states.



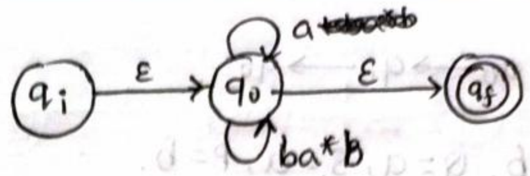
Eleminating q_1:-

$$\begin{tikzcd} q_0 \arrow[r] \arrow[rr, bend left, "\varepsilon"] & q_1 \arrow[r] & q_f \end{tikzcd}$$

$R=\phi,\ Q=b,\ S=a,\ P=b.$

$\therefore R.E = \phi + ba^*b.$

$= ba^*b.$



Eliminating q_0:-

$q_i \longrightarrow q_0 \longrightarrow q_f.$ $\{R=\phi,\ Q=\varepsilon,\ S=(a+ba^*b)^*,\ P=\varepsilon\}.$

$R.E = \phi + \varepsilon.(a+ba^*b)^*\varepsilon$

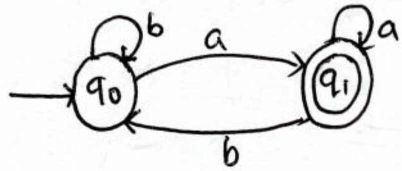
$= (a+ba^*b)^*$

$\therefore \longrightarrow q_i \xrightarrow{(a+ba^*b)^*} q_f.$

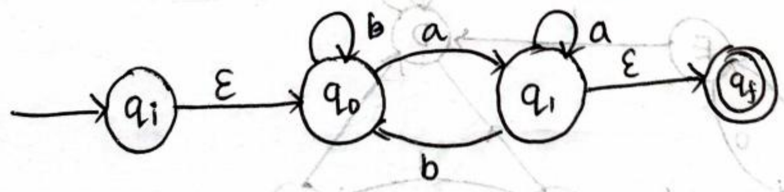
required $\boxed{R.E : (a+ba^*b)^*}$

---

4) Given,



Soln:- 'q_0' has an incoming edge and q_1 has an outgoing edge, consider new initial and final states.



Eliminating q_1:-

case i:- $q_0 \longrightarrow q_1 \longrightarrow q_f.$

$R=\phi,\ Q=a,\ S=a,\ P=\varepsilon.$

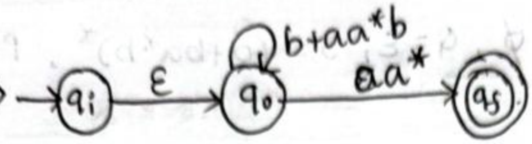
we have $R+QS^*P = \phi + aa^*\varepsilon.$

$= aa^*$

case ii:- $q_0 \rightarrow q_1 \rightarrow q_0$.

$R = b,\ Q = a,\ S = a,\ P = b.$

we have $R + QS^*P = b + aa^*b$

$\Rightarrow$ 

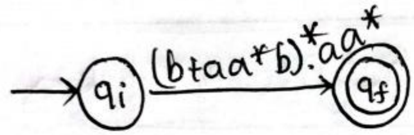
Eliminating q_0:-

$q_i \rightarrow q_0 \rightarrow q_f$.

$R = \phi,\ Q = \varepsilon,\ S = b + aa^*b,\ P = aa^*$

$R + QS^*P = \phi + \varepsilon.(b + aa^*b)^*.aa^*$

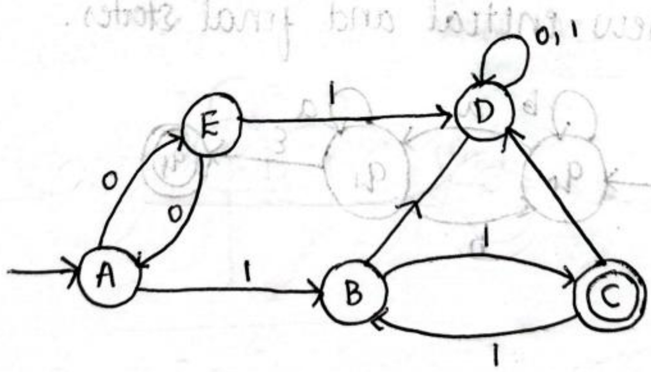
$= (b + aa^*b)^*.aa^*$.

$\therefore$ 

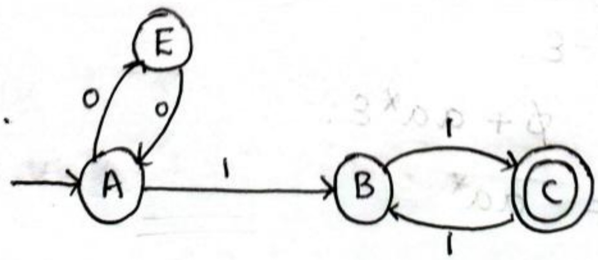
$\therefore$ required $\boxed{R.E = (b + aa^*b)^*.aa^*}$

***.

5). Given,

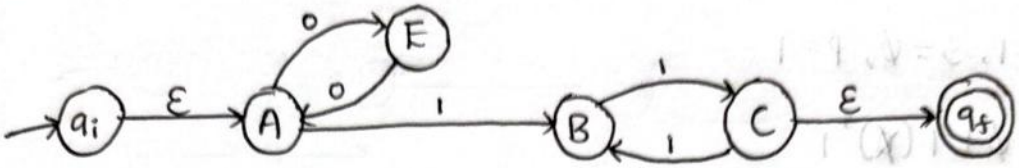


Soln:- D is a dead state $\Rightarrow$ eliminate it.



'A' has incoming edge ⇒ consider new state (initial)

'ε' has outgoing edge ⇒ consider new state (final).



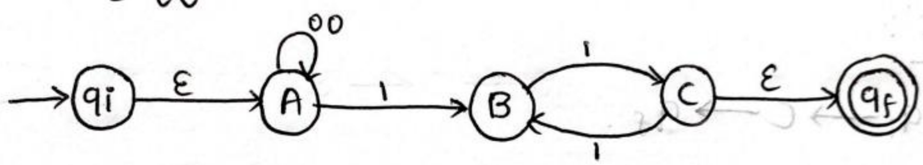
Eliminating 'E':-

consider $A \rightarrow E \rightarrow A$.

$R = \phi,\ Q = 0,\ S = \phi,\ P = 0$.

$R + QS^*P = \phi + 0(\phi)^*0$

$= 00$

∴ 

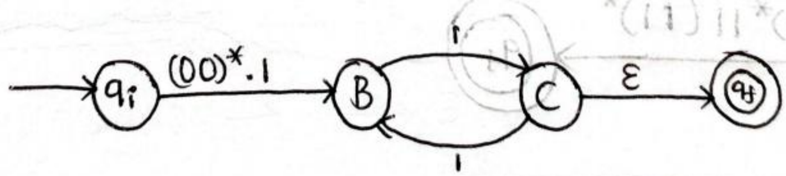
Eliminating 'A':-

consider $q_i \longrightarrow A \longrightarrow B$.

$R = \phi,\ Q = \varepsilon,\ S = 00,\ P = 1$

$R + QS^*P = \phi + \varepsilon.(00)^*.1$

$= (00)^*1$

⇒ 

Eliminating 'B':-

case 1:- $q_i \longrightarrow B \longrightarrow C$.

$R = \phi,\ Q = (00)^*.1,\ S = \phi,\ P = 1$.

$R + QS^*P = \phi + (00)^*1.(\phi)^*.1 = (00)^*11$

case 2:- consider

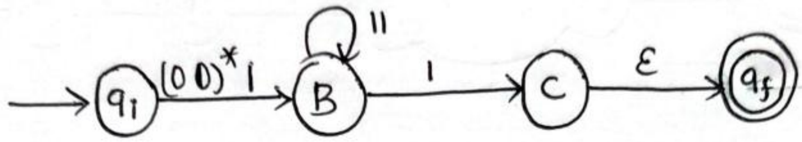
$B \longrightarrow C \longrightarrow B$.

$R = \phi, Q = 1, S = \phi, P = 1$

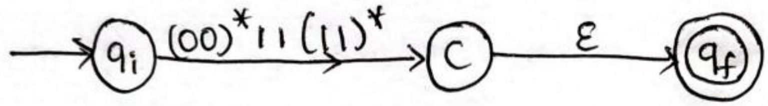
$R + QS^*P = \phi + 1(\cancel{\phi})^* 1$

$= 11.$

$\therefore$



and



Eliminating 'c':

consider $q_i \longrightarrow C \longrightarrow q_f$.

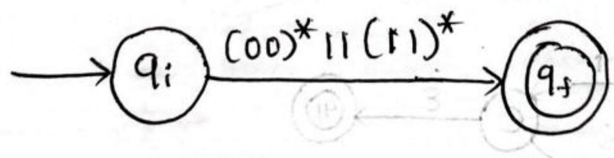
$R = \phi,\ Q = (00)^* 11 (11)^*,\ S = \phi,\ P = \varepsilon.$

$\therefore R + QS^*P = (00)^* 11 (11)^* . (\phi)^* . \varepsilon$

$= (00)^* 11 (11)^* . \varepsilon . \varepsilon$

$= (00)^* 11 (11)^*$

$\therefore$

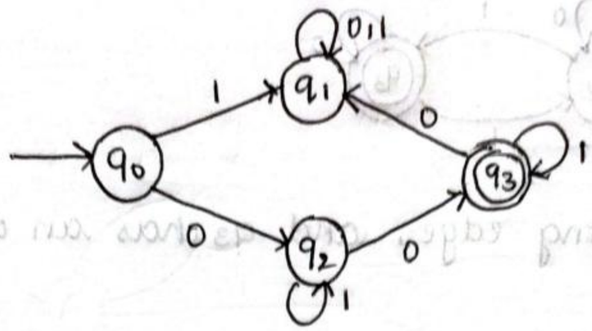


$\therefore$ required $\boxed{R.E = (00)^* 11 (11)^*}$

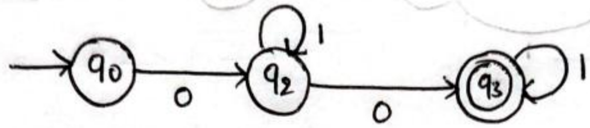
Imp

Note:- Its better to eliminate the states from the critical point (first to last).

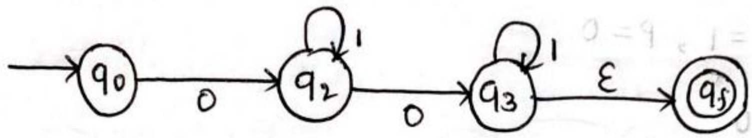
6). Given,



Soln:- q_1 is dead state $\Rightarrow$ eliminate it.



q_3 has an outgoing edge, consider new final state.



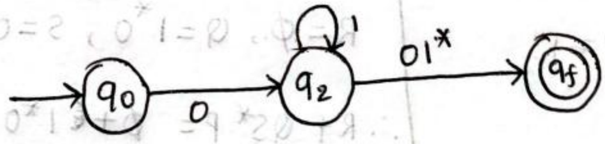
Eliminating q_3:-

$q_2 \rightarrow q_3 \rightarrow q_f$.

$R = \phi$, $Q = 0$, $S = 1$, $P = \varepsilon$.

$R + QS^*P = \phi + 01^*\varepsilon$.

$= 01^*$.

$\Rightarrow$ 

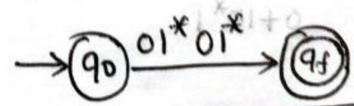
Eliminating q_2:-

$q_0 \rightarrow q_2 \rightarrow q_f$.

$R = \phi$, $Q = 0$, $S = 1$, $P = 01^*$.

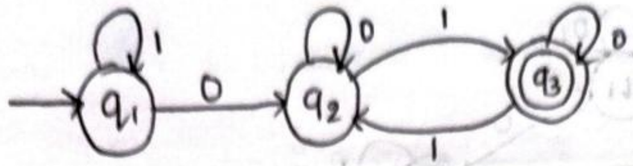
$R + QS^*P = \phi + 01^*01^*$

$= 01^*01^*$

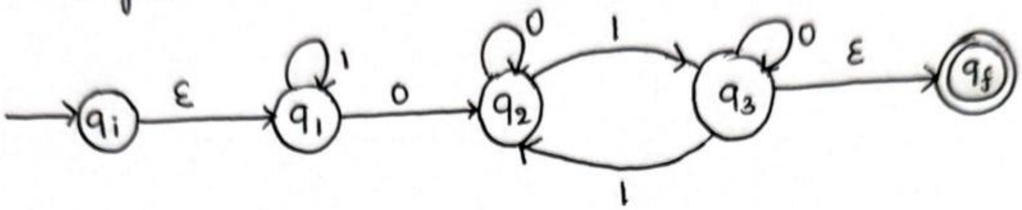
$\Rightarrow$ 

$\therefore$ required $\boxed{R.E = 01^*01^*}$

7. Given,



Soln:- q_1 has an incoming edge, and q_3 has an outgoing edge.



Eliminating q_1:-

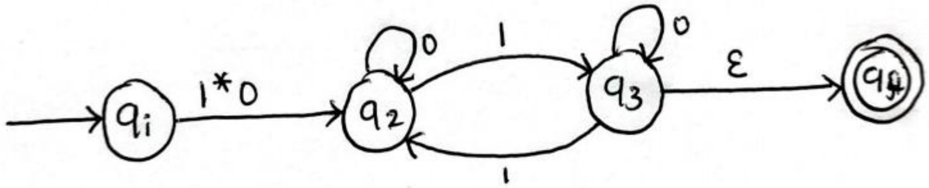
consider $q_i \longrightarrow q_1 \longrightarrow q_2$.

here $R = \phi$, $Q = \varepsilon$, $S = 1$, $P = 0$

$R + QS^*P = \phi + \varepsilon . 1^*0$

$= 1^*0$

$\Rightarrow$.



Eliminating q_2:-

case 1:- consider $q_3 \longrightarrow q_2 \longrightarrow q_3$.

$R = 0$, $Q = 1$, $S = 0$, $P = 1$

$R + QS^*P = 0 + 1(0)^*1$

$= 0 + 10^*1$

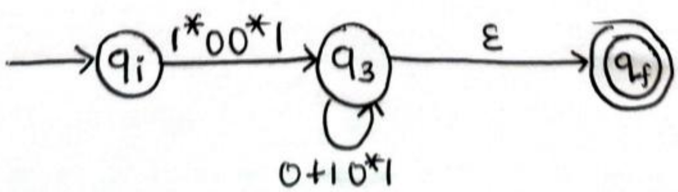
case 2:- consider $q_i \longrightarrow q_2 \longrightarrow q_3$

$R = \phi$, $Q = 1^*0$, $S = 0$, $P = 1$

$R + QS^*P = \phi + 1^*0 . 0^* . 1$

$= 1^*00^*1$

$\Rightarrow$.



Eliminating q_3:-

consider $q_i \longrightarrow q_3 \longrightarrow q_f$

$R = \phi$, $Q = 1^*00^*1$, $S = (0+10^*1)$, $P = \varepsilon$.

$\therefore R + QS^*P = \phi + 1^*00^*1 \,.\, (0+10^*1)^* \,.\, \varepsilon$

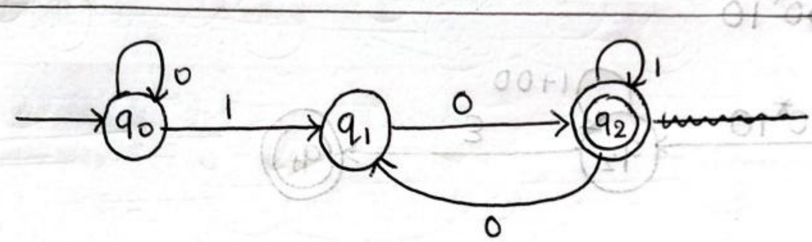
$= 1^*00^*1 \,.\, (0+10^*1)^*$

$\Rightarrow \longrightarrow (q_i) \xrightarrow{1^*00^*1.(0+10^*1)^*} ((q_f))$

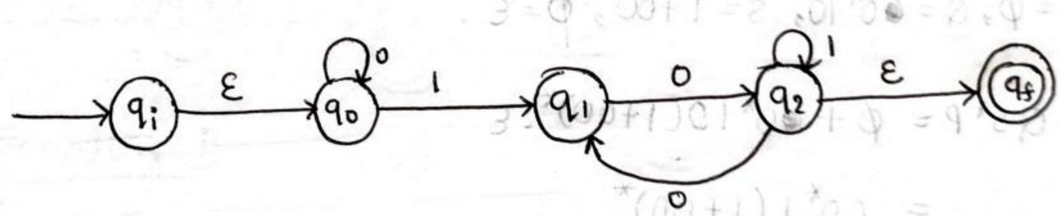
required $\boxed{RE = 1^*00^*1(0+10^*1)^*}$

---

8) Given,



Soln:- q_0 has incoming edge and q_2 has an outgoing edge, consider new initial and final states.



Eliminating q_0:-

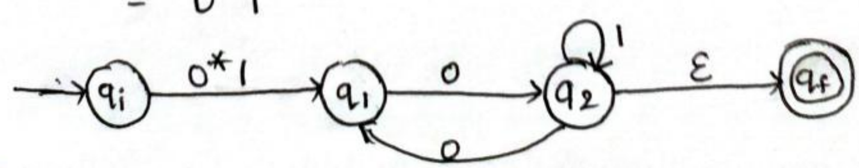
consider $q_i \longrightarrow q_0 \longrightarrow q_1$.

$R = \phi$, $Q = \varepsilon$, $S = 0$, $P = 1$

$\therefore R + QS^*P = \phi + \varepsilon.0^*.1$

$= 0^*1$

$\Rightarrow$



Eliminating q_1:-

case1:- consider $q_2 \rightarrow q_1 \rightarrow q_2$.

$R = 1,\ Q = 0,\ S = \phi,\ P = 0$.

$\therefore R + QS^*P = 1 + 0.\phi^*.0$

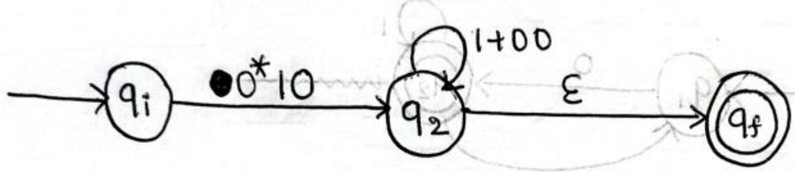
$= 1 + 00$

case2: consider $q_i \rightarrow q_1 \rightarrow q_2$.

$R = \phi,\ Q = 0^*1,\ S = \phi,\ Q = 0$.

$\therefore R + QS^*P = \phi + 0^*1.\phi^*.0$

$= 00^*10$

$\therefore$ 

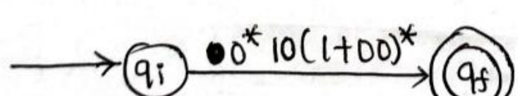
Eliminating q_2:-

consider $q_i \rightarrow q_2 \rightarrow q_f$.

$R = \phi,\ Q = 00^*10,\ S = 1 + 00,\ P = \varepsilon$.

$\therefore R + QS^*P = \phi + 00^*10(1+00)^*.\varepsilon$

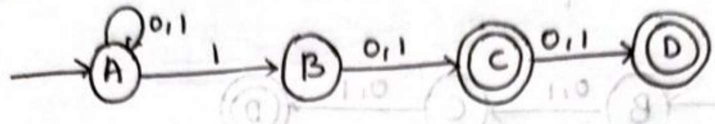
$= 00^*1(1+00)^*$

$\therefore$ 

$\therefore$ required $\boxed{R.E = 00^*10(1+00)^*}$

17/10/2024

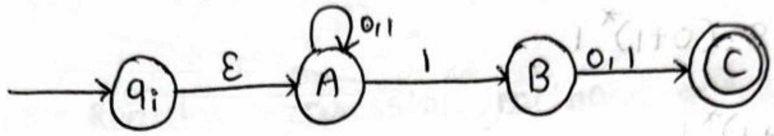
a). Given,



Sol: 'A' has an incoming edge and we have two final states.

case1:- considering 'C' as final state.

$\Rightarrow$



Eliminating A:

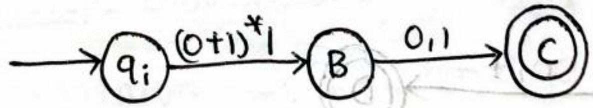
consider $q_i \longrightarrow A \longrightarrow B$.

$R = \phi,\ Q = \varepsilon,\ S = 0+1,\ P = 1.$

$\therefore\ R + QS^*P = \phi + \varepsilon.(0+1)^* 1.$

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

$\Rightarrow$



Eliminating B:-

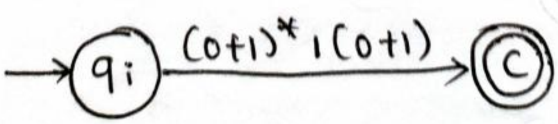
consider $q_i \longrightarrow B \longrightarrow C$.

$R = \phi,\ Q = (0+1)^* 1,\ S = \phi,\ P = (0+1).$

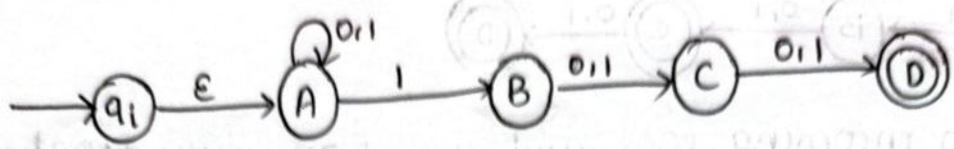
$R + QS^*P = \phi + (0+1)^* 1.\varepsilon.(0+1)$

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

$\therefore$



case ii:- Considering 'D' as final state.

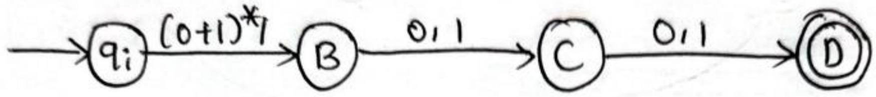


Eliminating A:

consider $q_i \rightarrow A \rightarrow B$.

$\Rightarrow R + QS^*P = \phi + \varepsilon.(0+1)^* 1$

$= (0+1)^* 1$

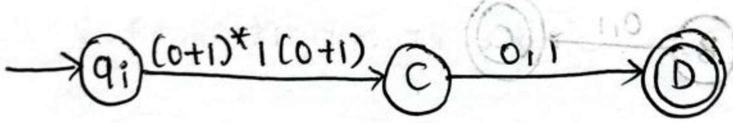


Eliminating B:

consider $q_i \rightarrow B \rightarrow C$.

$\Rightarrow R + QS^*P = \phi + (0+1)^* 1.\varepsilon.(0+1)$

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



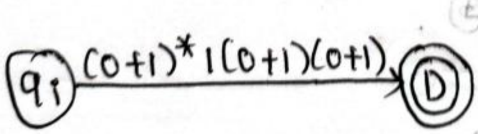
Eliminating C:

consider $q_i \rightarrow C \rightarrow D$.

$\Rightarrow R + QS^*P = \phi + (0+1)^* 1(0+1).\varepsilon.(0+1)$

$= (0+1)^* 1(0+1)(0+1) \longrightarrow (2)$.

$\therefore$



① + ②

$\Rightarrow$ required $\boxed{R.E = (0+1)^* 1 (0+1) + (0+1)^* 1 (0+1)(0+1)}$

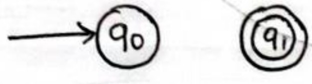
Conversion of R.E into F.A:-

If 'r' is the regular expression then there exists a finite automata (ϵ-NFA) that accepts (r).

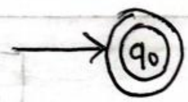
Basic step:- Regular expression without operations

i.e., ϕ (or) ϵ (or) a.

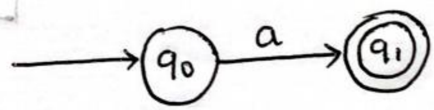
'ϕ':



'ϵ':



'a':

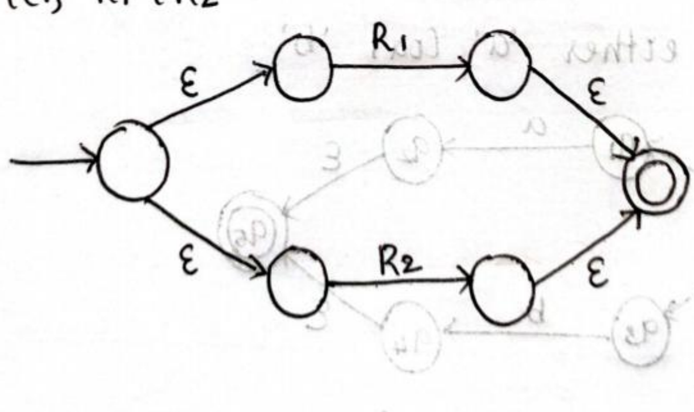


Induction step:- Consider regular expressions having operations.

a) union b) Concatenation c) Closure

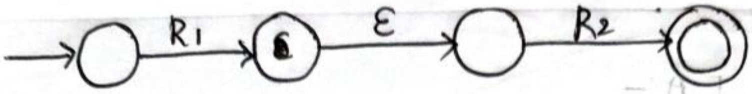
case 1:- union of two regular expressions.

i.e., $R_1 + R_2$



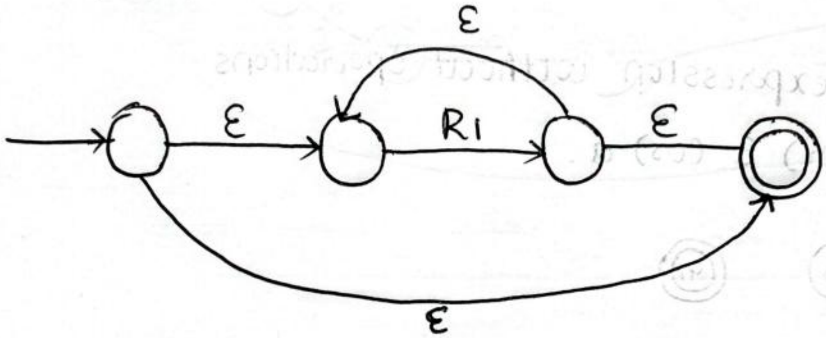
case 2:- Concatenation of two R.E's

i.e., R1. R2



case 3:- closure of R.E

i.e., R_1^*

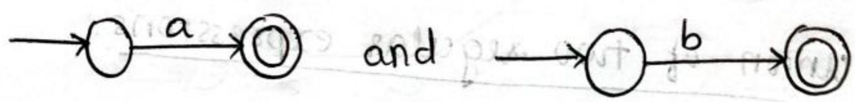


21/10/2024

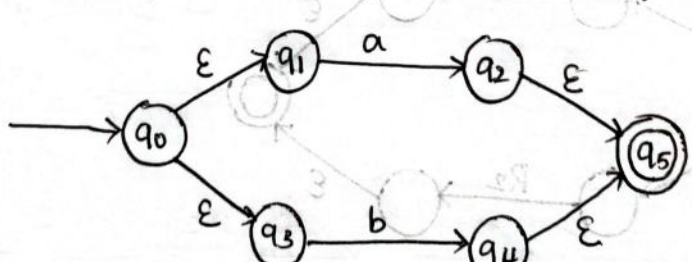
1). Given, R.E = a+b.

Soln:- Let RE1 = accepts only 'a'

RE2 = accepts only 'b'

i.e., 

∴ RE3 = accepts either 'a' (or) 'b'

⇒ 

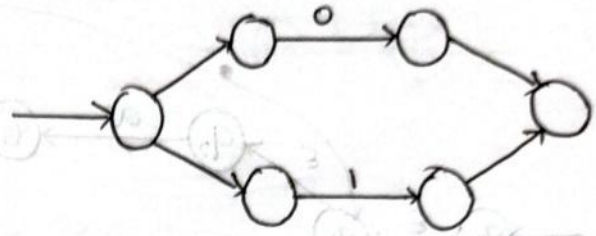
is the required F.A

2) Given $1(0+1)$

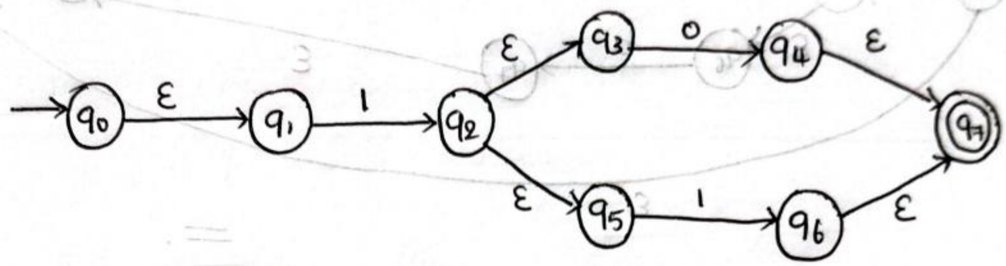
Soln:- Let RE1 = accepts '1'

RE2 = accepts either '0' (or) '1'

and



⇒

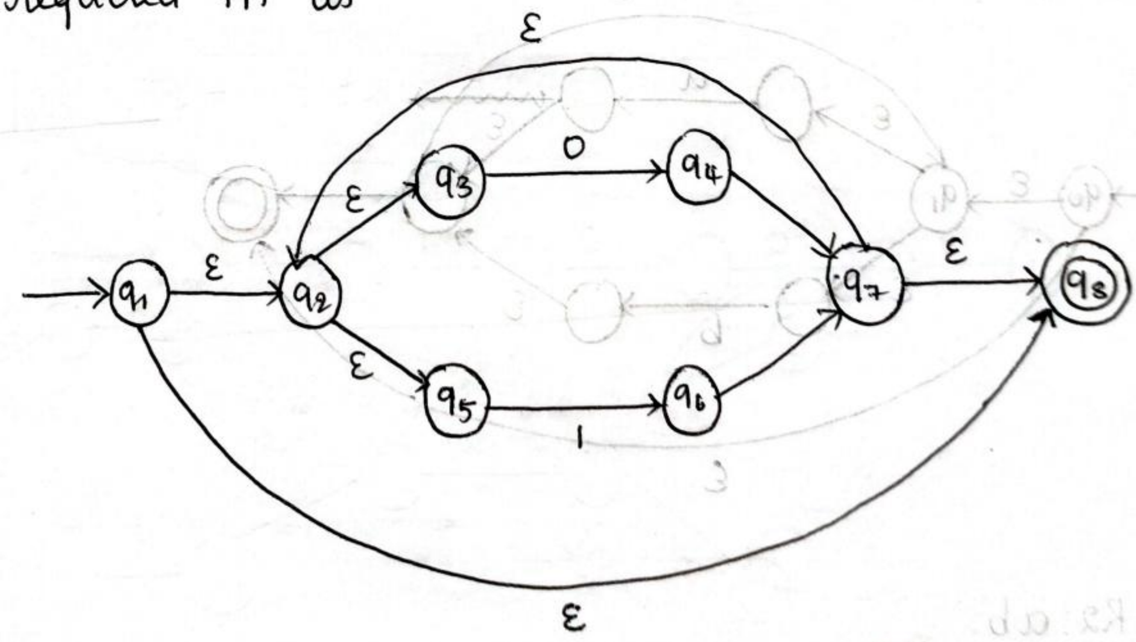


is the required F.A.

---

3) $(0+1)^*$

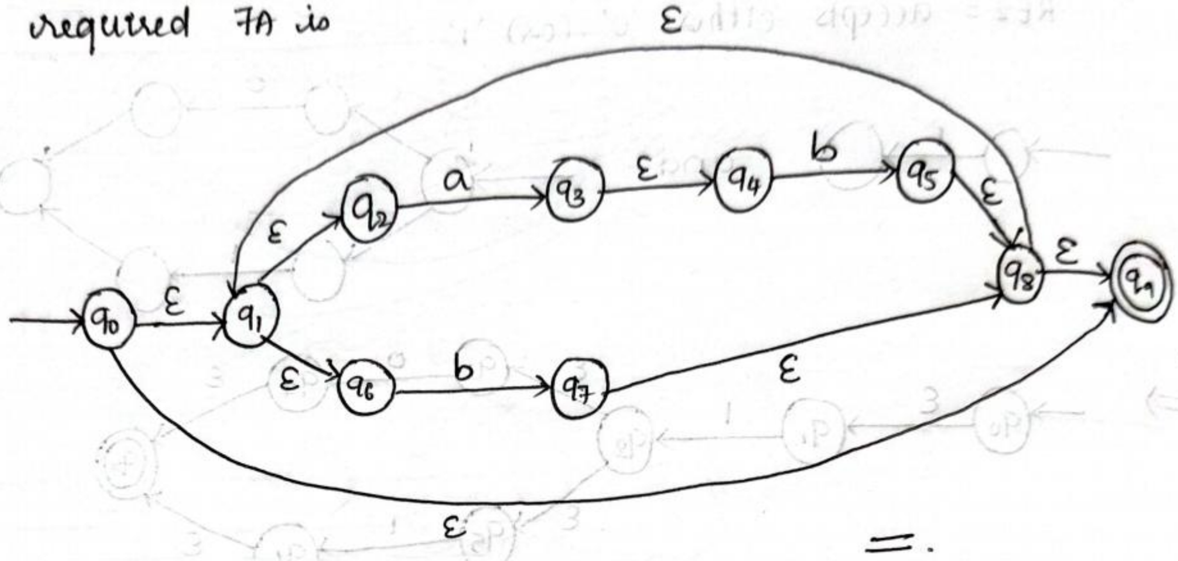
Soln:- required FA is



4. $(ab+b)^*$

Soln:

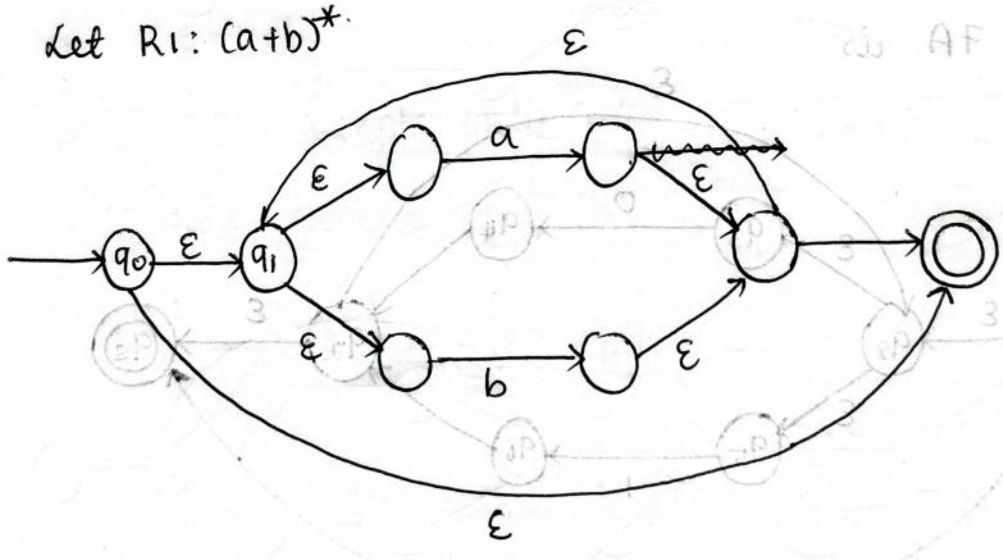
required FA is



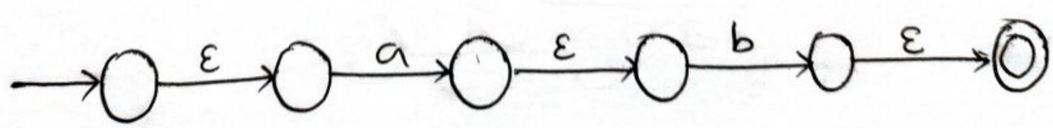
5). Given, $(a+b)^*.ab$

Soln: required FA is

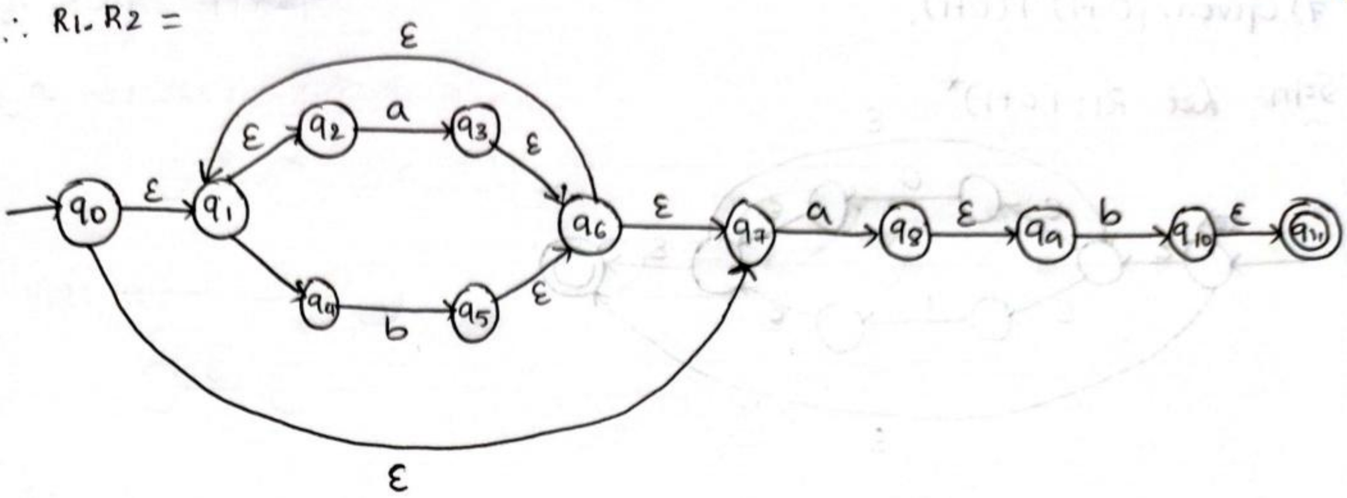
Let R1: $(a+b)^*$.



Let R2: ab.



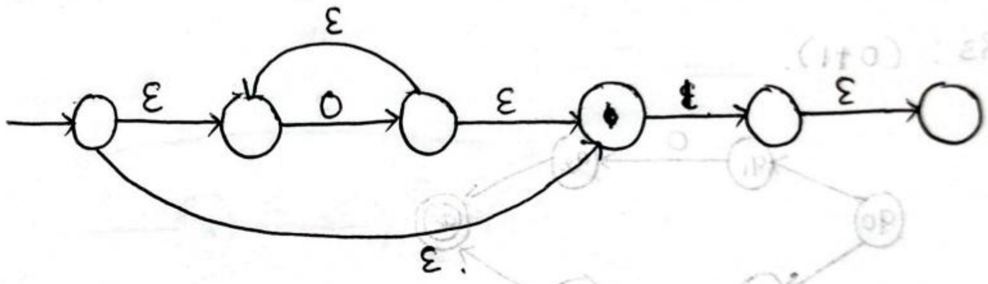
∴ $R_1 . R_2$ =



6}. Given, 0^*1+10.

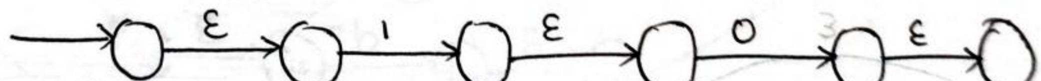
Sol: Let R_1: 0^*1

⇒

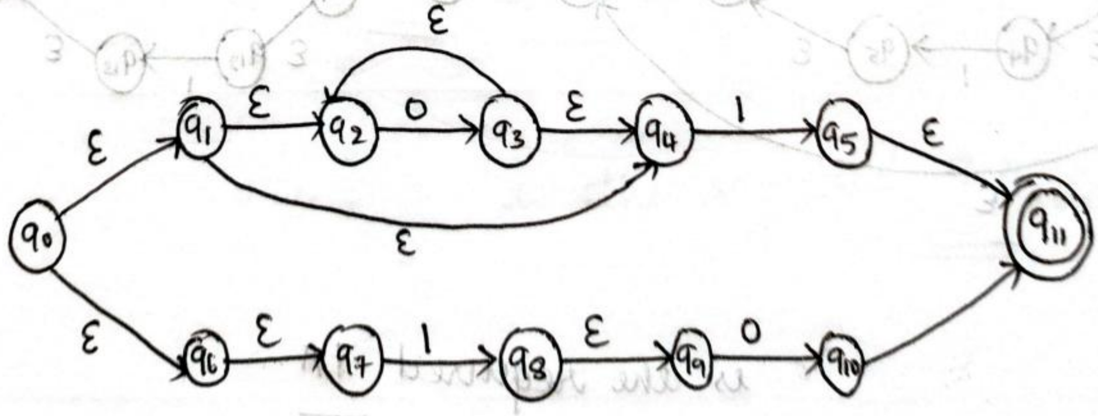


R_2: 10

⇒



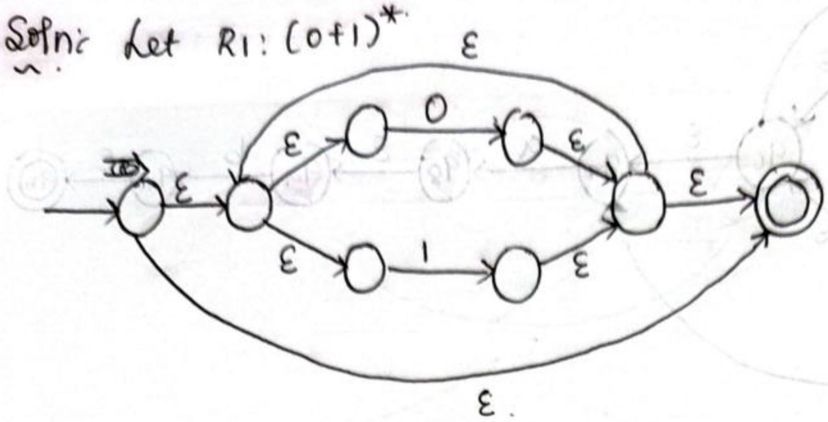
∴ $R_1 + R_2$.



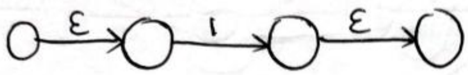
is the required FA.

7). Given, $(0+1)^*1(0+1)$.

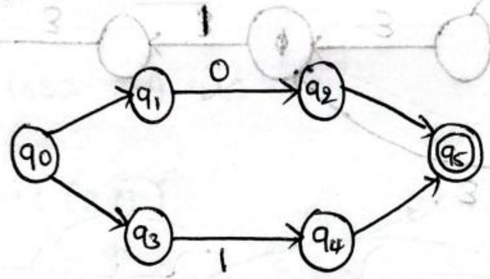
Soln: Let $R_1 : (0+1)^*$



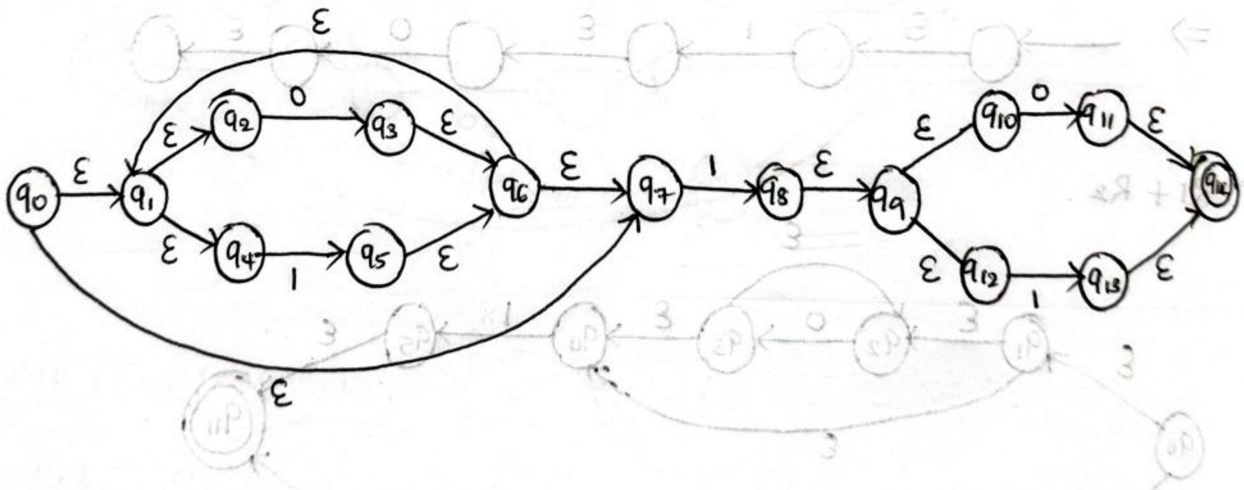
$R_2 : 1$

$\Rightarrow$ 

and $R_3 : (0+1)$.

$\Rightarrow$ 

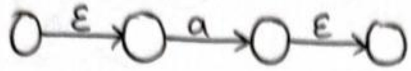
$\therefore R_1 . R_2 . R_3 =$.



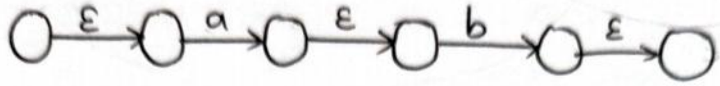
is the required FA.

8). Given, $(a+ab+b)^*$.

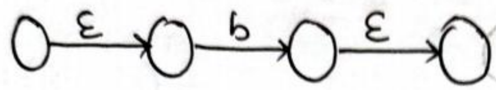
Soln: R_1: a.



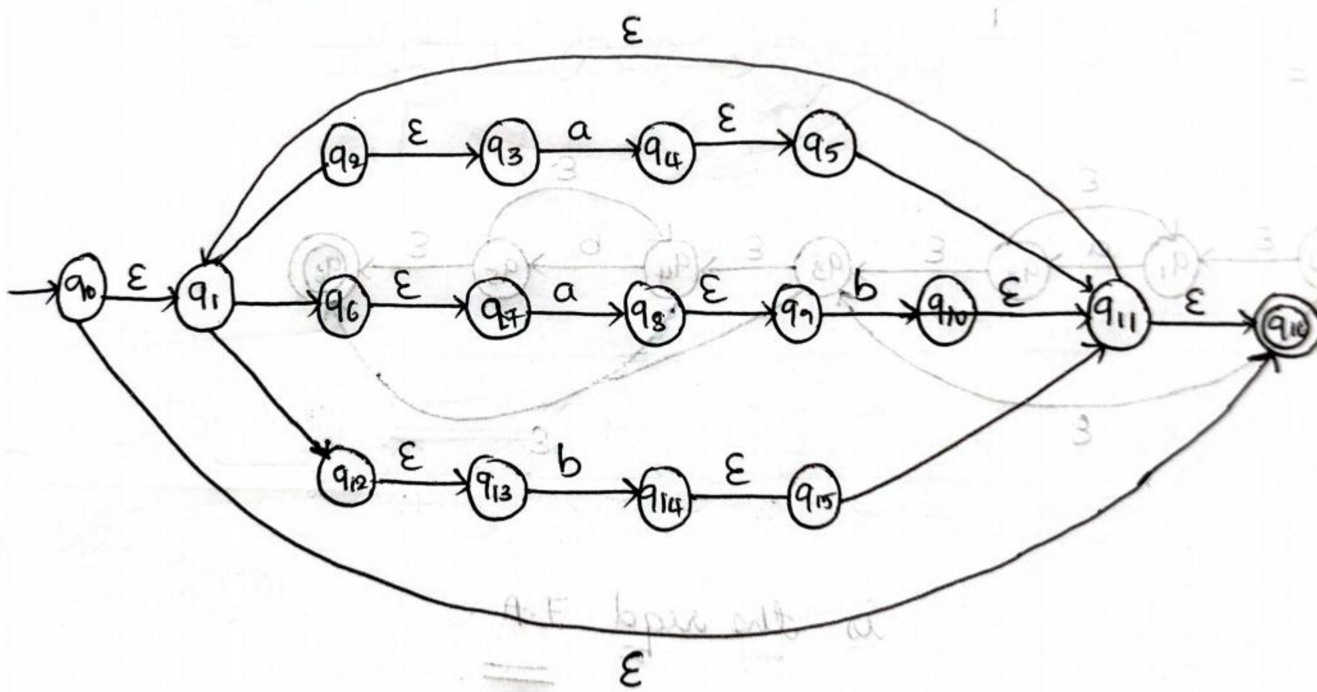
R_2: ab.



R_3: b.



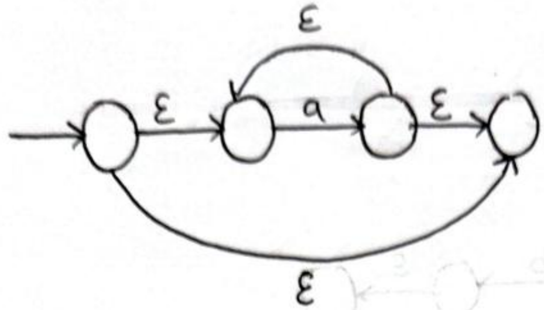
$(R_1+R_2+R_3)^* =$.



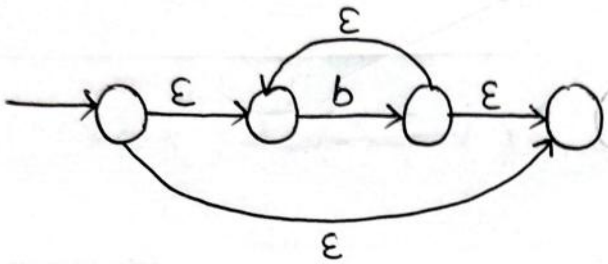
is the required F.A.

9). Given, $a^* b^*$

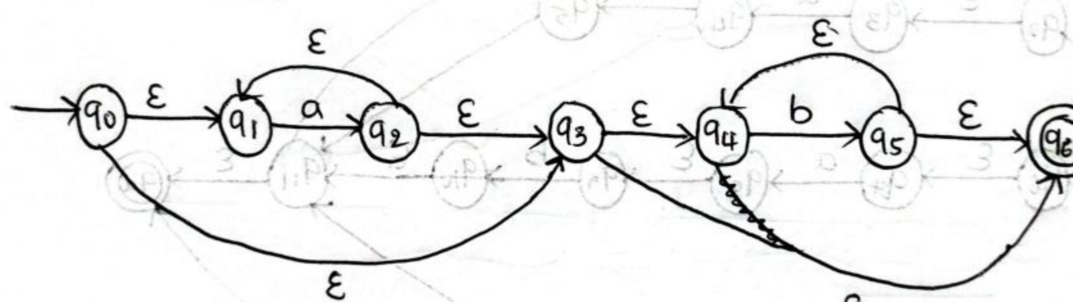
Soln:- Let R1: a^*



R2: b^*



R1.R2 =

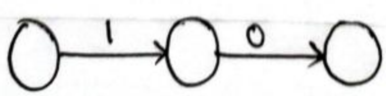


is the reqd F.A.

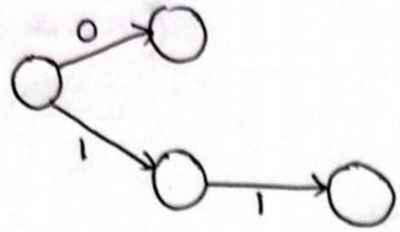
---

10). Given $10+(0+11)0^*1$ (only NFA) not ε-NFA.

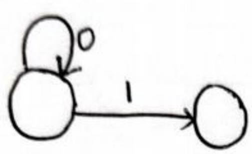
Soln:- Let R1: 10



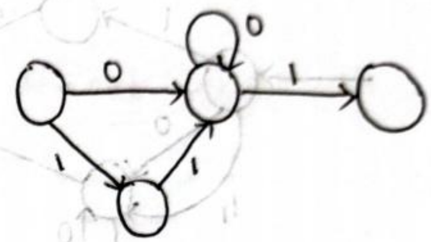
$R_2 : 0+11$



and $R_3 : 0^* 1$.

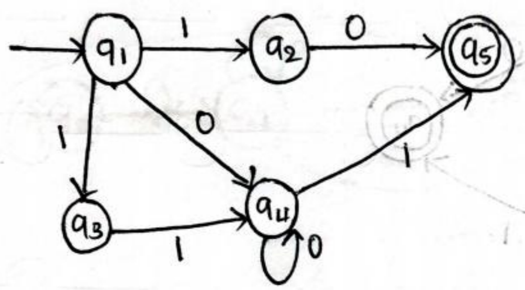


and $R_2 . R_3 =$



$\therefore R_1 + (R_2 . R_3) =$.

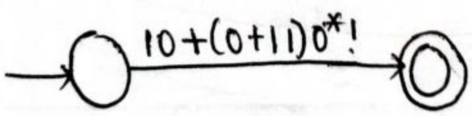
⇓



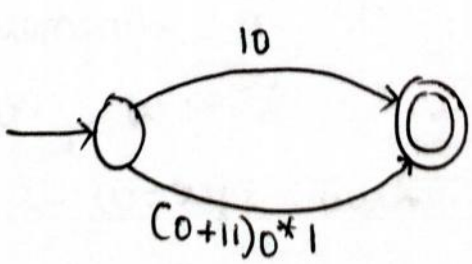
is the required F.A.

M2:-

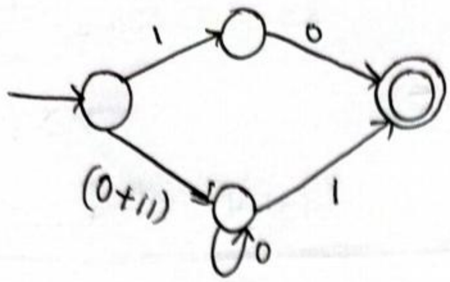
Given,



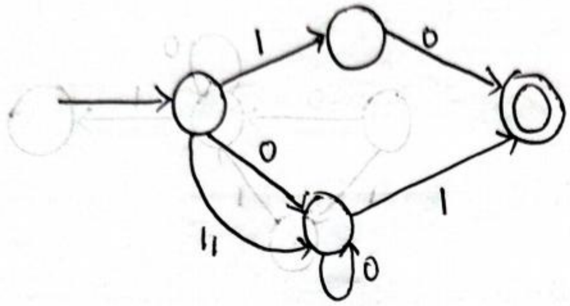
⇓



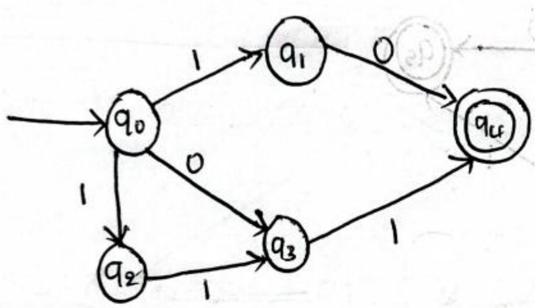
⇓



⇓



⇓

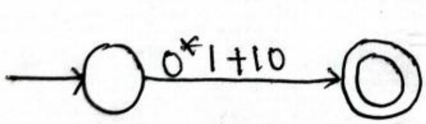


is the required F.A.

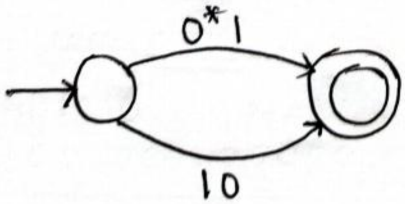
---

ii). Given, 0^*1+10.

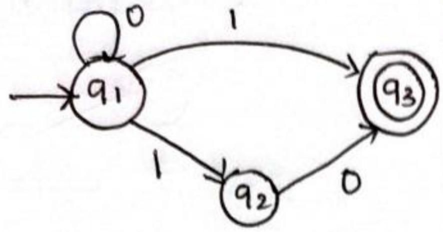
Soln:-



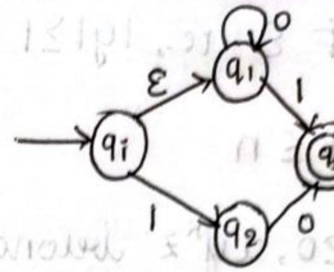
⇓



⇓



(or)

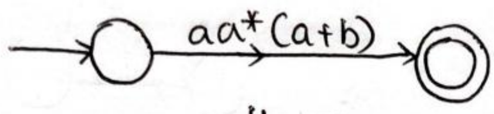


is the required NFA.

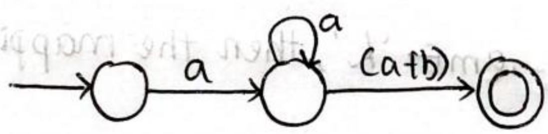
---

12). $aa^*(a+b)$.

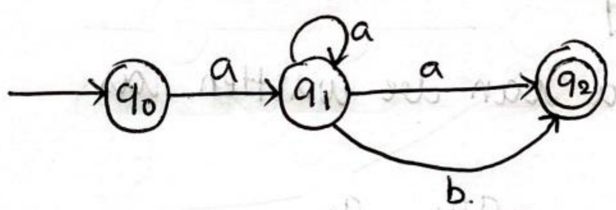
Sol:-



⇓



⇓



is the reqd NFA.

Pumping lemma for Regular Language:-

This lemma is used to prove that a given language is not regular.

Pumping lemma:- Let 'L' be a regular language accepted by finite automata, then there exists a constant 'n' such that every string 'w' $\in L$ where $|w| \geq n$, we can divide it into 3 parts, i.e., $w = xyz$ such that

$\boxed{\begin{array}{l} *)\ |y| \neq \varepsilon \text{ i.e., } |y| \geq 1 \\ *)\ |x.y| \leq n \\ *)\ \forall k \geq 0,\ xy^k z \text{ belongs to } L. \end{array}}$

Proof:- If the language L is regular, then it is accepted by a finite automata.

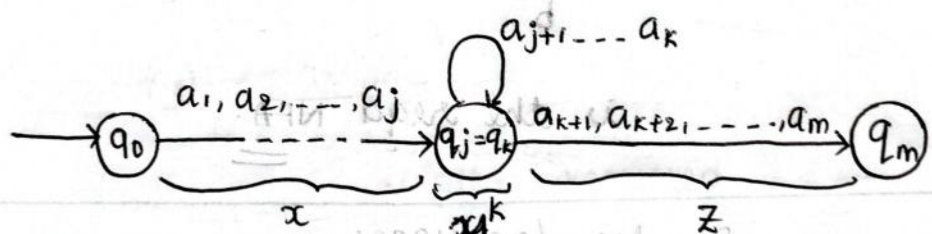
i.e., $M = (Q, \Sigma, q_0, \delta, F)$

with some no of states, lets say 'n'.

Consider an input $a_1, a_2, ---, a_m \in$ 'L', then the mapping function can be written as

$$\boxed{\delta(Q, a_1, a_2, ----, a_m) = Q}$$

Then, the finite automata can be written as



If $q_m \in F$, then $a_1, a_2, ---, a_m \in L(M)$.

$\therefore$ The language is regular.

---

1) Prove that $L = \{a^n : n \text{ is prime}\}$ is not regular.

Proof:- Consider that 'L' is regular.

$$L = \{\overset{n=2}{aa}, \overset{n=3}{aaa}, \overset{n=5}{aaaaa}, ---\}$$

Let us consider, $w = aaaaa$

$\therefore n = 5$

Divide 'w' into xyz

i.e., $w = \underbrace{aa}_{x}\underbrace{a}_{y}\underbrace{aa}_{z}$

here $\boxed{\begin{array}{l} |y| = 1 \geq 1 \\ |x.y| = 3 \leq 5 \end{array}}$

check whether $xy^kz \in L$ for $k=0$

i.e., $xy^0z = xz$

$= aaaa$

but $aaaa \notin L$

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

$\therefore L$ is not regular.

---

imp

2) Prove that $L = \{a^nb^n : n \geq 1\}$ is not regular. 23/10/2024

Proof:- Consider that 'L' is regular.

$L = \{\overset{n=1}{ab}, \overset{n=2}{aabb}, \overset{n=3}{aaabbb}, \ldots\}$

Let us consider, $w = aaabbb \Rightarrow n = 6$.

Divide 'w' into xyz

case1:- i.e., $w = \underbrace{a}_{x}\underbrace{aa}_{y}\underbrace{bbb}_{z}$

{for 6-8 marks, 1 case is enough, for 10-12 marks, write 3 cases}

here $|y| = 2 \geq 1$

$|x.y| = 3 \leq 6$

check whether $xy^kz \in L$ for $k=0$

i.e., $xy^0z = xz = abbb$

$\because abbb \notin L$

$\Rightarrow xy^kz \notin L, \forall k \geq 0.$

$\therefore L$ is not regular.

case 2:- consider $w = \underbrace{aa}_{x}\underbrace{ab}_{y}\underbrace{bb}_{z}.$

here $|y| = 2 \geq 1$

$|x.y| = 4 \leq 6$

and check whether $xy^kz \in L$ for $k \geq 2$.

$xy^kz = xy^2z = x\,y^2\,z$

$= aaababbb$

$\because aaababbb \notin L$

$\Rightarrow xy^kz \notin L$ for $k=2$.

$\therefore L$ is not regular.

case 3:- consider $w = \underbrace{aaa}_{x}\underbrace{b}_{y}\underbrace{bb}_{z}$

here $|y| = 1 \geq 1$

$|x.y| = 4 \leq 6.$

check whether $xy^kz \in L$ for $k=0$.

$xy^kz = xy^0z = xz.$

$= aaabb$

$\because aaabb \notin L$

$\Rightarrow xy^kz \notin L$ for $k=0$

$\therefore L$ is not regular.

3) Prove that $L = \{b^{i^2} : i \geq 1\}$ is not regular.

Proof:- Assume that L is regular.

$L = \{\overset{i=1}{b}, \overset{i=2}{bbbb}, \overset{i=3}{bbbbbbbbb}, \ldots\}$.

Consider $w = bbbb \Rightarrow n = 4$

Divide w into xyz.

ie., $w = \underset{x}{b}\underset{y}{b}\underset{z}{bb}$.

here $|y| = 1 \geq 1$

$|x.y| = 2 \leq 4$.

check whether $xy^kz \in L$ for $k=0$:

$xy^0z = xz$

$= bbb$

$\because bbb \notin L$

$\Rightarrow xy^kz \notin L$ for $k=0$

$\therefore L$ is not regular.

---

4) Prove that $L = \{0^n 1^{n+1} : n \geq 1\}$ is not regular.

Proof:- Assume that L is regular.

$L = \{\overset{n=1}{011}, \overset{n=2}{00111}, \overset{n=3}{0001111}, \ldots\}$.

consider $w = 00111$

$\Rightarrow n = 5$.

divide 'w' into xyz.

Case 1:-

$w = \underset{x}{0}\underset{y}{0}\underset{z}{111}$ here $|y| = 1 \geq 1$

$|x.y| = 2 \leq 5$

check whether $xy^kz \in L$ for $k=0$.

$xy^0z = xz$

$= 0111$

$\because 0111 \notin L$

$\Rightarrow xy^kz \notin L$ for $k=0$

$\therefore L$ is not regular.

<u>case 2:-</u> $w = \underline{0}\,\underline{01}\,\underline{11}$
$\quad\quad x\ \ y\ \ z$.

here $|y| = 2 \geq 1$

$|x.y| = 3 \leq 5$

check whether $xy^kz \in L$ for $k=2$.

$xy^2z = 0010111$

$\because 0010111 \notin L$

$\Rightarrow xy^kz \notin L$, for $k=2$.

$\therefore L$ is not regular.

<u>case 3:-</u> $w = \underline{00}\,\underline{1}\,\underline{11}$
$\quad\quad x\ \ y\ \ z$.

here $|y| = 1 \geq 1$

$|x.y| = 3 \leq 5$

check whether $xy^kz \in L$ for $k=0$

$xy^0z = 0011$

$\because 0011 \notin L$

$\Rightarrow xy^kz \notin L$, for $k=0$

$\therefore L$ is not regular.

5) Prove that $L = \{ww^R : w \in (0+1)^*\}$ is not regular.

Proof:- Assume that L is regular.

$L = \{\varepsilon, 00, 11, 0110, 010010, \ldots\}$.

consider $w = 0100$

$\Rightarrow n = 4$.

divide 'w' into xyz.

$\Rightarrow w = \underline{0}\,\underline{1}\,\underline{10}$
$\quad\quad\; x\;y\;z$.

where $|y| = 1 \geq 1$

$|x.y| = 2 \leq 4$.

check whether $xy^kz \in L$ for $k=0$.

$\Rightarrow xy^0z = xz$.

$= 010$

$\because 010 \notin L$

$\Rightarrow xy^kz \notin L$, for $k=0$.

$\therefore L$ is not regular.

Imp.

Minimization of DFA:-

Reducing the no of states in a given DFA, ~~the new~~ is known as the minimization of DFA.

The new DFA obtained is known as minimized DFA.

Note:- can be done in two methods

i) Equivalence method

ii) Table filling method (not there in syllabus).

Imp.

Distinguishable and indistinguishable states:-

$\boxed{\text{Non-equivalent}}$ $\qquad$ $\boxed{\text{Equivalent}}$

Let us consider two states 'P' and 'Q'. Let 'w' be the string $\in L$

if $\delta(P, w)$ = Final state (or) $\delta(P, w)$ = non-final state
$\delta(Q, w)$ = Final state $\qquad$ $\delta(Q, w)$ = non-final state,

then, 'P' and 'Q' are indistinguishable states.

if $\delta(P, w)$ = Final state (or) $\delta(P, w)$ = non-final state
$\delta(Q, w)$ = non-final state $\qquad$ $\delta(Q, w)$ = final state,

then, 'P' and 'Q' are distinguishable states.

Equivalence:- $\boxed{24/10/2024}$

Length of the String
- 0-equivalence
- 1-equivalence
- 2-equivalence
- ⋮
- n-equivalence

consider

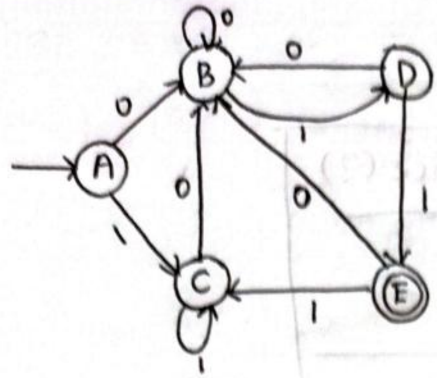
→A			
B			
C			
D			
*E			

To find 0-equivalence, we make two sets
i) {A, B, C, D} → with all non-final states
ii) {E} → with final states (DFA can have multiple final states)

* In order to find 2-equivalence, we need to know 1-equivalence as well as 0-equivalence. So, we need to know all the equivalence value in order to find the 'n'th equivalence.

* the problem terminates when the values of $(n-1)^{th}$ and n^{th} equivalence are same.

1). Given,



minimize this DFA.

Soln:- Construct transition table for given DFA:-

Q	0	1
→A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

0 - equivalence:-

$\underbrace{\{A, B, C, D\}}_{\text{non-final states}}$ $\{E\}$ ↓ final state.

1 - equivalence:-

Q	Transition	Equivalence (?)
(A, B)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,D)$ ✓	✓
(A, C)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,C)$ ✓	✓
(A, D)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,E)$ ✗	✗

Note:-

{ here (B, B) are same, ⇒ no need to check. and (C, D) are diff, hence need to check previous equivalence sets }.

∴ {A, B, C} {D} {E}

2- equivalence:-

Q	Transition	Equivalence (?)
(A,B)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,D)$ ×	×
(A,C)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,C)$ ✓	✓

$\therefore$ {A,C} {B} {D} {E}

3- equivalence:-

Q	Transition	Equivalence (?)
(A,C)	$\delta(Q,0) = (B,B)$ ✓ $\delta(Q,1) = (C,C)$ ✓	✓

$\therefore$ {A,C} {B} {D} {E}

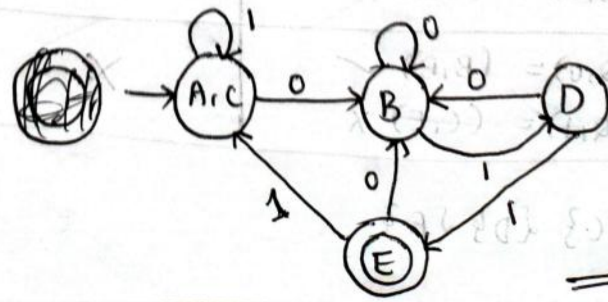
$\because$ 3-equivalence = 2-equivalence $\Rightarrow$ (stop).

$\Rightarrow$ the DFA is minimized.

Transition table for minimized DFA:-

Q	0	1
→ (A,C)	B	(A,C)
B	B	D
D	B	E
*E	B	(A,C)

required minimized DFA is



2) ~~Minimize the DFA~~

given,

	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

Soln:-

0- equivalence:-

$\{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$ $\{q_2\}$

1- equivalence:-

Q	Transition	Equivalence (?)
(q_0, q_1)	$\delta(Q,0) = (q_1, q_6)$ ✓ $\delta(Q,1) = (q_5, q_2)$ ✗	✗
(q_0, q_3)	$\delta(Q,0) = (q_1, q_2)$ ✗ $\delta(Q,1) = (q_5, q_6)$ ✓	✗
(q_0, q_4)	$\delta(Q,0) = (q_1, q_7)$ ✓ $\delta(Q,1) = (q_5, q_5)$ ✓	✓
(q_0, q_5)	$\delta(Q,0) = (q_1, q_2)$ ✗ $\delta(Q,1) = (q_5, q_6)$ ✓	✗
(q_0, q_6)	$\delta(Q,0) = (q_1, q_6)$ ✓ $\delta(Q,1) = (q_5, q_4)$ ✓	✓
(q_0, q_7)	$\delta(Q,0) = (q_1, q_6)$ ✓ $\delta(Q,1) = (q_5, q_2)$ ✗	✗

∴ ~~$\{q_0, q_4, q_6\}$ $\{q_1,$~~

(q_1, q_3)	$\delta(Q,0) = (q_6, q_2)$ ✗ $\delta(Q,1) = (q_2, q_6)$ ✗	✗
(q_1, q_5)	$\delta(Q,0) = (q_6, q_2)$ ✗ $\delta(Q,1) = (q_2, q_6)$ ✗	✗
(q_1, q_7)	$\delta(Q,0) = (q_6, q_6)$ ✓ $\delta(Q,1) = (q_2, q_2)$ ✓	✓
(q_3, q_5)	$\delta(Q,0) = (q_2, q_2)$ ✓ $\delta(Q,1) = (q_6, q_6)$ ✓	✓

∴ $\{q_0, q_4, q_6\}$ $\{q_1, q_7\}$ $\{q_3, q_5\}$ $\{q_2\}$.

2 - equivalence :-

Q	Transition	Equivalence (?)
(q_0, q_4)	$\delta(Q,0) = (q_1, q_7)$ ✓ $(Q,1) = (q_5, q_5)$ ✓	✓
(q_0, q_6)	$\delta(Q,0) = (q_3, q_6)$ ✗ $\delta(Q,1) = (q_5, q_4)$ ✗	✗
(q_1, q_7)	$\delta(Q,0) = (q_6, q_6)$ ✓ $\delta(Q,1) = (q_2, q_2)$ ✓	✓
(q_3, q_5)	$\delta(Q,0) = (q_2, q_2)$ ✓ $\delta(Q,1) = (q_6, q_6)$ ✓	✓

∴ $\{q_0, q_4\}$ $\{q_1, q_7\}$ $\{q_3, q_5\}$ $\{q_2\}$ $\{q_6\}$

3- equivalence:-

Q	Transition	Equivalence (?)
(q_0, q_4)	$\delta(Q,0) = (q_1, q_7)$ ✓ $\delta(Q,1) = (q_3, q_5)$ ✓	✓
(q_1, q_7)	$\delta(Q,0) = (q_6, q_6)$ ✓ $\delta(Q,1) = (q_2, q_2)$ ✓	✓
(q_3, q_5)	$\delta(Q,0) = (q_2, q_2)$ ✓ $\delta(Q,1) = (q_6, q_6)$ ✓	✓

$\therefore \{q_0, q_4\}\ \{q_1, q_7\}\ \{q_3, q_5\}\ \{q_2\}\ \{q_6\}$

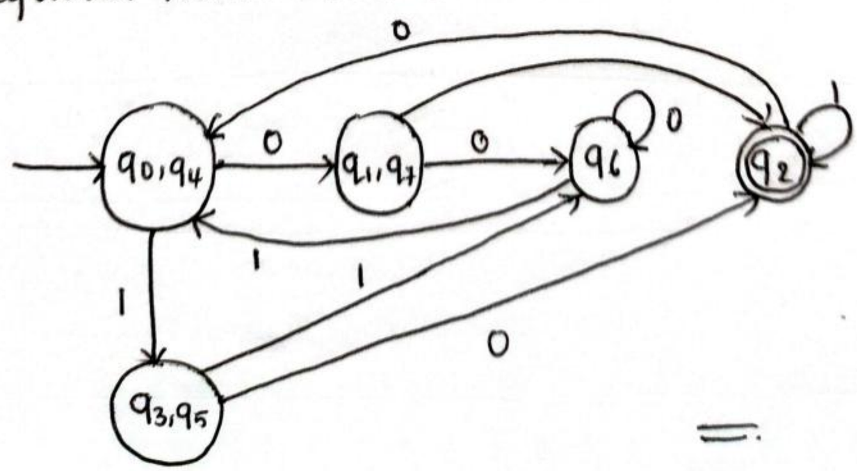
$\because$ 2-equivalence = 3-equivalence

$\Rightarrow$ DFA is minimized.

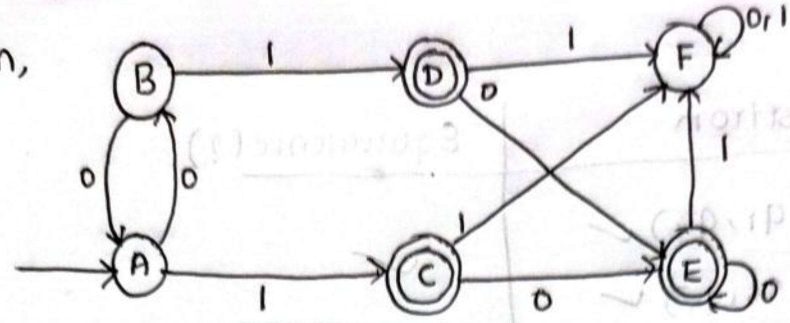
Transition table for minimized DFA:-

	0	1
$\rightarrow (q_0, q_4)$	(q_1, q_7)	(q_3, q_5)
(q_1, q_7)	q_6	q_2
(q_3, q_5)	q_2	q_6
$*\ (q_2)$	(q_0, q_4)	q_2
(q_6)	q_6	(q_0, q_4)

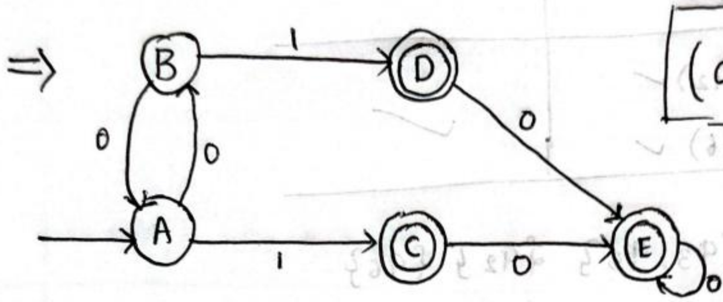
required minimized DFA is



3) Given,



Soln :- ∵ F is a dead state ⇒ ~~eliminate it~~.

⇒ 

(dont remove the dead state)

(only remove unreachable state)

Transition table for DFA :-

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

0- equivalence :-

{A, B} {C, D, E}

1- equivalence :-

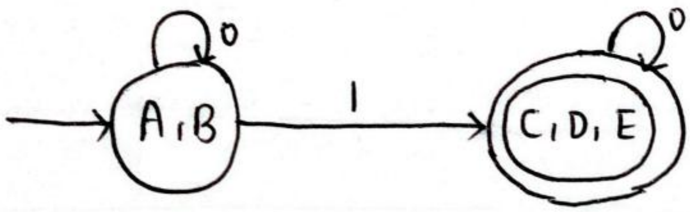
Q	Transition	Equivalence (?)
(A, B)	$\delta(Q, 0) = (B, A)$ ✓ $\delta(Q, 1) = (C, D)$ ✓	✓
(C, D)	$\delta(Q, 0) = (E, E)$ ✓ $\delta(Q, 1) = (\phi, \phi)$ ✓	✓
(C, E)	$\delta(Q, 0) = (E, E)$ ✓ $\delta(Q, 1) = (\phi, \phi)$ ✓	✓

$\therefore$ {A, B} {C, D, E}

Transition table for minimized D.F.A:

	0	1
→(A, B)	{A, B}	{C, D, E}
* (C, D, E)	{C, D, E}	ϕ

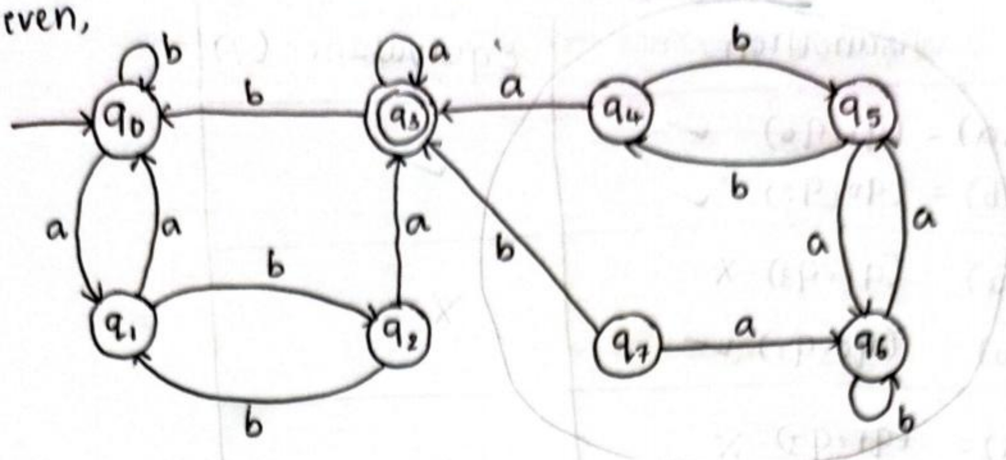
required minimized DFA is



=.

4) Given,

24/10/2024



Soln:- Transition table for given DFA:- <can remove q_7, q_4, q_5, q
as it is unreachable>

	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
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

0- equivalence:-

$\{q_0, q_1, q_2, q_4, q_5, q_6, q_7\}$

$\{q_3\}$.

1- equivalence:-

Q	Transition	Equivalance (?)
(q_0, q_1)	$\delta(Q,a) = (q_1, q_0)$ ✓ $\delta(Q,b) = (q_0, q_2)$ ✓	✓
(q_0, q_2)	$\delta(Q,a) = (q_1, q_3)$ ✗ $\delta(Q,b) = (q_0, q_1)$ ✓	✗
(q_0, q_4)	$\delta(Q,a) = (q_1, q_3)$ ✗ $\delta(Q,b) = (q_0, q_5)$ ✓	✗
(q_0, q_5)	$\delta(Q,a) = (q_1, q_6)$ ✓ $\delta(Q,b) = (q_0, q_4)$ ✓	✓
(q_0, q_6)	$\delta(Q,a) = (q_1, q_5)$ ✓ $\delta(Q,b) = (q_0, q_6)$ ✓	✓
(q_0, q_7)	$\delta(Q,a) = (q_1, q_6)$ ✓ $\delta(Q,b) = (q_0, q_3)$ ✗	✗
(q_2, q_4)	$\delta(Q,a) = (q_3, q_3)$ ✓ $\delta(Q,b) = (q_1, q_5)$ ✓	✓
(q_2, q_7)	$\delta(Q,a) = (q_3, q_6)$ ✗ $\delta(Q,b) = (q_1, q_3)$ ✗	✗

$\therefore \{q_0, q_1, q_5, q_6\} \quad \{q_2, q_4\} \quad \{q_7\} \quad \{q_3\}$.

2-equivalence:-

Q	Transition	Equivalence (?)
(q_0, q_1)	$\delta(Q, a) = (q_1, q_0)$ ✓ $\delta(Q, b) = (q_0, q_2)$ ×	×
(q_0, q_5)	$\delta(Q, a) = (q_1, q_6)$ ✓ $\delta(Q, b) = (q_0, q_4)$ ×	×
(q_0, q_6)	$\delta(Q, a) = (q_1, q_5)$ ✓ $\delta(Q, b) = (q_0, q_6)$ ✓	✓
(q_1, q_5)	$\delta(Q, a) = (q_0, q_6)$ ✓ $\delta(Q, b) = (q_2, q_4)$ ✓	✓
(q_2, q_4)	$\delta(Q, a) = (q_3, q_3)$ ✓ $\delta(Q, b) = (q_1, q_5)$ ✓	✓

$\therefore \{q_0, q_6\}$ $\{q_1, q_5\}$ $\{q_2, q_4\}$ $\{q_7\}$ $\{q_3\}$.

3- equivalence:-

Q	Transition	Equivalence (?)
(q_0, q_6)	$\delta(Q, a) = (q_1, q_5)$ ✓ $\delta(Q, b) = (q_0, q_6)$ ✓	✓
(q_1, q_5)	$\delta(Q, a) = (q_0, q_6)$ ✓ $\delta(Q, b) = (q_2, q_4)$ ✓	✓
(q_2, q_4)	$\delta(Q, a) = (q_3, q_3)$ ✓ $\delta(Q, b) = (q_1, q_5)$ ✓	✓

$\therefore$ ~~$\{q_0, q_6\}$~~ $\{q_0, q_6\}$ $\{q_1, q_5\}$ $\{q_2, q_4\}$ $\{q_7\}$ $\{q_3\}$.

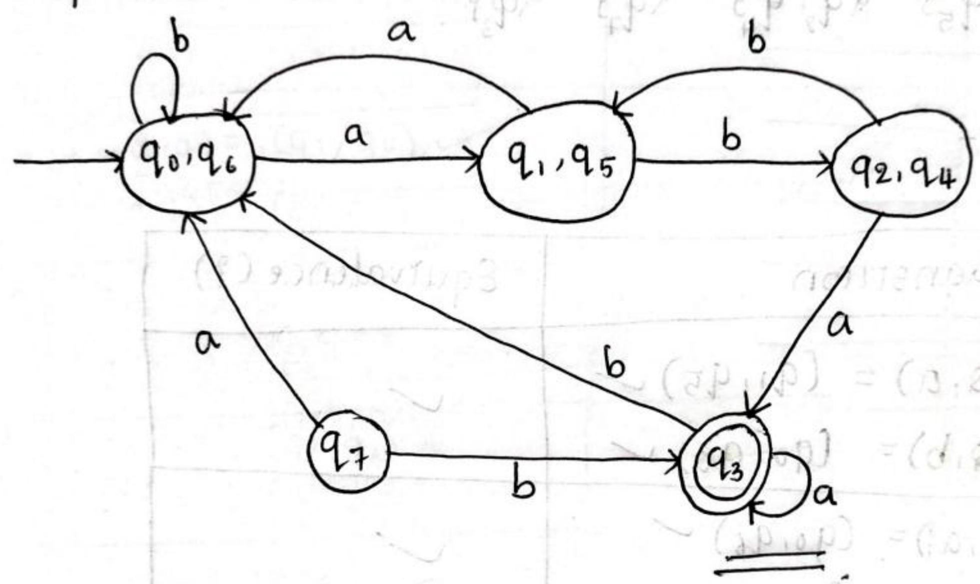
$\because$ 2-equivalence = 3-equivalence

$\Rightarrow$ DFA is minimized.

Transition table for minimized DFA :-

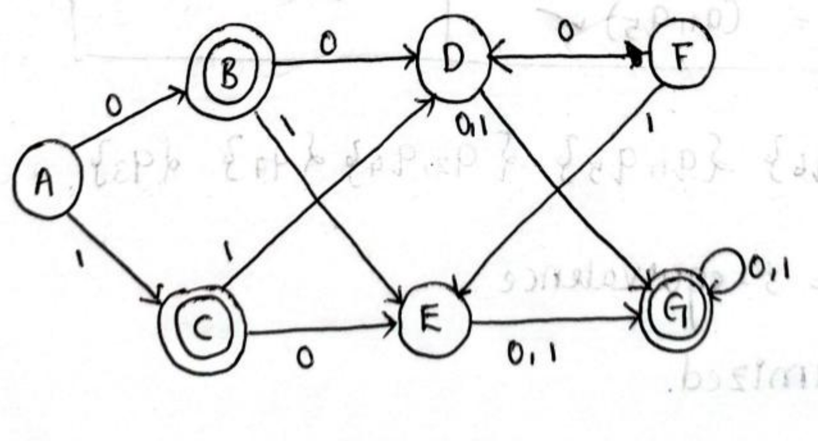
	a	b
→(q_0, q_6)	(q_1, q_5)	(q_0, q_6)
(q_1, q_5)	(q_0, q_6)	(q_2, q_4)
(q_2, q_4)	(q_3)	(q_1, q_5)
q_7	(q_0, q_6)	(q_3)
* q_3	q_3	(q_0, q_6)

∴ required minimized DFA is



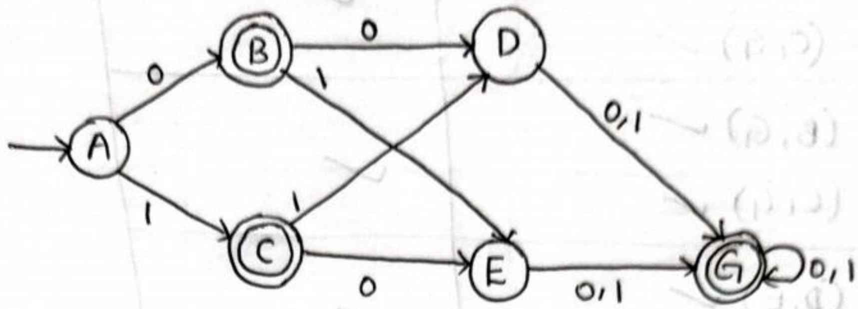
25/10/2024

5).



Soln:- Here, 'F' is unreachable state.

∴ Remove F and consider modified DFA.



Transition table for above DFA :-

	0	1
→ A	B	C
*B	D	E
*C	E	D
D	G	G
E	G	G
*G	G	G.

0- equivalence:-

{A, D, E}   {B, C, G}.

1- equivalence:-

Q	Transition	Equivalence (?)
(A,D)	$\delta(Q,0) = (B,G)$ ✓ $\delta(Q,1) = (C,G)$ ✓	✓
(A,E)	$\delta(Q,0) = (B,G)$ ✓ $\delta(Q,1) = (C,G)$ ✓	✓
(B,C)	$\delta(Q,0) = (D,E)$ ✓ $\delta(Q,1) = (E,D)$ ✓	✓
(B,G)	$\delta(Q,0) = (D,G)$ ✗ $\delta(Q,1) = (E,G)$ ✗	✗

$\therefore \{A, D, E\} \{B, C\} \{G\}$

2-equivalence:-

Q	Transition	Equivalence (?)
(A,D)	$\delta(Q,0) = (B,G)$ ✗ $\delta(Q,0) = (G,G)$ ✗	✗
(A,E)	$\delta(Q,0) = (B,G)$ ✗ $\delta(Q,1) = (C,G)$ ✗	✗
(D,E)	$\delta(Q,0) = (G,G)$ ✓ $\delta(Q,1) = (G,G)$ ✓	✓
(B,C)	$\delta(Q,0) = (D,E)$ ✓ $\delta(Q,1) = (E,D)$ ✓	✓

$\therefore \{D, E\} \{B, C\} \{A\} \{G\}$

3- equivalence :-

Q	Transition	Equivalence (?)
(D, E)	$\delta(Q,0) = (G, G)$ ✓ $\delta(Q,1) = (G, G)$ ✓	✓
(B, C)	$\delta(Q,0) = (D, E)$ ✓ $\delta(Q,1) = (E, D)$ ✓	✓
	ℓ.	

∴ {D, E} {B, C} {A} {G}.

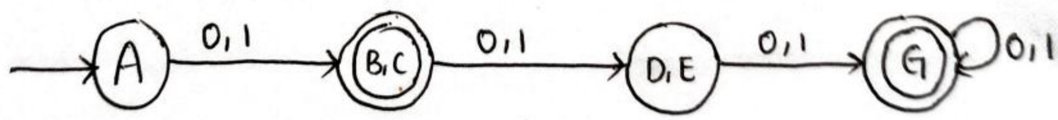
∵ 2-equivalence = 3-equivalence

⇒ DFA is minimized.

Transition table for minimized DFA :-

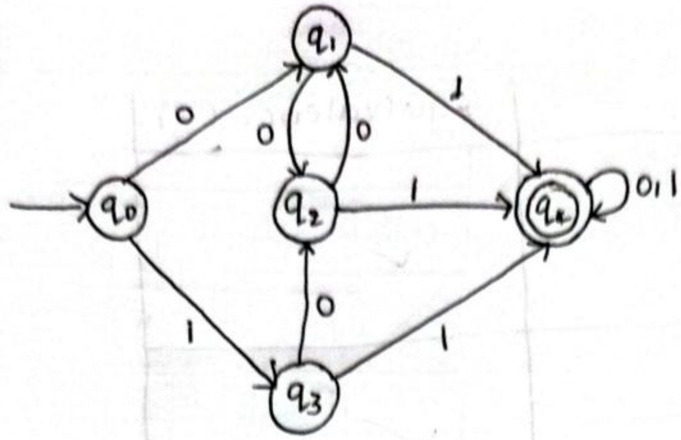
	0	1
→A	(B, C)	(B, C)
*(B, C)	(D, E)	(D, E)
(D, E)	G	G
*G	G	G

required minimized DFA is



=.

6)



Soln:- Transition table for DFA:-

	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- equivalence:-

$\{q_0, q_1, q_2, q_3\}$ $\{q_4\}$

1- equivalence:-

Q	Transition	Equivalence (?)
(q_0,q_1)	$\delta(Q,0) = (q_1,q_2)$ ✓ $\delta(Q,1) = (q_3,q_4)$ ×	×
(q_0,q_2)	$\delta(Q,0) = (q_2,q_1)$ ✓ $\delta(Q,1) = (q_3,q_4)$ ×	×
(q_0,q_3)	$\delta(Q,0) = (q_1,q_2)$ ✓ $\delta(Q,1) = (q_3,q_4)$ ×	×
(q_1,q_2)	$\delta(Q,0) = (q_2,q_1)$ ✓ $\delta(Q,1) = (q_4,q_4)$ ✓	✓
(q_1,q_3)	$\delta(Q,0) = (q_2,q_2)$ ✓ $\delta(Q,1) = (q_4,q_4)$ ✓	✓

$\therefore \{q_1,q_2,q_3\}\{q_0\}\{q_4\}$.

2-equivalence:-

Q	Transition	Equivalence (?)
(q_1,q_2)	$\delta(Q,0)=(q_2,q_1)$ ✓ $\delta(Q,0)=(q_4,q_4)$ ✓	✓
(q_1,q_3)	$\delta(Q,0)=(q_2,q_2)$ ✓ $\delta(Q,1)=(q_4,q_4)$ ✓	✓

$\therefore \{q_1,q_2,q_3\}\{q_0\}\{q_4\}$

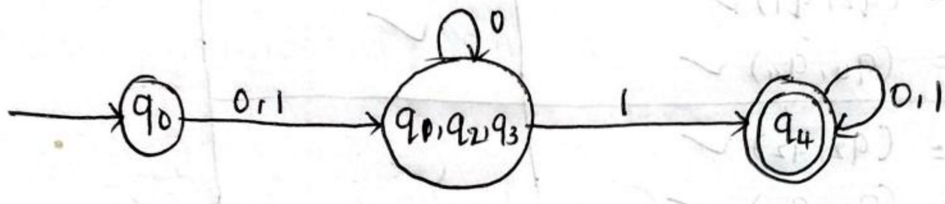
$\because$ 1-equivalence = 2-equivalence

$\Rightarrow$ DFA is minimized

Transition table for minimized DFA:-

	0	1
$\rightarrow q_0$	(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

required minimized DFA is



v.imp.

7). Given,

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

Soln:- Here E, F, G, H are unreachable states, so eliminate them.

$\therefore$ ~~modif~~ Modified DFA transition table is

	0	1
→A	B	A
B	A	C
C	D	B
*D	D	A

0- equivalence:-

{A, B, C} {D}

1- equivalence:-

Q	Transition	Equivalence (?)
(A, B)	$\delta(Q,0) = (B,A)$ ✓ $\delta(Q,1) = (A,C)$ ✓	✓
(A, C)	$\delta(Q,0) = (B,D)$ ✗ $\delta(Q,1) = (A,B)$ ✓	✗

∴ {A, B} {C} {D}

2- equivalence:-

Q	Transition	Equivalence (?)
(A, B)	$\delta(Q,0) = (B,A)$ ✓ $(Q,1) = (A,C)$ ✗	✗

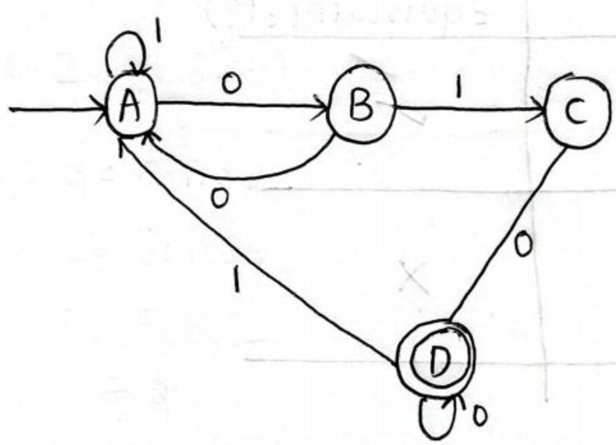
∴ {A} {B} {C} {D}

∵ no two states are merged, we can say that the given DFA is minimized.

Transition table for minimized DFA

	0	1
→A	B	A
B	A	C
C	D	B
*D	D	A

required minimized DFA is



Closure Properties of Regular language:-

*) Closure properties on regular languages are defined as certain operations on regular language that are guaranteed to produce regular language.

*) We say that regular language is closed under these operations.

*) The different operations include:

1) Closure of regular languages under Boolean operations:

i.e., *) Union

*) Intersection and

*) Complement.

2) Difference of two regular languages is regular.

3) The reversal of a regular language is regular.

4) The closure (star) of a regular language is regular.

5) The concatenation of regular languages is regular.

6) Homomorphism of a regular language is regular.

7) The inverse homomorphism of a regular language is regular.

Closure of regular languages under Boolean operations:-

1) Regular languages are closed under union operation.

Proof:- Let 'L' and 'M' be two regular languages over alphabet set 'Σ'.

∵ both 'L' and 'M' are regular, they have regular expressions, say

$L = L(R) \longrightarrow (1)$

$M = L(S) \longrightarrow (2)$

Then from definition of regular expression,

$L \cup M = L(R) + L(S) = \boxed{L(R+S) \text{ is regular}}$.

∴ Regular languages are closed under union operation.

2) Regular languages are closed under <u>intersection</u> operations.

<u>Proof</u>:- Let there be two regular languages L_1 and L_2.

This means there exists two DFA's M_1 and M_2 such that,

$M_1 = (Q_1, \Sigma, q_1, \delta_1, F_1)$ and

$M_2 = (Q_2, \Sigma, q_2, \delta_2, F_2)$.

Let 'L' be the language obtained by $L_1 \cap L_2$, which is accepted by

$M = (Q, \Sigma, q, \delta, F)$

where, $Q = Q_1 \cap Q_2$

$\delta = \delta_1 \cap \delta_2$

$F = F_1 \cap F_2$

$q \in Q$

From this it is clear that there exists a DFA which accepts $L_1 \cap L_2$

$\boxed{\therefore\ L_1 \cap L_2 \text{ is regular}}$

ie., Regular languages are closed under intersection operation.

3) Regular languages are closed under complement operation.

Proof:- Let us consider that L is a regular language.

Then this language is accepted by DFA,

$M = (Q, \Sigma, \delta, q, F)$.

Let $\bar{L}$ be the complement of 'L'

If we consider $\bar{M}$ as the machine then,

$$\bar{M} = (Q, \Sigma, \delta, q, Q-F)$$

ie., for this machine, the final state(s) will be the non-final state(s) of M.

However, there exists a DFA, $\bar{M}$.

$\therefore \bar{L}$ will have a DFA that accepts $\bar{L}$.

$\therefore \bar{L}$ is regular.

ie., Regular languages are closed under complement operation.

(*) Difference of two regular language is Regular.

Proof:- If 'L_1' and 'L_2' are two regular languages, then difference is represented as '$L_1 - L_2$'

Mathematically,

$$\boxed{L_1 - L_2 \Longleftrightarrow L_1 \cap \overline{L_2}}$$

Let 'L_1' be accepted by $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and 'L_2' be accepted by $M = (Q_2, \Sigma, \delta_2, q_2, F_2)$

W.K.T

$\overline{L_2}$ is also regular accepted by

$M = (Q_2, \Sigma, \delta_2, q_2, Q_2 - F_2)$

$\because L_1$ and $\overline{L_2}$ are regular,

$\Rightarrow L_1 \cap \overline{L_2}$ is regular

~~where R_1: Regular expression $L(R_1)$~~.

~~$\overline{R_2}$~~:

$\therefore L_1 - L_2$ is regular.

ie., Regular language is closed under difference operation.

Note:- Let $w = a_1 a_2 \ldots a_n$ be a string, then

$w^R = a_n a_{n-1} \ldots a_1$

$\forall L \subseteq \Sigma^*$

$L^R = \{ w \in \Sigma^* : w^R \in L \}$

(*) **Reversal** of regular language is regular

Proof:- Let us consider 'L' is regular i.e., 'L' is accepted by DFA

i.e., $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$.

Let M^R is machine

i.e., $M^R = (Q_2, \Sigma, \delta_2, q_2, F_2)$

where $F_2 = q_1 \in M$

$q_2 = F_1 \in M$.

δ_2 contains edges in reverse direction to the one's in $\delta_1 \in M$.

From this, it is clear that there exists a DFA M^R.

$\therefore L^R$ is regular

i.e., Regular language is closed under reverse operation.

⊛ The closure of regular language is regular.

Proof:- Let 'L' be a regular language.

ie., 'L' is accepted by DFA.

$M = (Q, \Sigma, \delta, q_0, F)$

If 'R' is the regular expression represented by 'L'

ie., $L = L(R)$ then,

by defenition of regular expression,

R^* is regular

ie., there exists a language such that

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

∴ Regular language is closed under closure operation.

⊛ The concatenation of regular language is regular.

Proof:- Let 'L' and 'M' be two regular languages over alphabet set 'Σ'.

∵ both 'L' and 'M' are regular languages, then they have regular expressions.

$L = L(R) \longrightarrow ①$

$M = L(S) \longrightarrow ②$

Then from defenition of regular expression,

$L.M = L(R).L(S) = \boxed{L(R.S) \text{ is regular}}$

∴ Regular languages are closed under concatenation operation.

Homomorphism:-

Homomorphism means substitution of a string by some other symbols.

i.e., if $w = aabb$, then it can be written as

$w = 0011$

i.e., here $a \longrightarrow 0$

$b \longrightarrow 1$

If 'Σ' is the set of input alphabets and 'T' be the set of substitution symbols, then

$\Sigma^* \longrightarrow T^*$ is homomorphism.

if $w = a_1 a_2 a_3 \ldots a_n$, then

$$\boxed{h(w) = h(a_1)\,h(a_2) \ldots h(a_n)}$$

⊛ Homomorphism of a regular language is regular.

Proof:- Let 'L' be a regular language such that $L \rightarrow L(R)$

∵ 'R' is a regular expression with symbols in Σ,

let $h(R)$ be the expression obtained by replacing

each symbol, a, of Σ in R by $h(a)$.

Let $H(R)$ defines the language $H(L)$

Let us consider a case of union operation,

i.e., let $R = R_1 + R_2$

$$H(R) = H(R_1 + R_2)$$

$H(R) = H(R_1) + H(R_2)$ is a regular expression that

defines.

$$L(H(R)) = L(H(R_1 + R_2))$$
$$= L(H(R_1)) + L(H(R_2))$$

This show that for regular expression $H(R)$, there exists

a language $L(H(R))$

∴ Regular language are closed under homomorphism

* Inverse Homomorphisim of a regular language is regular.

Proof:- Let $\Sigma^* \longrightarrow T^*$ is homomorphism.

ie., Σ is a input alphabet.

T is replacement symbols used by homomorphic function.

If 'L' is a regular language, W.K.T, $H(L)$ is a homomorphic language & is regular.

Let $M = (Q, \Sigma, \delta, q_0, F)$ accepts 'L' and hence it accepts $H(L)$.

For complement of L ie., L', the inverse homomorphic language is $\boxed{H^{-1}(L)}$.

W.K.T there exists a DFA for L' and hence there exists a DFA for $H^{-1}(L)$

ie., $\boxed{H^{-1}(L) \text{ is regular}}$

ie., Regular language is closed under inverse homomorphic operation.