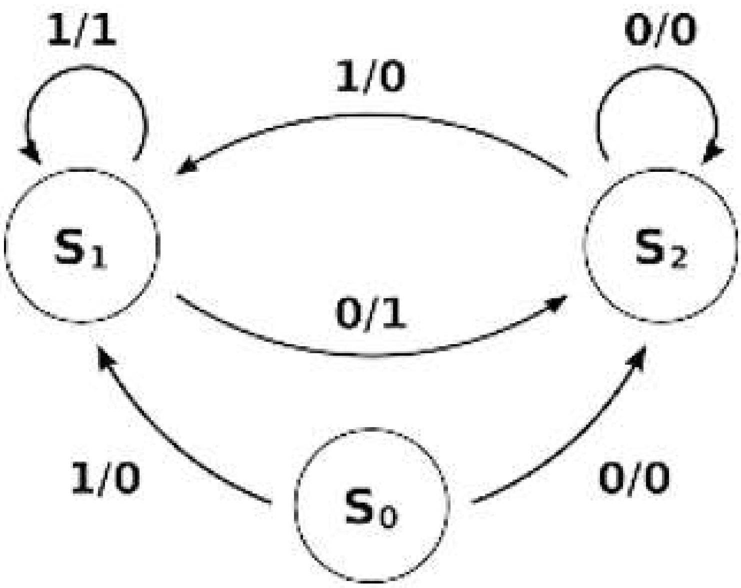
GRADAMIC

B.Tech-2nd Year
Session 2023-24
Even Semester



Dr. A. P. J. Abdul Kalam Technical University
Lucknow, Uttar Pradesh

THEORY OF AUTOMATA AND FORMAL LANGUAGES

(BCS 402)

app.gradamic.com

UNIT - 2nd

Automata

1) Regular and non-Regular expression

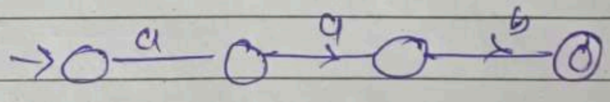
Regular language :- A language that can be represented by finite automata machine are called regular languages.

The regular languages are those languages ~~that~~ whom regular expression can be made.

Eg :-

$L = \{aab\}$.

$\hookrightarrow RE = aab$



Non-regular languages :- Non regular languages are those type of languages which cannot be represented by any finite automata.

The classic example of nregular languages are :-

$L = \{a^m d^m\}$ $m \geq 1$

Page No.
Date : / /

The following example is a non-regular language b/c there is no - counter that counts the exact number of "a" and "d"

Regular expression :- A regular expression is a way of representing every string of a regular language.

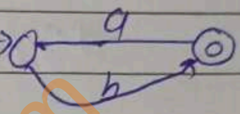
Operations on regular expression

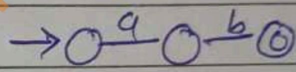
+ → or
. → Concatination
* → Closure.

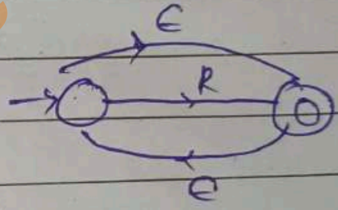
Laws of Regular expression

1) $RE = \phi \rightarrow L = \{\phi\} \rightarrow \bigcirc$

2) $RE = \epsilon \rightarrow L = \{\epsilon\} \rightarrow \circledcirc$

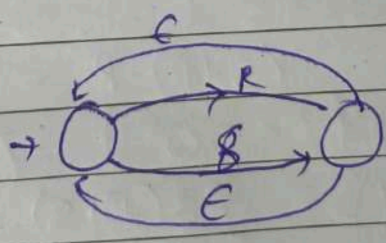
3) $RE = a+b \rightarrow L = \{a, b\} \rightarrow$ 

4) $RE = ab \rightarrow L = \{ab\} = \rightarrow$ 

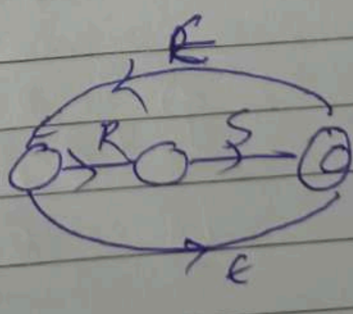
5) $R^* = \epsilon, R^1 R^2 R^3 =$ 

6) $RE = (R+S)^*$

↳ L = {R, and S combo}

↳ 

7) $RE = (RS)^*$

↳ 

Page No.
Date : / /

a $R+S = S+R$

a2 but $RS \neq SR$

b $R\varepsilon = \varepsilon R = R$

c $R \cdot \phi = \phi = \phi \cdot R$

d) $R \cdot R^* = R^+$

$\hookrightarrow R \cdot [R^0 + R^1 + R^2 + R^3 + \text{---}]$

$\hookrightarrow R \cdot \epsilon + R^1 + R^2 \text{---}$

$R^1 + R^2 + R^3$

$\underline{\underline{R^+}}$

e) $R(S+P) = (R \cdot S) + (R \cdot P)$

$(S+P) \cdot R = \underline{SR + PR}$

f) $R+(P+S) = (R+P)+S$.

g) $R \cdot (P \cdot S) = (R \cdot P) \cdot S$.

h) $\varepsilon^* = \varepsilon^0, \varepsilon^1 \text{---}$

$= \epsilon$

$\phi^0 = \epsilon$

i) $\phi^* = \phi^0, \phi^1, \phi^2$

$= \epsilon, \phi^1, \phi^2 \text{---}$

$= \underline{\underline{\epsilon}}$

Page No.
Date : / /

Q Simplyfy.

$$R^* \cdot R + R.$$

from property :- $R^* R = R^+$

$$R^+ + R$$
$$= R^+$$

Q $x \cdot (\underbrace{x^* \cdot x}_{x^+} + x^*) + x^*$

$$x(\underbrace{x^+ + x^*}_{x^*}) + x^*$$

$$x(x^+) + x^*$$

$$x^*$$

Q Simplyfy the expression :-

$$(x+y)^* + x \cdot y + yx)^*$$

$L = \{e, x, xx, xxx ---, y, yyy ---$
$xy, yx, xxy, yyx ---\}$

any combination of x and y.

$$L = \{(x+y)^*\}$$

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

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

Soln from LHS.

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

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

$$(1+00^*1)[\underbrace{\epsilon}_{\epsilon}+\underbrace{(0+10^*1)^*}_{R^*}\underbrace{(0+10^*1)}_{R}]$$

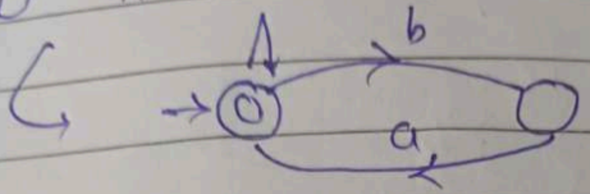
$$\underbrace{(1+00^*1)}_{\text{take } 1 \text{ common}}(0+10^*1)^*$$

$$1(\epsilon+\underbrace{0}_{R}\underbrace{0^*}_{R^*})(0+10^*1)^*$$

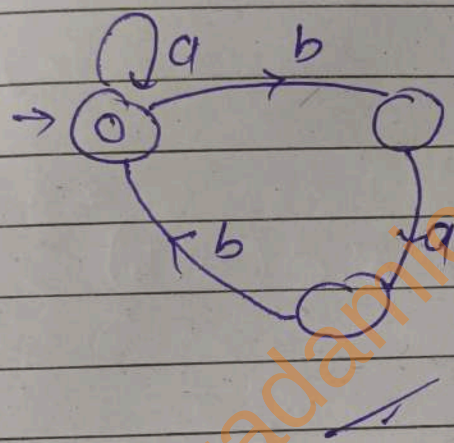
$$(00^*1)(0+10^*1)^*$$

RE to FA

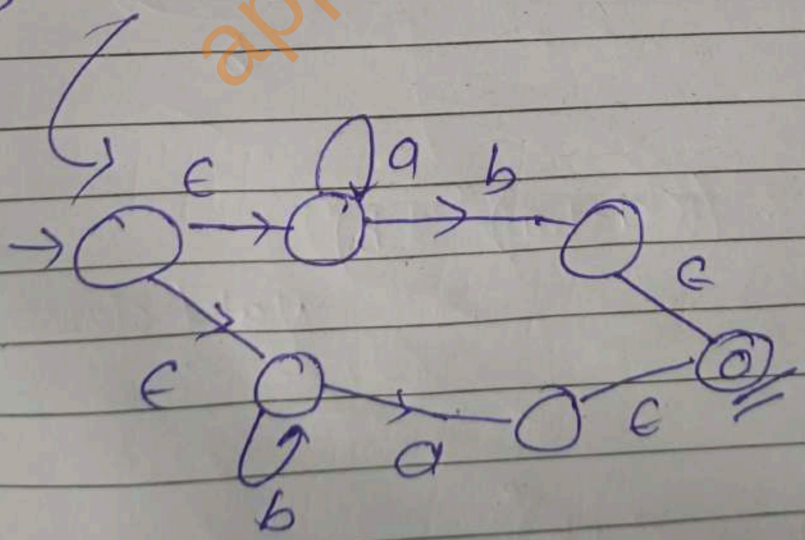
① $RE = (a+ba)^*$



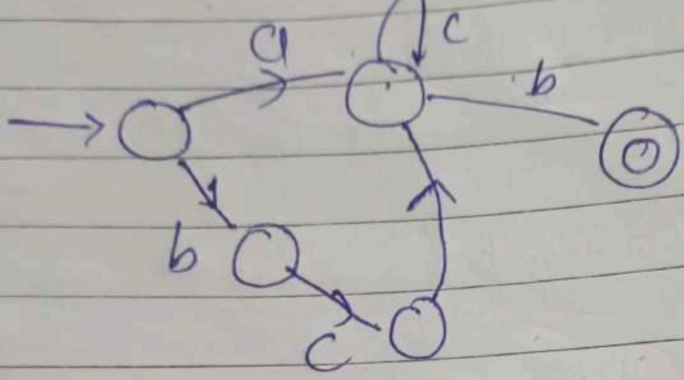
② $RE = (a+bab)^*$



③ $RE = a^* \cdot b + b^* \cdot a$

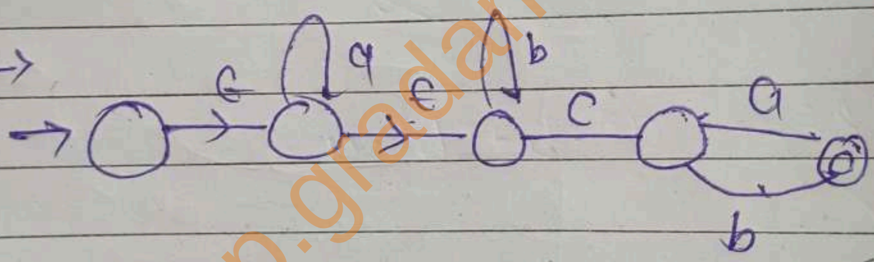


(4) $(a+bc)c^* \cdot b$.



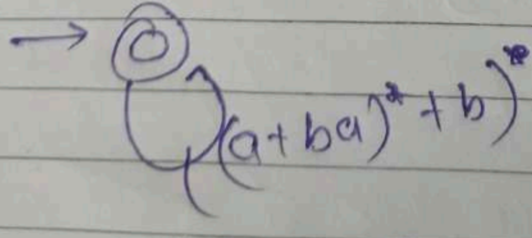
(10/2) = (5) $a^* b^* . c\,(a+b)$

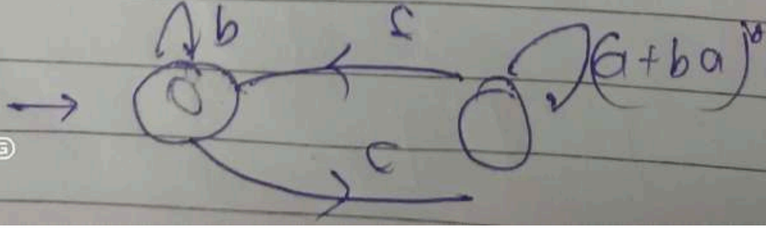
C→

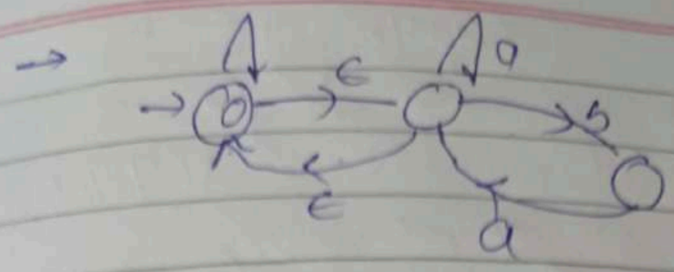


(6) $RE = ((a+ba)^* + b)^*$

Nested closure

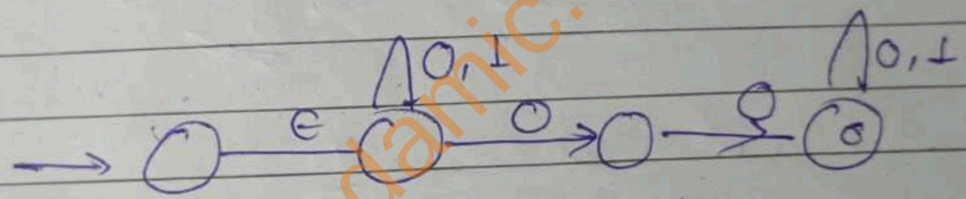


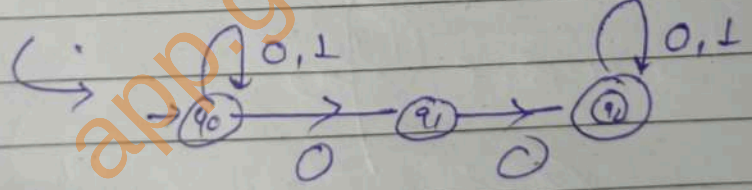




RE to NFA and DFA

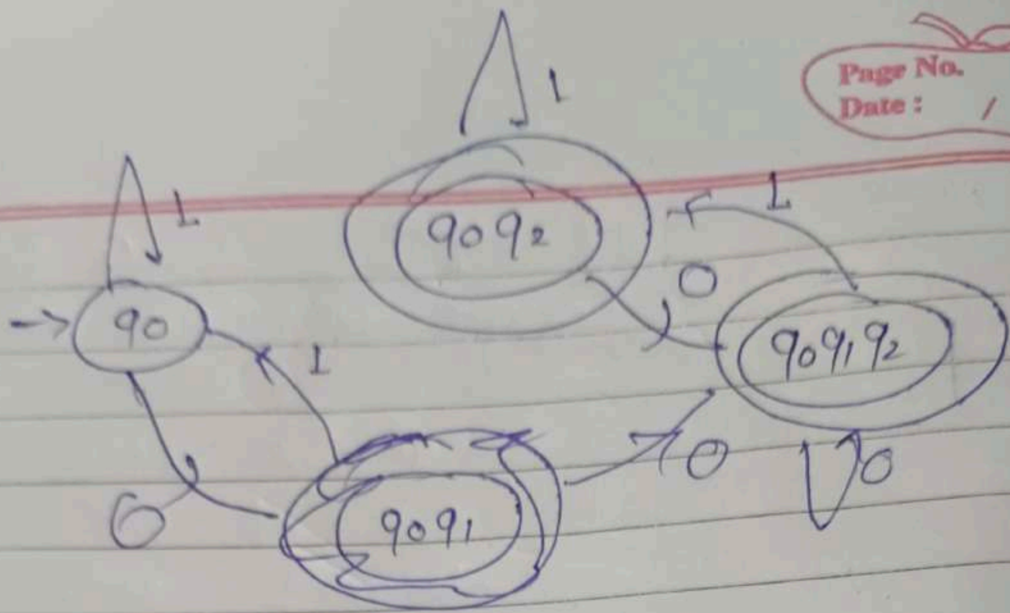
$L = \{ (0+1)^* 0 \cdot 0 \, (0+1)^* \}$





	0	1
→ q_0	q_0q_1	q_0
q_1	q_2	X
* q_2	q_2	q_2

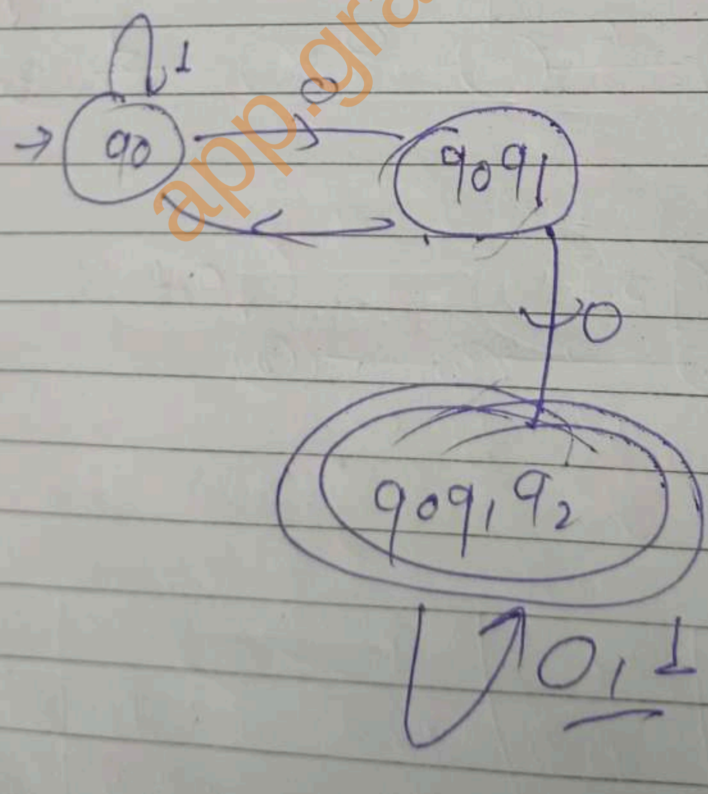
	0	1
→ q_0	q_0q_1	q_0
* q_0q_1	$q_0q_1q_2$	q_0
* $q_0q_1q_2$	$q_0q_1q_2$	q_0q_2
* q_0q_2	$q_0q_1q_2$	q_0q_2



Minimize.

[q_0 q_0q_1] [q_0q_2, $q_0q_1q_2$]

[q_0] [q_0q_1] [q_0q_2 $q_0q_1q_2$]

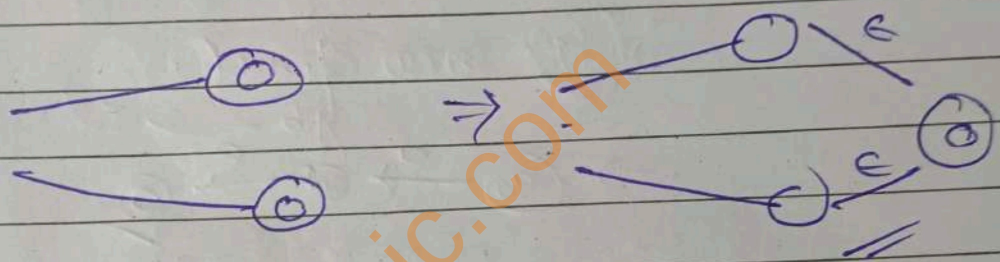


State Elimination Method

→ Convert given FA to RE.

Step 1:- If initial have any incoming then remove it from initial.

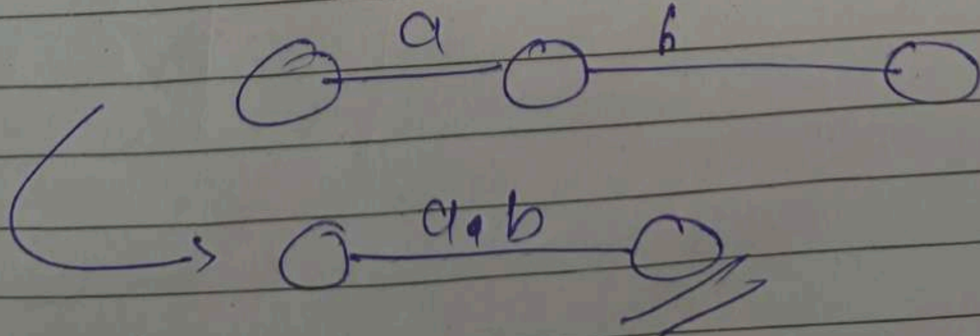
Step 2:- if there are more than one final state then make it one.



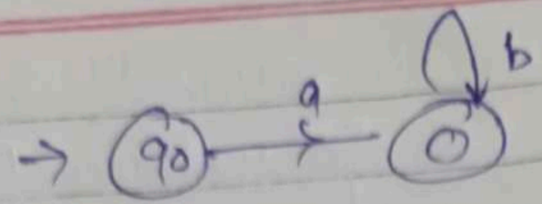
Step 3:- If final state have any outgoing then add new final. (including self loop).

Step 4:- Repeat until only initial and final remains.

→ Eliminate remaining states.



Q 1)

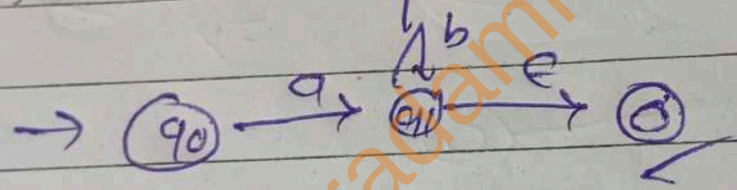


Step 1 No incoming

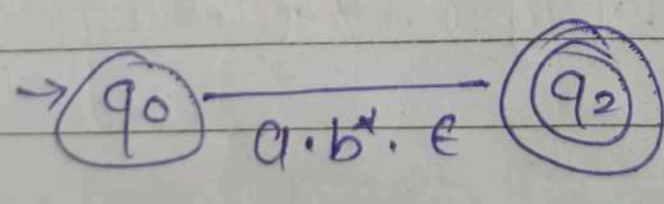
Step 2 : 1 final.

Step 3! Outgoing ✓

Make new final.

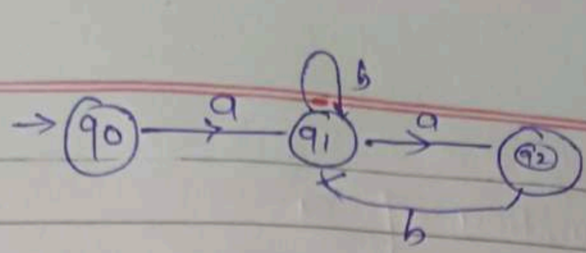


Step 4! Eleminate q1



$RE = (a \cdot b^*)$

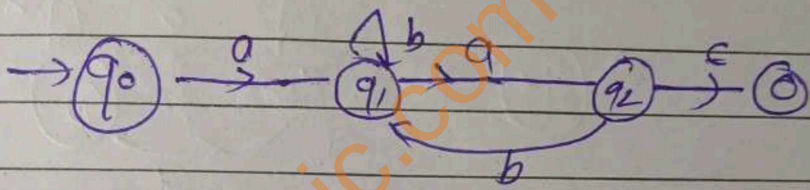
2)



Step1:- No incoming edge.

Step2:- 1 final

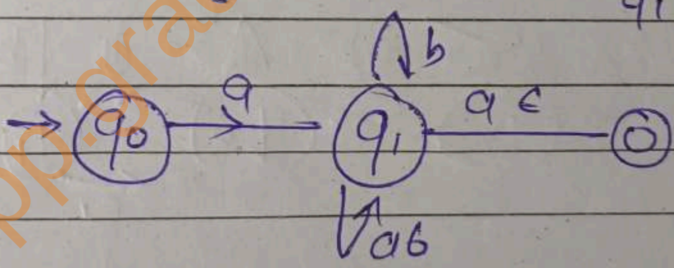
Step3:- Outgoing ✓ new f

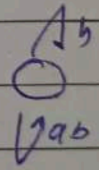


Step 4 Eleminate q_2

$q_1\ q_2\ q_3 = a\epsilon$

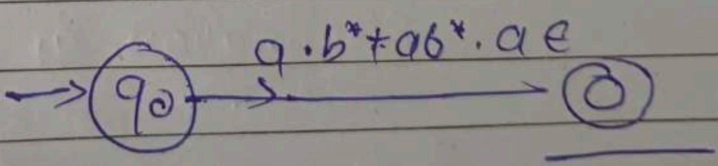
$q_1\ q_2\ q_1 = ab$



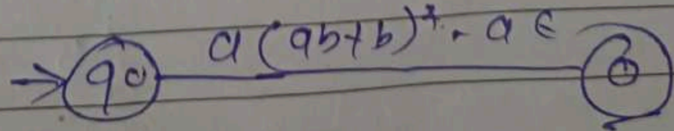


$= ab+b$

Eleminate q_1

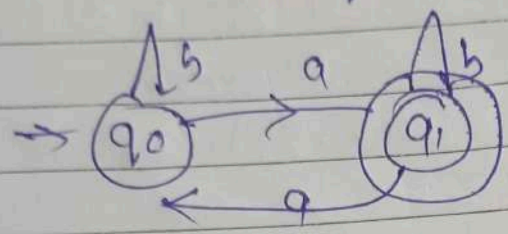


or

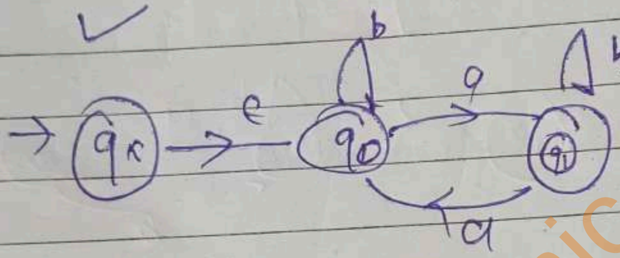


$RE = a\cdot(ab+b)^*+a$

Odd number of a

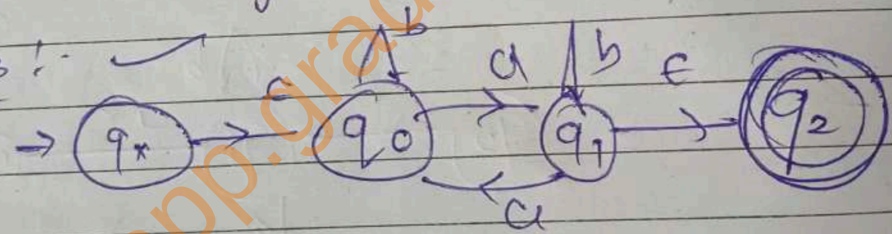


Step1 →



Step 2 :- Single final

Step3 :-

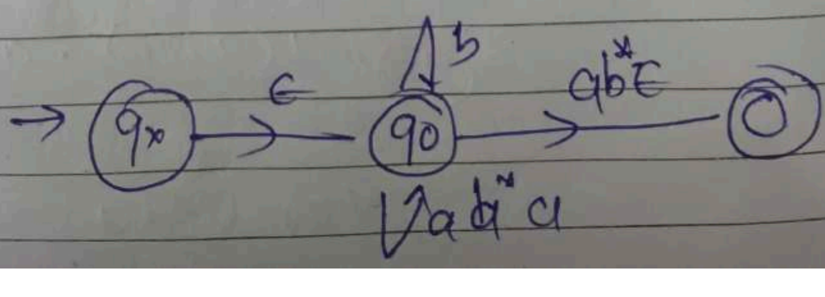


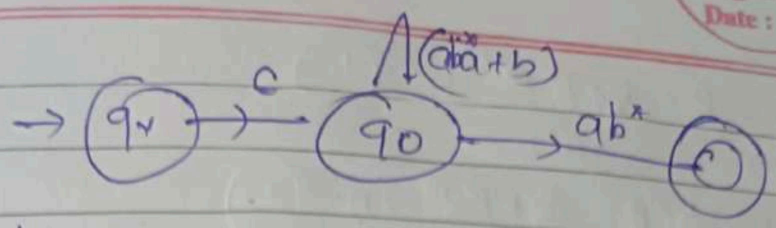
Step4 :-

Elemination q_1

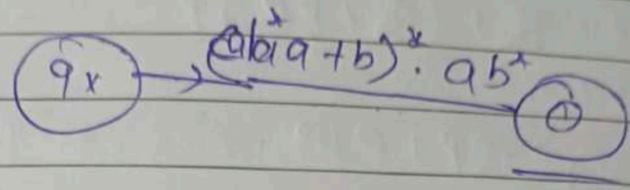
$q_0\ q_1\ q_2 = ab^*\epsilon$

$q_0\ q_1\ q_0 = ab^*a$





Eleminate q_0



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

Arden's Theorem

If P and Q are two RE over Σ and if P doesn't contain 'ϵ' then the following eqn in R given by.

$$R = Q + RP$$

The eqn will be replaced by

$$R = QP^*$$

↳ Why?

PTO for clarity

$R = Q + RP$ — (1)

put $R = Q + RP$ in (1)

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

$R = Q + QP + RP^2$

$\cancel{R = Q(P + RP^2)}$

again put $R = Q + RP$

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

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

again

$R = Q + QP + QP^2 + QP^3 + RP^4 \ldots$

a——

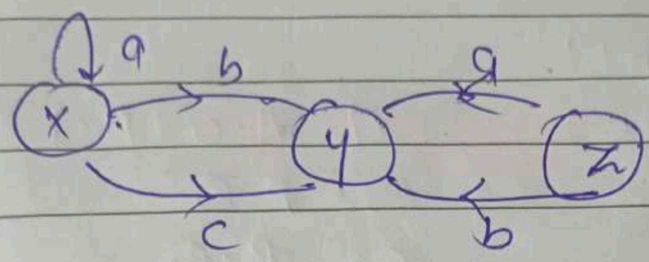
take Q as c——

$R = Q + (P^0 + P^1 + P^2 + P^3 \text{——})$

$\boxed{R = Q + P^*}$

Conversion to RE

Step 1:- find the eqⁿ for all of the states shown.



Check indegree eqⁿ.

$$X = X \cdot a$$

$$y = x \cdot b + x \cdot c + z \cdot b + z \cdot a$$

$$y = x(b+c) + z(b + a)$$

z = no indegree.

Step 2:- put ϵ in the initial state

$$x = \epsilon + x \cdot a$$

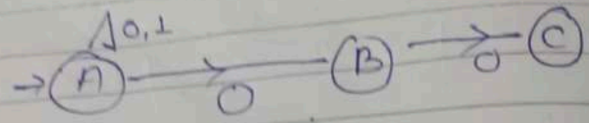
$$y = x(b+c) + z(b+a)$$

Step 3: find the solution by substituting method.

$$R = Q + RP$$

$$\hookrightarrow R = QP^*$$

Q.



Step.1 :

$A = A\cdot 0 + A\cdot 1$

$B = A\cdot 0$

$C = B\cdot 0$

Step 2 :- Add ϵ.

$A = \epsilon + A\cdot 0 + A\cdot 1$ = $A = \epsilon + A($

$B = A\cdot 0$

$C = B\cdot 0$

Step 3 :

$$\underbrace{A}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{A}_{R}\underbrace{(0+1)}_{P}$$

$\hookrightarrow$ $A = \epsilon + (0+1)^*$

put value of A in B.

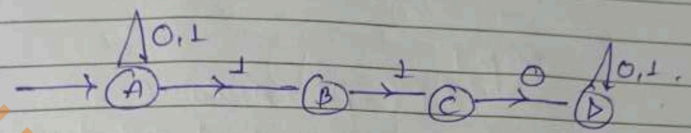
$B = (0+1)^* 0$

put value of B in C.

$C = (0+1)^* 0\cdot 0$ = RE

Q Design NFA and find RE using Arden's Method for a language L that consist of substrings.

$L = \{110\}$. $\Sigma = \{0,1\}$



Step 1: Eq^ns.

$A = A(0+1)$

$B = A \cdot 1$

$C = B \cdot 1$

$D = C \cdot 0 + D(0+1) = C \cdot 0(0+1)^*$

$A = \epsilon + A(0+1)$

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

put in B.

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

put in C.

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

put in D

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

Page No.
Date: / /

Pumping lemma for regular expression

Pumping lemma is also known as negative test. Pumping lemma is used to prove that the language is non-regular.

Pumping lemma is never been used to prove that the language is regular, but it only used to prove that the language is non-regular.

Pumping lemma Theorem.

Pumping lemma theorem states that if the language is regular that at some point the language will be divided into xy^iz with following conditions.

1) $y \notin \epsilon$ or $|y| > 0$
2) $|x.y| \leq P$
3) Every string pattern $xy^iz \in L$ such that $i \geq 0$

$$L = \{0^n 1^n\} \quad 1 \leq n.$$

$$S = \{0^P . 1^P\} \quad |S| \geq P = 2P$$

$$P = a + b + c.$$

$$S = \underbrace{0^a}_{x} \underbrace{0^b}_{y} \underbrace{c^c \quad 1^P}_{z}.$$

for $i = 1$

$xy^1z = 0^a 0^b 0^c 1^p \in L.$

for $i = 2.$

$xy^2z = 0^a 0^b 0^b 0^c 1^p$

$\hookrightarrow \underbrace{0^a 0^b 0^c}\, 0^b\, 1^p$

$\downarrow$

$0^p\, 0^b\, 1^p \notin L$

failed because of an extra zero.

failed then proved that the followi[ng] expression is non-regular.

Page No.
Date : / /

Q Prove that the language $L = \{0^p \mid p \text{ is a prime no}\}$ is non-regular using Pumping lema.

$\cancel{S=0^1}$ $S = 0^p$ $|S| \geq p$

$s = xyz$

$$S = 0^p = \underbrace{0^a}_{x}\underbrace{0^b}_{y}\underbrace{0^c}_{z} \quad \Big| \quad P = a+b+c$$

$x y^i z \qquad i \geq 1$

for $i=1$:

$$S_1 = xy^1z = 0^a(0^b)^1 0^c = 0^a 0^b 0^c \in L \checkmark$$

for $i=2$:

$$S_2 = xy^2z = 0^a(0^b)^2 0^c = 0^a 0^b 0^b 0^c$$

$$\underbrace{0^a 0^b 0^c}\ 0^b$$

$$\downarrow$$

$$0^p\ 0^b = 0^{p+b}$$

for $i=3$,

$$S_3 = xy^3z = 0^a(0^b)^3 0^c$$

Page No.
Date : / /

$\hookrightarrow 0^a 0^b 0^b 0^b 0^c$

$\Rightarrow \underbrace{0^a 0^b 0^c}_{p} \underbrace{0^b}_{b} 0^b$

$\rightarrow S_2\, 0^b \notin L$

Q Prove that language $L = \{a^n b^{2n}\}_{n \geq 1}$ is not a regular language.

$S = \underbrace{aaa}_{x}\underbrace{abb}_{y}\underbrace{bbbbbb}_{z} \in L$

$i = 1$, $S_1 = xy^1z = aaa(abb)^1 bbbbbb$

$i = 2$, $S_2 = xy^2z = \underbrace{aaa}_{x}\underbrace{(abbabb)}_{y}\underbrace{bbbbbb}_{z}$

$\notin L$

$S = \underbrace{aaaa}_{x}\underbrace{bbbb}_{y}\underbrace{bbbb}_{z} \in L$

$i = 1$, $S_1 \in L$

$i = 2$, $S_2 = xy^2z = aaaa(bbbbbbbb)bbbb$

$\notin L$