5/9/17

SURYA Gold
Date Page

Module-2

Regular Expression

Regular Expression

It is a string that can be formed by following rules:-

i) ϕ is a Regular Expression
ii) ε is a "
iii) Every symbol in Σ is a Regular Expression
iv) Given 2 Regular expressions α & β then $\alpha \cup \beta$ is a Regular expression.
v) Given a Regular expression α & β then $\alpha.\beta$ is a RE.
vi) Given here a RE let it be α, then α^* is a RE.
vii) Given a RE let it be α, then α^+ is a RE
viii) Given a RE let it be α, (α) is also a RE.

Note:- Regular expressions are supported by 3 operators and are listed, as follows from higher precedure to lower precedence.

1) (*) Kleen star
2) dot / concatenation (.)
3) union operator ($\cup$)

SURYA Gold
Date Page

RE	Language	Interpretation
ϕ	$L = \{\ \}$	Empty language
ε	$L = \{\varepsilon\}$	Lang contains only Epsilon.
a	$L = \{a\}$	Lang contains only 1 string a.
a^*	$L = \{\varepsilon, a, aa, aaa...\}$	Lang contains 0 or more no. of a's.
a^+	$L = \{a, aa, aaa....\}$	Lang contains 1 or more no. of a's.
$a \cup b$	$L = \{a, b\}$	Lang contains either a or b.
$(a \cup b)^*$	$L = \{\varepsilon, a, aa, b, bb, ab, ba, bba...\}$	strings of a's & b's of any length.

7/9/17

Give regular expression for the following languages.

a) strings of a's and b's that contain the substring ab at the beginning

$$ab(a \cup b)^*$$

$$\{\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\}$$
$$\{(a \cup b)^* = (a \cup b)^0 \cup (a \cup b)^1 \cup (a \cup b)^2 \cup \ldots\}$$

b) strings of a's and b's of even length.

$\{aa \cup bb \cup ab \cup ba\}$.

$$((a \cup b) . (a \cup b))^*$$

c) strings of a's and b's of length ≤ 2.

$$(\varepsilon \cup a \cup b) . (\varepsilon \cup a \cup b)$$

d) strings of zero's and 1's that contain the substring 000.

$$(0 \cup 1)^* 000 (0 \cup 1)^*$$

e) $L = \{w \in \{a,b\}^* ; |w|_a \equiv_3 0\}$.

accepts strings of length 3 and multiples of 3.

$$\{(a \cup b).(a \cup b).(a \cup b)\}^*$$

f] strings of 0's and b's whose 10th symbol from the right end is a.

~~(aub).(aub).(aub).(aub).(aub).(aub).(aub).(aub).(aub).a.~~

$(a \cup b)^* a (a \cup b)^9$.

g] $L = \{a^n b^m ; n \geq 4, m \leq 3\}$

$(a \cup b)^* \neq a^* \cup b^*$

$\{\varepsilon, a, aa, aaa \ldots\} \cup \{\varepsilon, b, bb, bbb, \ldots\}$

$aaaaa^* . (\varepsilon \cup b) . (\varepsilon \cup b) . (\varepsilon \cup b)$

h] strings of a's and b's of even length as well as odd length.

~~$\{(a \cup b) . (a \cup b)\}^* \cup \{a \cup b\}^*$~~

$\{(a \cup b) . (a \cup b)\}^* . a \cup \{(a \cup b) . (a \cup b)\}^* . b$

2 Design a transducer for bar code reader.

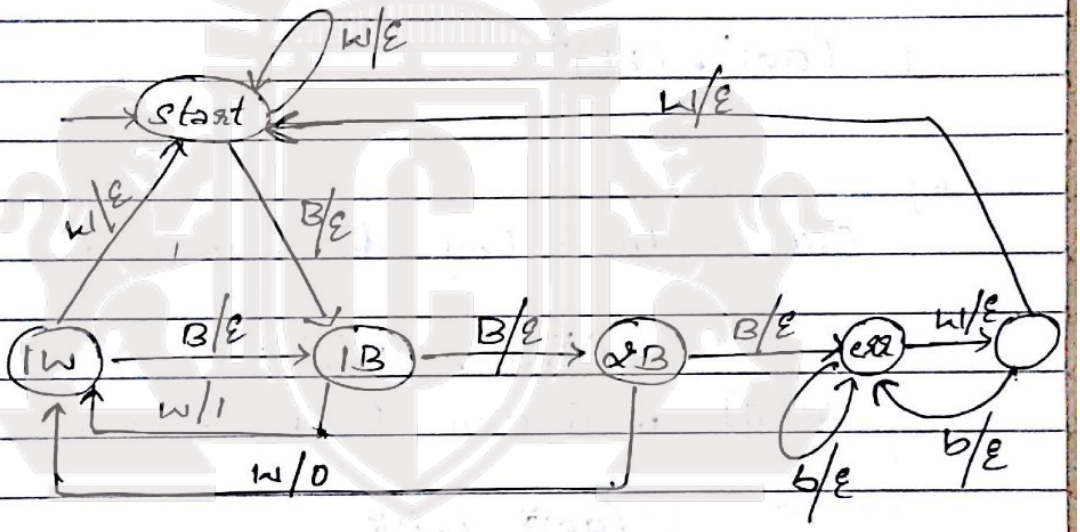
① Each bar code contains black columns & white columns of same width.

② A single black column is encoded as 1

③ A double black column is encoded as 0.

④ White columns are used as delimiter.

⑤ After every bar code is completed or error bar code exist. the machine has to read 2 white columns in order to reset to read the next bar code.



3. Set of strings of 0's & 1's that do not end with 01.

4. strings of a's & b's containing not more than 3 a's.

5. $L = \{a^{2n} b^{2m} / n>0, m>0\}$.

6. $L = \{a^n b^m / m+n \text{ is even}\}$

7. strings of 0's and b's whose second symbol from right is a & four symbol from right is b. also draw NDFSM

8. Strings of a's & b's of length ≤ 10.

3. $(0 \cup 1)^* . (00 \cup 11 \cup 10)$.

4. $(\varepsilon \cup a \cup b) . (\varepsilon \cup a \cup b) . (\varepsilon \cup a \cup b) . (\varepsilon \cup b)^*$

$(\varepsilon \cup a \cup b)^3 . (\varepsilon \cup b)^*$

~~$b^* . (\varepsilon \cup a)^3 . b$~~

$b^* . a . b^* . a\, b^* . a\, b^*$

$b^* . (\varepsilon \cup a) . b^* . (\varepsilon \cup a) . b^* . (\varepsilon \cup a) . b^*$

5] $(aa)^+ . (bb)^+$

6]

Case i) m is Odd & n is Odd.

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

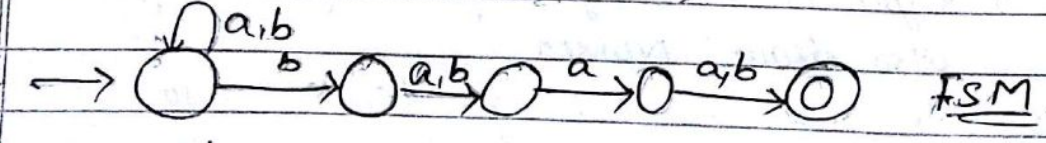
Case ii] m is even & n is even.

$(aa)^* . (bb)^*$

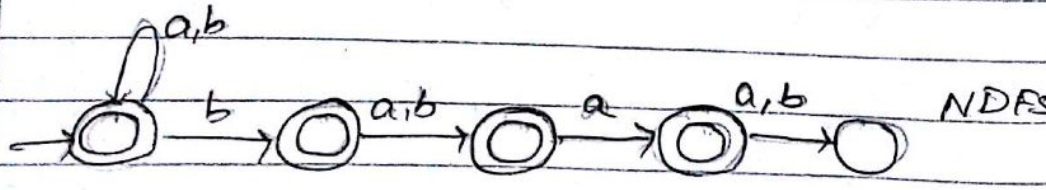
$\{(aa)^* . a . (bb)^* . b\} \cup \{(aa)^* . (bb)^*\}$

7]

$(a \cup b)^* b . (a \cup b) . a . (a \cup b)$



FSM



NDFSM.

8] $(\varepsilon \cup a \cup b)^b$

9] $L = \{ w / n_a(w) \bmod 3 = 0 \}$.
on $\Sigma = \{a, b\}^*$
length of a in string is 3 or multiples of 3.

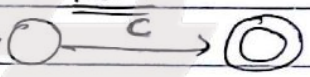
$b^* (a\ b^*\ a\ b^*\ a)^*\ b^*$
$\Rightarrow (b^* a b^* a b^* a b^*)^*$

11/9/14

Kleen's theorem

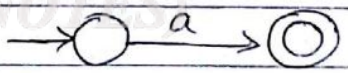
Part 1 :- For every regular expression there is an equivalent FSM,
Any language that can be defined with a regular expression can be expected by some FSM and is regular.

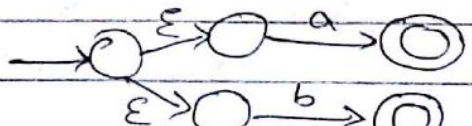
RE's (Regular Expressions) FSM

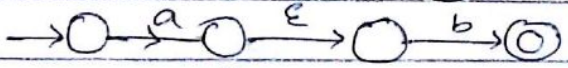
1) $c \in \Sigma$ 

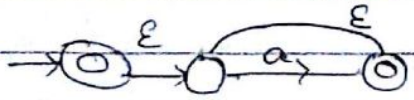
2) $\emptyset$ 
neither start state, nor accepting state

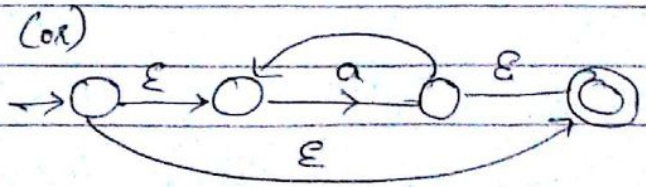
3) $\emptyset^* = \varepsilon$ 

4) $a \in \Sigma$ 

5) $a \cup b$ 

6) $a \cdot b$ 

7) a^* 

(or) 

→ For the regular expression $\alpha \cup \beta$, $L(\alpha)$ & $L(\beta)$ are the corresponding regular language.

Let us construct a language L_3 to accept the language $L(\alpha \cup \beta)$ given $M_1 = (K_1, \Sigma, \delta_1, S_1, A_1)$

$M_2 = (K_2, \Sigma, \delta_2, S_2, A_2)$

Soln

$$M_3 = \{ \overbrace{S_3 \cup K_1 \cup K_2}^{K_3}, \Sigma, \delta_3, S_3, \overbrace{A_1 \cup A_2}^{A_3} \}.$$

$$\delta_3 \rightarrow \{ \delta_1 \cup \delta_2 \cup ((S_3, \varepsilon), S_1) \cup ((S_3, \varepsilon), S_2) \}.$$

For RE $\alpha \cup \beta$, $L(\alpha)$ & $L(\beta)$ are the corresponding Regular languages.

Now let us construct M_3 to accept the language $L(\alpha \beta)$, given,

$M_1 = (K_1, \Sigma, \delta_1, S_1, A_1)$

$M_2 = (K_2, \Sigma, \delta_2, S_2, A_2)$.

$$M_3 = \{ K_1 \cup K_2, \Sigma, \delta_3, S_1, A_2 \}, \quad q \in A_2$$

$$\delta_3 = \{ \delta_1 \cup \delta_2 \cup ((q_2, \varepsilon), S_2), \ q_2 \in A_1 \}.$$

For the regular expression α, let us construct the machine M_2 which accepts the language $L(\alpha)^*$

Given $M_1 = (K_1, \Sigma, \delta_1, S_1, A_1)$

$$M_2 = (K_1 \cup S_2, \Sigma, \delta_2, S_2, A_1 \cup S_2).$$

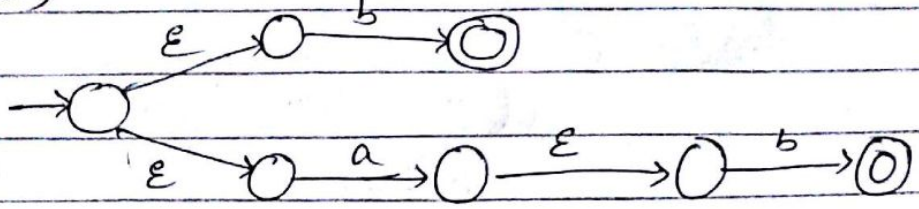
$$\delta_2 = \{ \delta_1 \cup ((S_2, \varepsilon), S_1) \cup ((q, \varepsilon), S_1)) \ q \in A_1 \}$$

$b^* \cup (ab)^*$ SURYA Gold

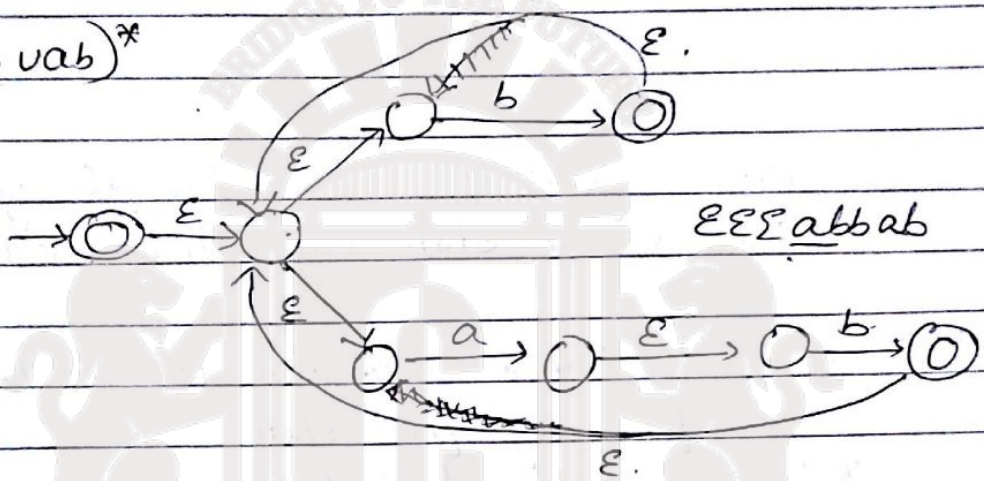
Date _____ Page _____

Draw FSM from the given regular expression.

(1) $(b \cup ab)$

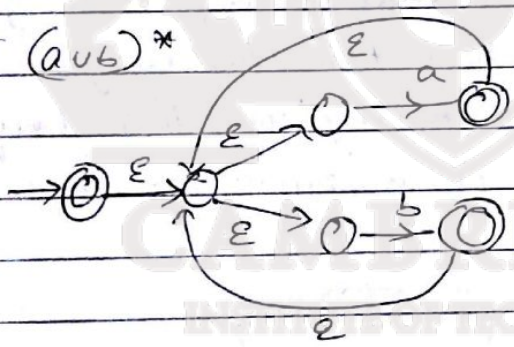


(2) $(b \cup ab)^*$

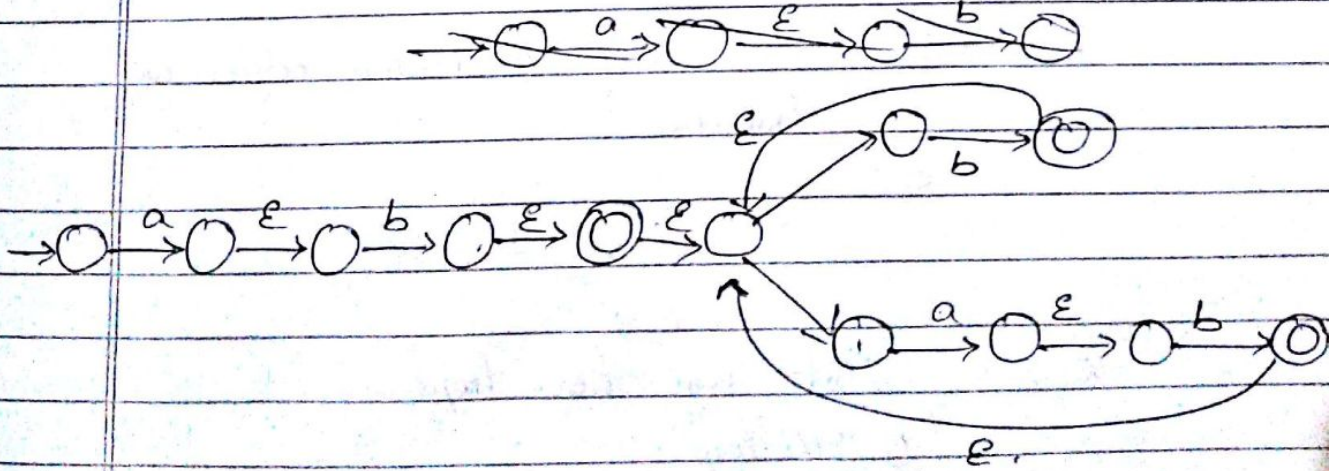


εεε abbab

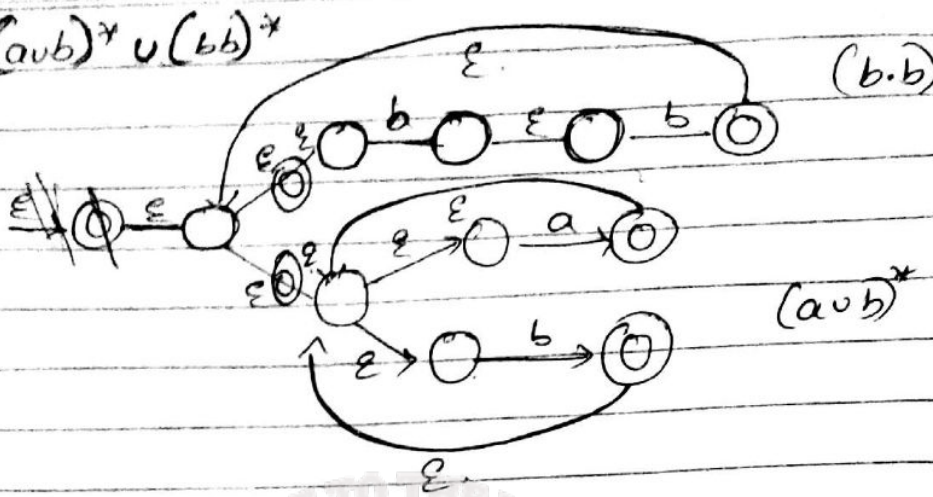
(3) $(a \cup b)^*$



(4) ~~$ab \cdot b \cup (ab)^*$~~ $ab(b \cup ab)^*$



5) $(a \cup b)^* \cup (bb)^*$



Build Regular Expression from the given FSM using state elimination Method.

Rule 1 :- Regular Expression is resolved for every final state.

Rule 2 :- eliminate dead or trap state from given FSM transition diagram.

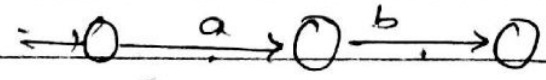
Rule 3 : A state having more than 1 edge or transition in a single direction.

Ex :-

RE = a∪b.

Rule 4 :- states having consecutive edges or transitions.

Ex :-



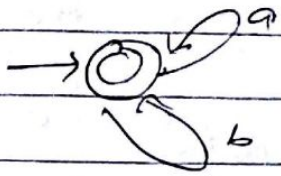
RE = ab

Rule 5 :- States can have loops.

i) Self loop

RE = a*

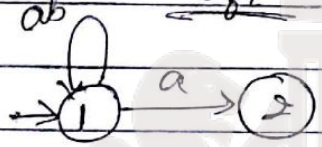
ii] More than 1 self loop.



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

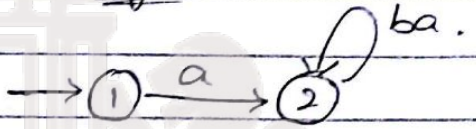
iii] loops b/w 2 states

Left



$(ab)^* \cdot a$

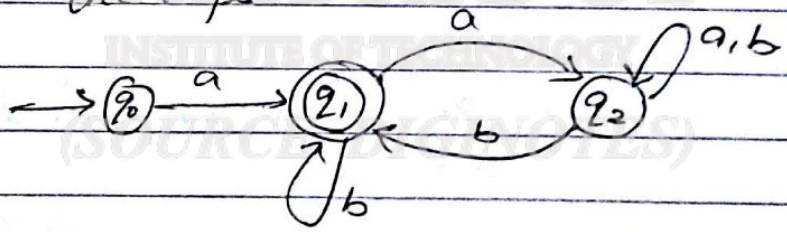
Right

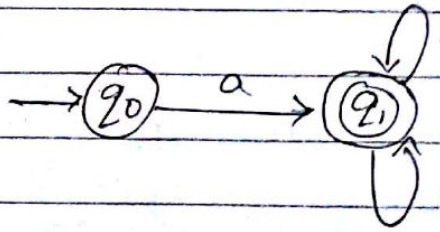


$a \cdot (ba)^*$

$$(ab)^* a = a(ba)^*$$

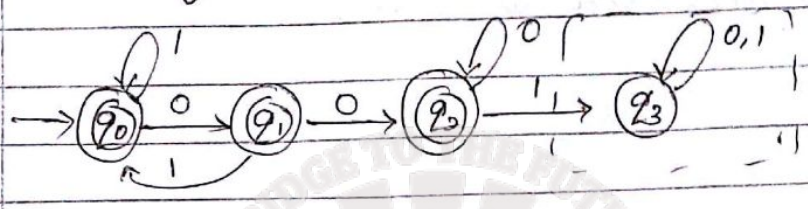
Rule 6 :- if the final state is middle state with the loops.



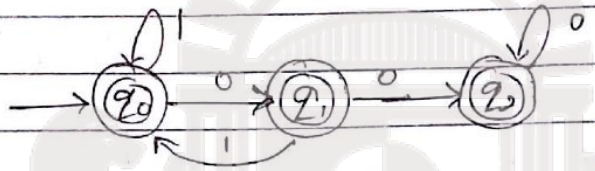


$$= a \cdot (b \cup a(a \cup b)^* b)^*$$

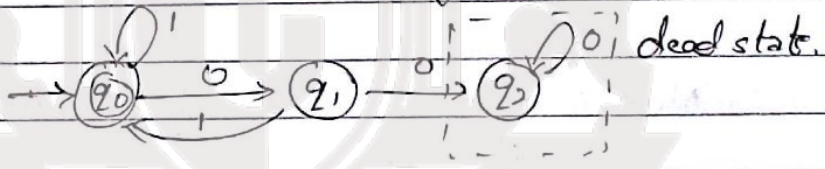
Rule 4 :- if there is more than 1 final state, then write RE for every final state & bring them into single regular expression using union operator

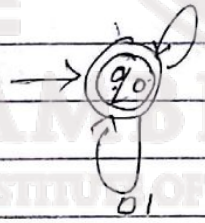


eliminate dead state q_3.



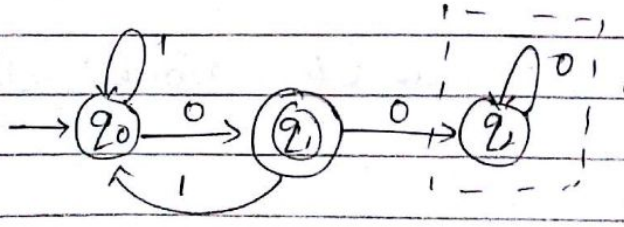
Consider q_0 as final state, q_1 & q_2 as non-final state.

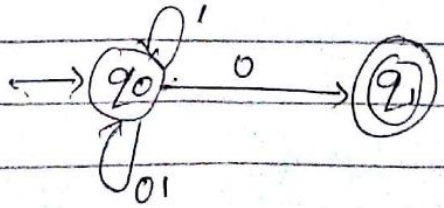




$RE_1 = (1 \cup 01)^*$

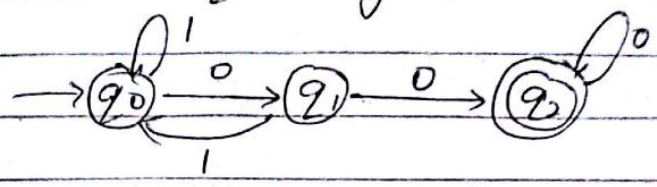
consider q_1 as final state.





$RE_2 = (1 \cup 01)^* . 0$

Consider q_2 as final state.



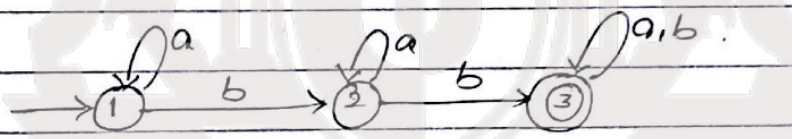
$RE_3 = (1 \cup 01)^*.0.0.0^*$.

$RE = RE_1 \cup RE_2 \cup RE_3$.

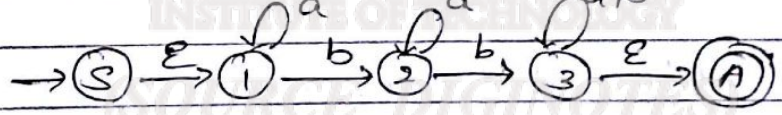
$RE = (1 \cup 01)^* \cup (1 \cup 01)^*.0 \cup (1 \cup 01)^*.0.0.0^*$

$= (1 \cup 01)^* (\varepsilon \cup 0 \cup 000^*)$

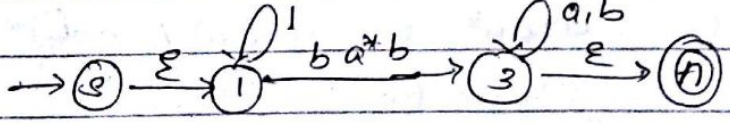
① Build RE from the given FSM.



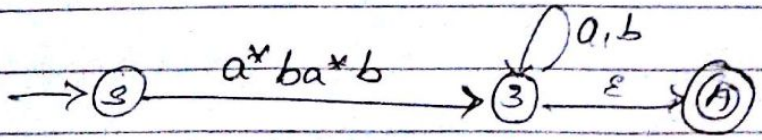
Introduce the new ~~state~~ start state and accepting state, leaving these 2 states eliminate all other states.



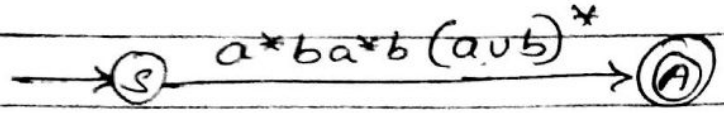
Eliminate state 2 :-

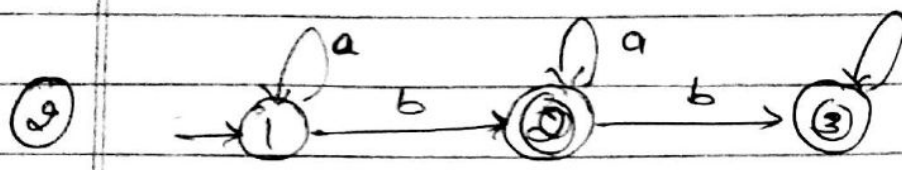


eliminate state 1 :-

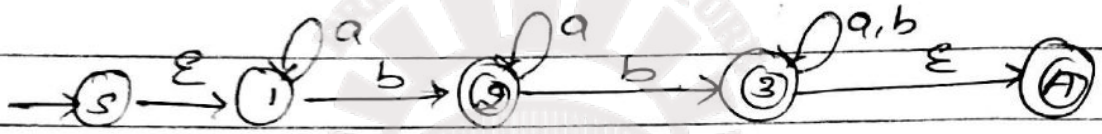


eliminate state 3:-





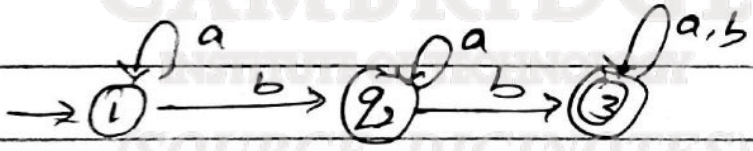
add the new start state & accepting state.



Consider q_2 as final state.

$RE_1 = a^*ba^*$

Consider q_3 as final state.



$RE_2 = a^*ba^*b(a \cup b)^*$

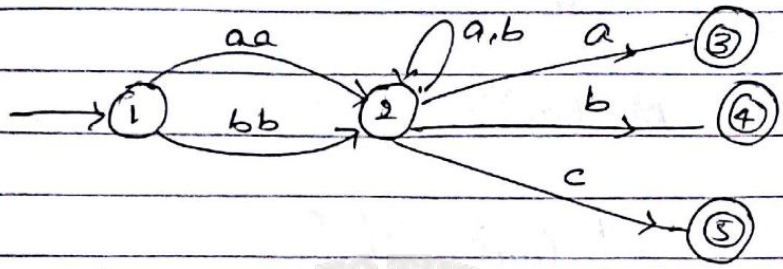
$RE = a^*ba^* \cup a^*ba^*b(a \cup b)^*$

$= \boxed{a^*ba^*(\varepsilon \cup b(a \cup b)^*)}$

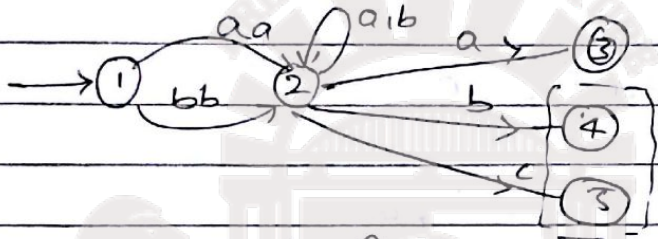
12/9/17

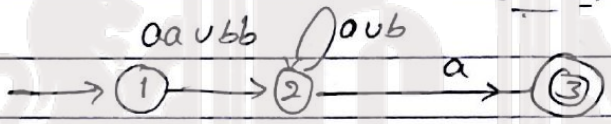
SURYA Gold
Date _____ Page _____

(3) Find the RE from the given FSM state elimination Method.



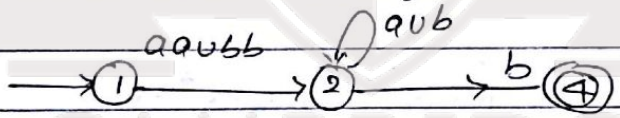
consider, 3 as final state.





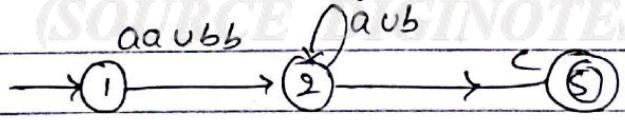
$$RE_1 = (aa \cup bb) \cdot (a \cup b)^* a$$

consider 4 as final state



$$RE_2 = (aa \cup bb) \cdot (a \cup b)^* \cdot b$$

consider 5 as final state.



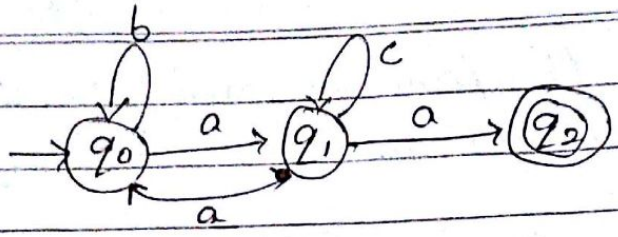
$$RE_3 = (aa \cup bb)(a \cup b)^* \cdot c$$

$$RE = RE_1 \cup RE_2 \cup RE_3$$

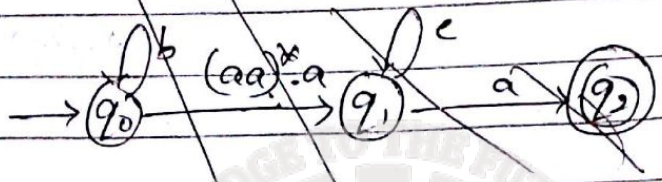
$$= (aa \cup bb) \cdot (a \cup b)^* (a \cup b \cup c)$$

Date Page

(4)

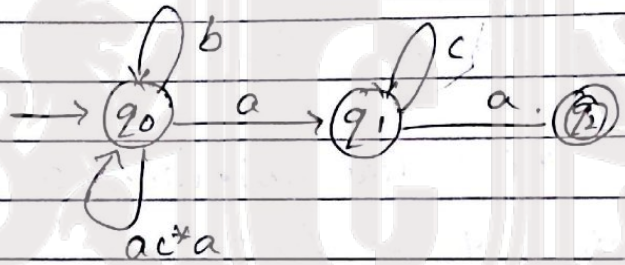


$b^* \cdot (aa)^* \cdot a \cdot$



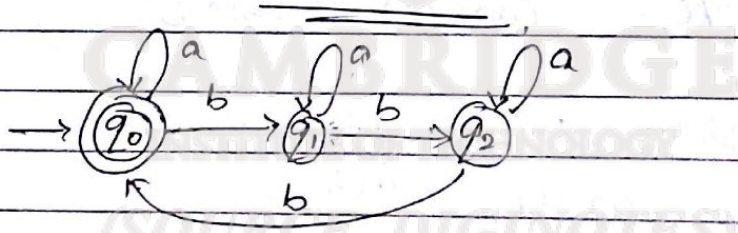
$b^* (aa)^* a \cdot c^* \cdot a$

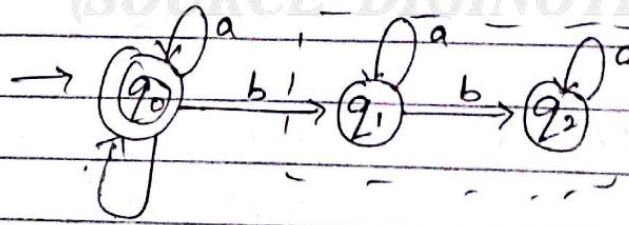
$b^* (ac^*a)^* \cdot a$



$[b^* \cup (ac^*a)^*] \cdot a \cdot c^* \cdot a$

(5)

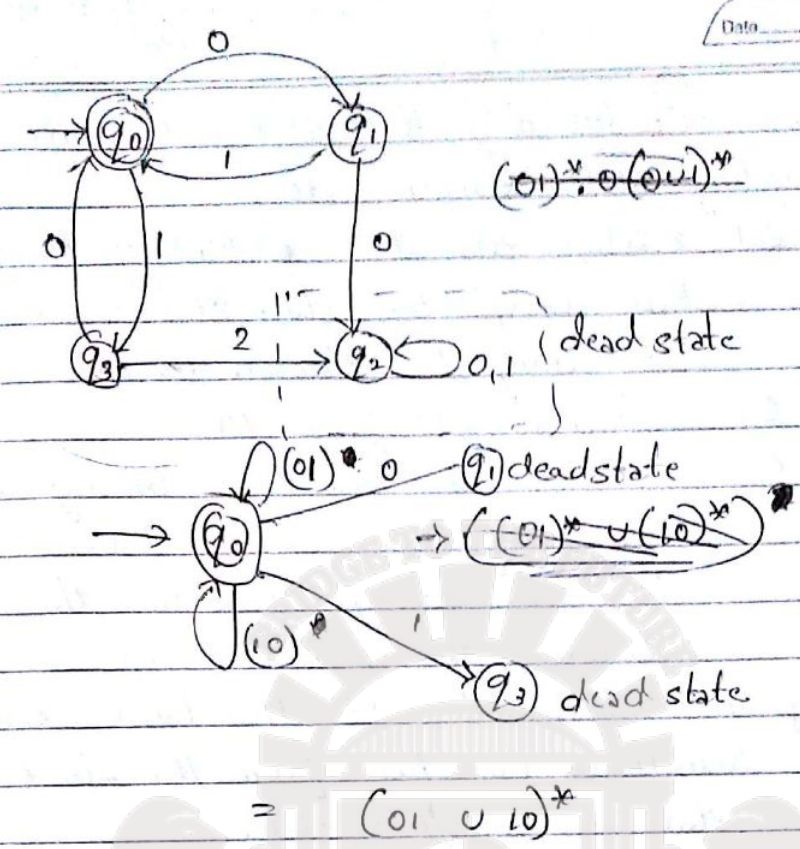




$b a^* b a^* b$

$= [a^* \cup (ba^* ba^* b)^*] \cdot b \cdot a^* \cdot b \cdot a^*$

(6)



~~$(01)^* \cdot 0 (0 \cup 1)^*$~~

$= (01 \cup 10)^*$

Building a RE from FSM

Step 1 → Remove from machine M (FSM) any states that are unreachable.

Step 2 → if M as no accepting state then halt and return ∅

Step 3 → if the start state of the machine is part of loop. create new state & s and connect s to start state via a new transition

Step 4 → if their is more than 1 accepting state but if there are transitions out of it create a new transitions states and connect each of M's accepting state to new one by epsilon transition.

Step 5 → if M as only one start, of which is both start and accepting state, then $L(M) = \varepsilon$, provided if M as no other transition.

Step 6 → until only the start state and accepting state remain do.

6.1 → Select some state (RIP) of M any state except the start state or accepting state may be choosen as RIP.

6.2 → Remove RIP from M

6.3 → Modify the transitions among the remaining states so that M accepts the same strings and now the labels becomes the Regular Expressions.

Step 7 → Return the RE that labels one remaining transition from the start state to the final state.

Step 6 → untill only the start state and accepting state remain do,

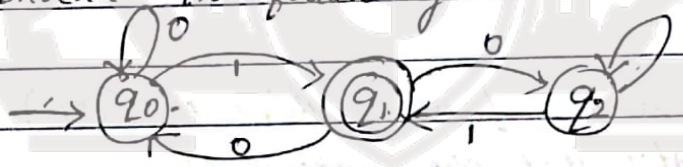
6.1 → select some state • (RIP) of M any state except the start state or accepting state may be choosen as RIP.

6.2 → Remove RIP from M

6.3 → Modify the transitions among the remaining states so that M accepts the same strings and now the labels becomes the Regular Expressions.

Step 7 → Return the RE that labels one remaining transition from the start state to the final state.

Convert the following NDFSM to DFSM.

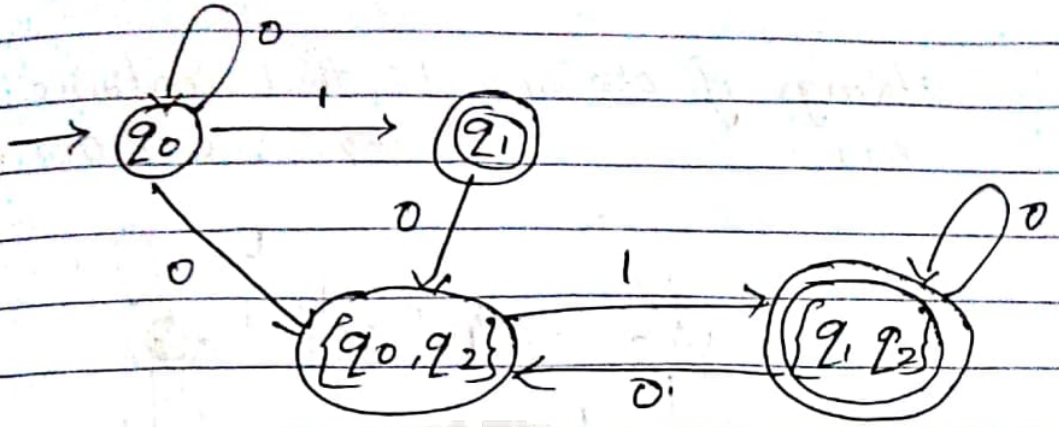


A	0	1
→ q_0	q_0	q_1
* q_1	q_2, q_0	-
q_2	-	q_2, q_1

NDFSM

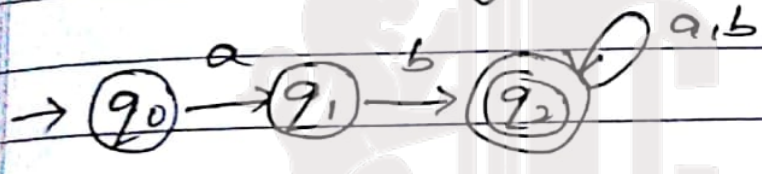
Transition of DFSM.

A	0	1
→ q_0	$\{q_0\}$	$\{q_1\}$
* q_1	$\{q_0\ q_2\}$	—
$\{q_0\ q_2\}$	$\{q_0\}$	$\{q_1\ q_2\}$
$\{q_1\ q_2\}$	$\{q_0\ q_2\}$	$\{q_1\ q_2\}$

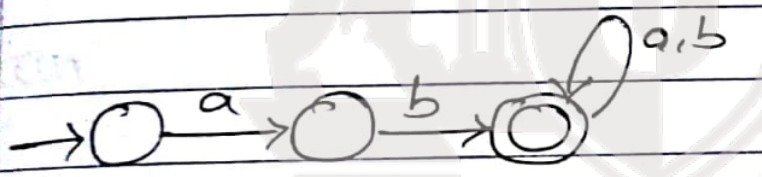


Recap

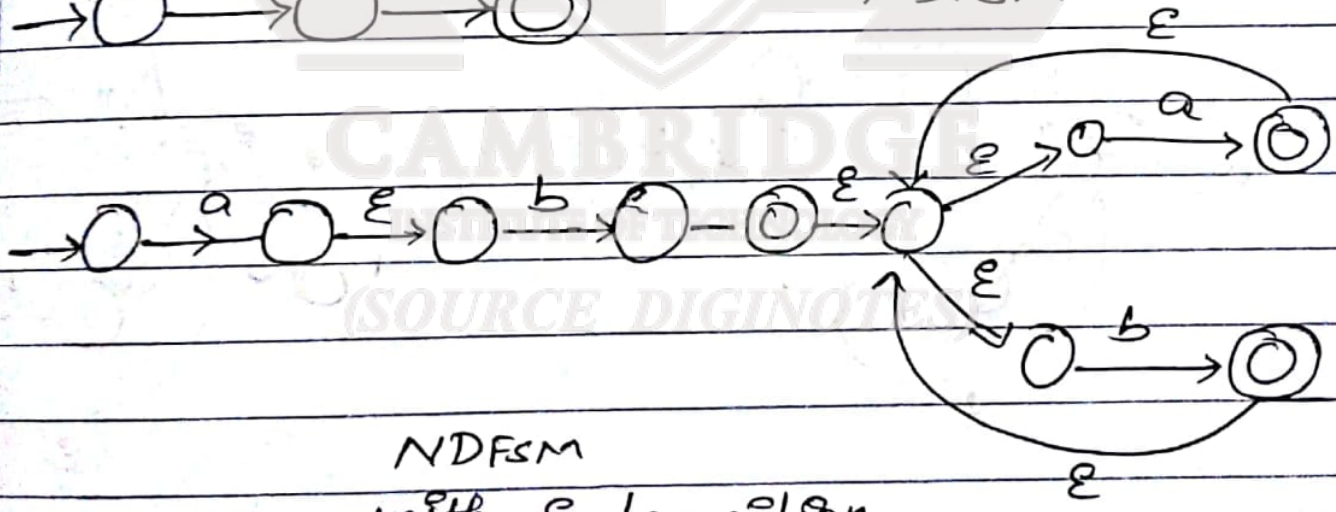
$ab(a \cup b)^*$ → Strings of a's and b's that begin with ab.



DFSM



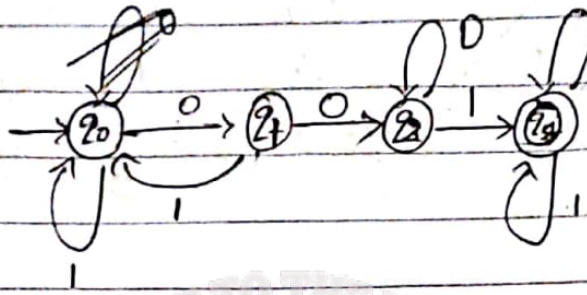
NDFSM

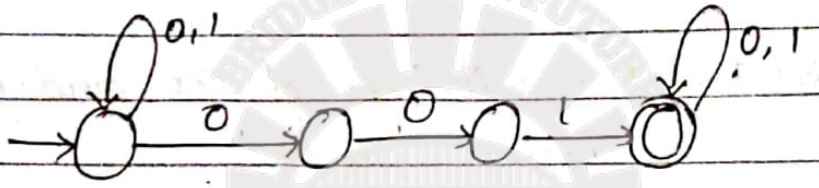


NDFSM with ε transition.

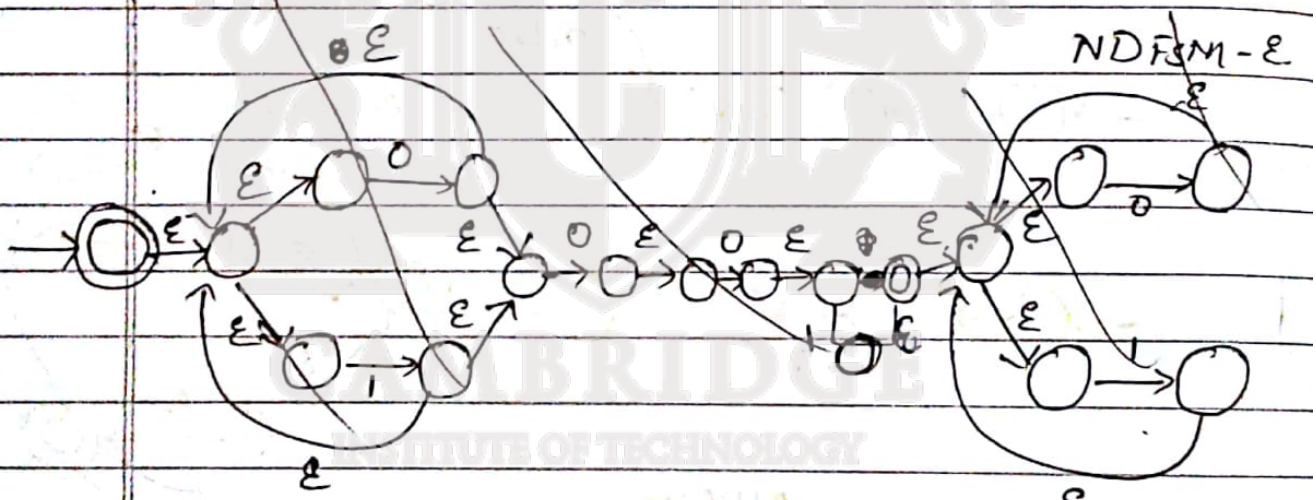
strings of 0's and 1's that contains substring 001.

$\Rightarrow (0 \cup 1)^* 001 (0 \cup 1)^*$

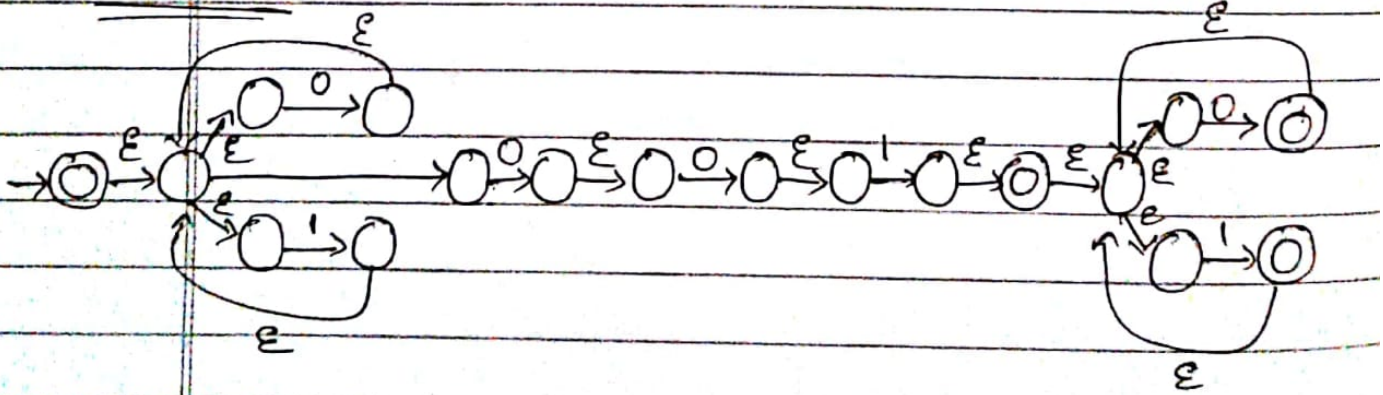




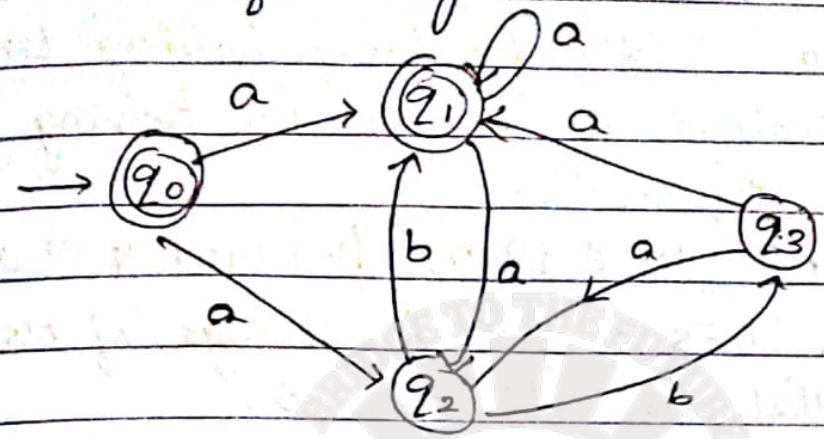
NDFSM-ε



NDFSM-ε



convert the following NDFSM to DFSM.

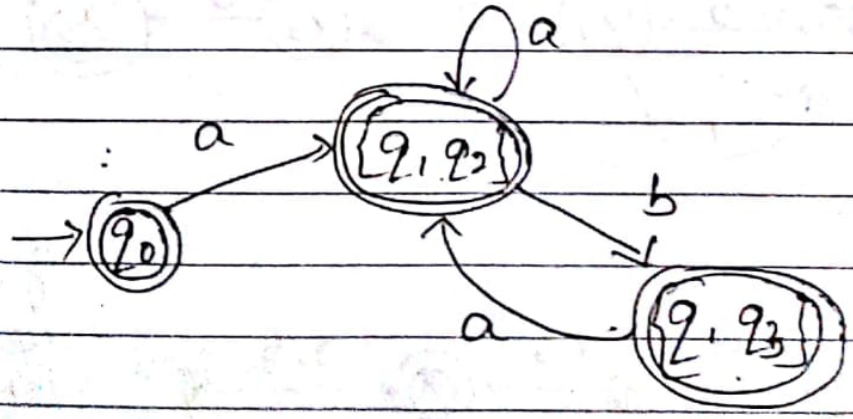


NDFSM

A	a	b
→*q_0	q_1, q_2	—
*q_1	q_1, q_2	—
q_2	—	q_1, q_3
q_3	q_1, q_2	—

DFSM.

A	a	b
→*q_0	$\{q_1, q_2\}$	—
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_3\}$
$\{q_1, q_3\}$	$\{q_1, q_2\}$	—



21/9/17

Regular grammer

Every regular language can be written as a set of rules which is called as Regular grammer

Regular grammer is defined as follows

$G = (V \Sigma R S)$

V = set of variables (Non-terminals) (caps)

Σ = set of input alphabet of terminals.

R = set of production Rules of the form

$$x \rightarrow y$$

S = start symbol, which belongs to V

LHS of the Production Rule always contain a single variable.

RHS of the Production Rule can contain either ε or a single terminal (a). or a terminal followed by a variable.

Ex:-

R: $S \rightarrow aS$ $S \rightarrow as$

$S \rightarrow a$ $S \rightarrow \varepsilon$

Ex:- $S \rightarrow aS$ (Recursive)

$S \rightarrow a$ $S \rightarrow aS$

aaS

$aaaS$

$aaaa$.

$V = \{S\}$ (caps)

$T = \{a\}$ (small)

S → start symbol

Ex:- $S \rightarrow aT$

$T \rightarrow a$

$\Rightarrow S \rightarrow aa$

$L = \{aa\}$.

$L = \{a, aa, \ldots aaaa, \ldots\}$.

$L = a^+$

Ex $S \to aS$

$S \to \varepsilon$

$S \to a$

aS

$S \to aS$

$S \to aaS$

$aaaS$

$aaaa\underline{S}$

$\downarrow \varepsilon$

$L = a^*$

Find the language generated by the following Regular grammer

(1) $S \to aS$

$S \to bS$

$S \to \varepsilon$.

$S \to a$

aS

aaS

$aabS$

$aabaS$

$S \to b$

bS

baS

$babS$.

So,

$(a \cup b)^*$

(2) Fing Regular grammer for Regular Expression $(a \cup b)^* aaaa$

$S \to aS$

$S \to bS$

~~$S \to \varepsilon$~~

$S \to aA$

~~$A \to aB$~~

$B \to aC$

$C \to a$

} generating aaaa.

$\{\varepsilon, a, b, aa, ab, ba, bb \ldots\}$. aaaa on concatenation, Not ε.

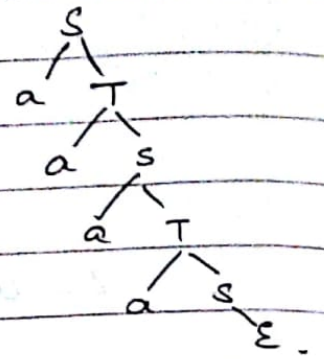
3. Find the Regular grammer for the Regular Expression $(aa)^*$.

a

$S \to aS$ $S \to aT$

$S \to a$ $T \to aS$

$S \to \varepsilon$



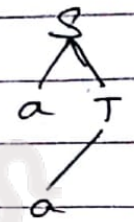
4. Find the Regular grammer for your Regular Expression $(aa \cup ab \cup ba \cup bb)$.

$\Rightarrow (a \cup b).(a \cup b)$

$S \to aT$

$S \to bT$

$T \to a/b$.



5. $(aa \cup ab \cup ba \cup bb)^*$.

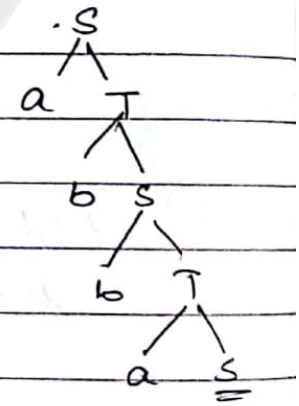
$S \to aT$ $S \to aT$

$S \to aT$ $S \to bT$

$T \to a/b/\varepsilon$. $T \to aS$

$S \to \varepsilon$ $T \to bS$

$S \to \varepsilon$.



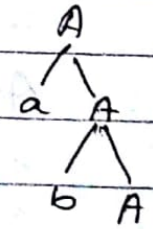
6. $a(a \cup b)^* b$

$A \to aA$ $S \to aT$

$A \to bA$ $T \to aT$

$A \to \varepsilon$ $T \to bT$

$T \to b$



7. $abb(a \cup b)^*$

$S \to aA$

$A \to bB$

$B \to bC$

$C \to aC$

$C \to bC$

$C \to \varepsilon$

8. $(a \cup b)^* abb (a \cup b)^*$

$S \to aS$

$S \to bS$

$S \to aT$

$T \to bB$

$B \to bC$

} abb.

$C \to aC$

$C \to bC$

$C \to \varepsilon$

9. Find RG for the strings of 0's and 1's that contain either even length or odd length of 0's and 1's.

$S \to 0T$

$S \to 1T$

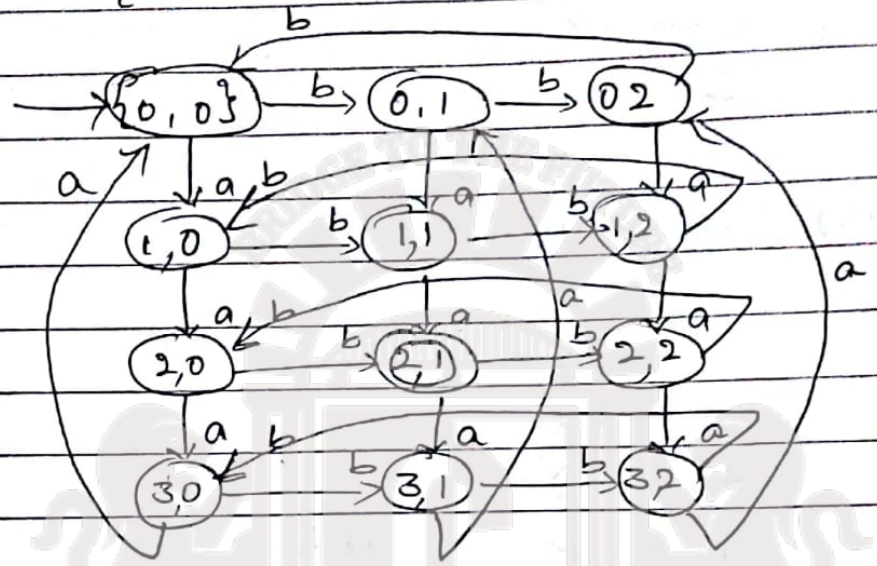
$T \to 0S$

$T \to 1S$

$S \to \varepsilon / 0 / 1$

Design DFSM for the language $L = \{N_a \bmod 4 = 2, N_b \bmod 3 = 1\}$.

$\{0\ 1\ 2\ 3\} \times \{0\ 1\ 2\}$



25/9/17

Properties of Regular languages.

RL's are closed under,

- union
- Concatenation
- Kleen star
- Complementation
- intersection
- difference
- Reversal.

The first 3 properties are proved using Kleen star part 1.

closed under complementation

Given a RL 'L' to prove L' is regular

This is prove by the method of construction

→ Construct a DFSM for the given RL 'L'

→ convert the accepting states into non-accepting states and vice versa & name the machine as M'.

Now M' is a machine which accepts the strings that are not present in regular language L, it means, it accepts the strings present in L'

Ex:-

L = strings of a's and b's that end with abb.

L' = strings of a's and b's that do not end with abb.

closed under intersection.

Let us consider there are two Regular languages L_1 and L_2.

L_1' and L_2' are also regular

Find union of L_1' and L_2'

$L_1' \cup L_2'$ is also regular lang

According to Demorgan's law.

$$(L_1' \cup L_2') = \overline{L_1 \cap L_2}.$$

closed under difference operator

Given 2 regular languages L_1 and L_2, we have to prove that $L_1 - L_2$ is regular

This is prove indirectly by the Rule

$$A - B = A \cap B'$$

$$\Rightarrow L_1 \cap L_2' = L_1 - L_2$$

~~It says X_1~~

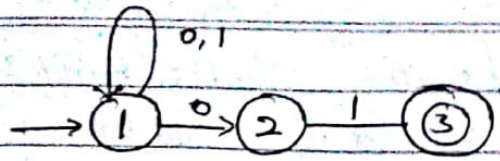
closed under Reversal property

Reverse of a string w with symbols $a_1, a_2 \ldots a_n$. is represented as w^R,

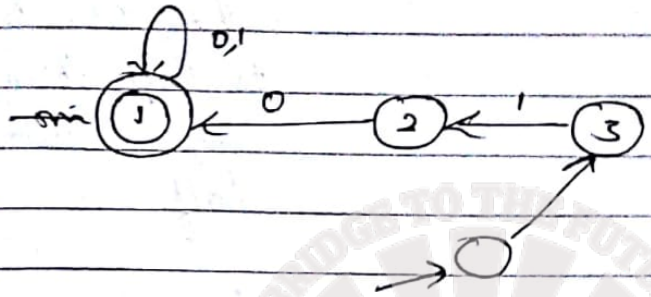
$$w^R = a_n a_{n-1} a_{n-2} \ldots a_1$$

We can prove these exist DFSM which recongnize w as well as w^R by the method of construction

1) A new state is created with ε-transition which is connecting to the final state of existing DFSM M that accepts string w.
2) Initial state is made as final state
3) All the transitions that are defined for M need to be reversed for M'
4) M' is a DFSM which recognizes reverse of the given language.



w = 01011101



Pumping Theorem for RL

It is a tool to prove, the given language is not regular, but not vise-versa.

Every finite language is regular

If a language is recognised by any FSM, then the language is R.

Able to find Regular Expression then the language is Regular.

If Regular language exist for L, then L is Regular, thus every infinite language may not be Regular to prove this consider the infinite language L as Regular, then using pumping theorem, generate more strings for that language w belongs to L.

if the generated string w, $w \notin L$ then the language is not regular.

This is prove by the method of contradiction.

Ex:- $a^n | n > 0 \longrightarrow a^+$

$a^n | n \geq 0 \longrightarrow a^*$

$a^n b^m | n, m \geq 0 \longrightarrow a^* b^*$

$a^n b^n \mid n \geq 0$ Cannot build RE.

$ww^R \mid w \in \{a,b\}^*$ No RE, as it is not having memory to hold no. of count of a or, to hold to half string and compare with another half

If L is a regular language then there exist K, which is $K \geq 1$, K = Pumping length

Then the string w, which belongs to L, is split into x, y, z such that length of $|xy| \leq K$, $y \neq \varepsilon$ and $xy^i z \in L \; \forall \; i > 0$

It is a negative test.

1) Show that Language $L = a^n b^n$ such that $n \geq 0$ is not regular.

$L = \{ a^n b^n : n \geq 0 \}$ is not regular }

Assume the given language is Regular
Consider the string $a^P b^P$ where $P > m$.
Let $P = 7$

(case i) $w = \underset{x}{aaa}\underset{y}{a}|\underset{y}{aaa}\underset{z}{|bbbbbbb}|$

$i = 2$.

$xy^2z \rightarrow aaaa\underbrace{aaa}_{1}\underbrace{aaa}_{2}bbbbbbb \in L$

$10 \neq 7 \Rightarrow \notin L$.

(case ii)

$w = aaaaaaa\underset{x}{|}\underset{y}{b}\underset{z}{|bbbbbbb}$

$i = 2$.

$xy^2z = aaaaaaa\underset{w_1}{bb}\underset{w_2}{b}bbbbbbb \in L$.

$8 \neq 7 \notin L$

2) Show that $L = \{ a^P / P$ is a prime no $\}$ is not regular.

$a^x \longrightarrow a^a|\underset{y^i}{a^b}|a^c \qquad x = a+b+c$

i

28/9/17

Kleen's Theorem - part II.

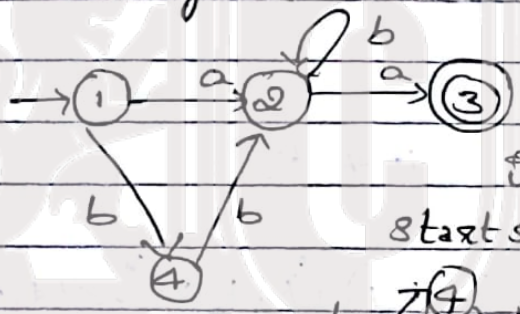
It says that we can built regular expressions from the given finite state machines.

$$R'(P,Q) = R(P,Q) \cup R(P, rip) \cdot (R(rip, rip))^* \cdot R(rip, Q)$$

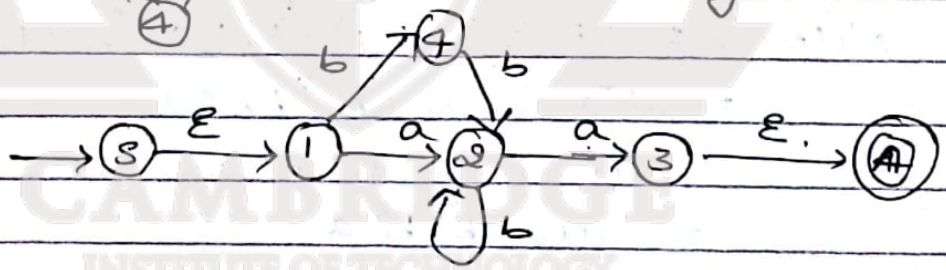
$R(P,Q)$ — direct transistor

$R(P, rip) \cdot (R(rip, rip))^* \cdot R(rip, Q)$ — Indirect transition via intermediate state 'rip'.

(1) Find the regular expression from the given FSM using Kleen's Theorem.



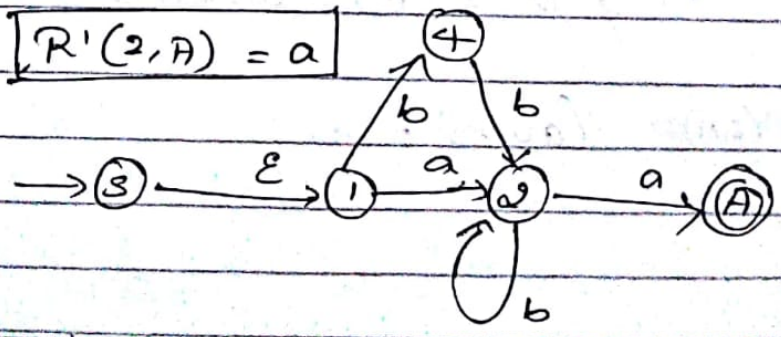
introduce a new start state and final state.



Let rip = 3 2-3-A

$$R'(2,A) = R(2,A) \cup R(2,3) \cdot (R(3,3))^* \cdot R(3,A)$$
$$= \phi \cup a \cdot (\phi)^* \cdot \varepsilon$$

$$\boxed{R'(2,A) = a}$$



let $x_{ip} = 2$ — 1-2-A, 4-2-A

$R'(1,A) = R(1,A) \cup R(1,2)\,(R(2,2))^* R(2,A)$

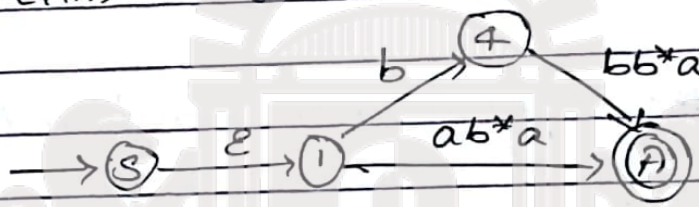
$= \phi \cup a.(b)^*.a$

$R'(1,A) = ab^*a.$

$R'(4,A) = R(4,A) \cup R(4,2)\,(R(2,2))^* R(2,A)$

$= \phi \cup bb^*.a$

$R'(4,A) = bb^*a.$



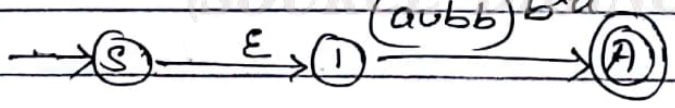
let $x_{ip} = 4$ — 1-4-A.

$R'(1,A) = R(1,A) \cup R(1,4)\,(R(4,4))^*.R(4,A)$

$= ab^*a \cup b.(\phi)^* bb^*a$

$= ab^*a \cup bbb^*a.$

$= (a \cup bb)\,b^*a$

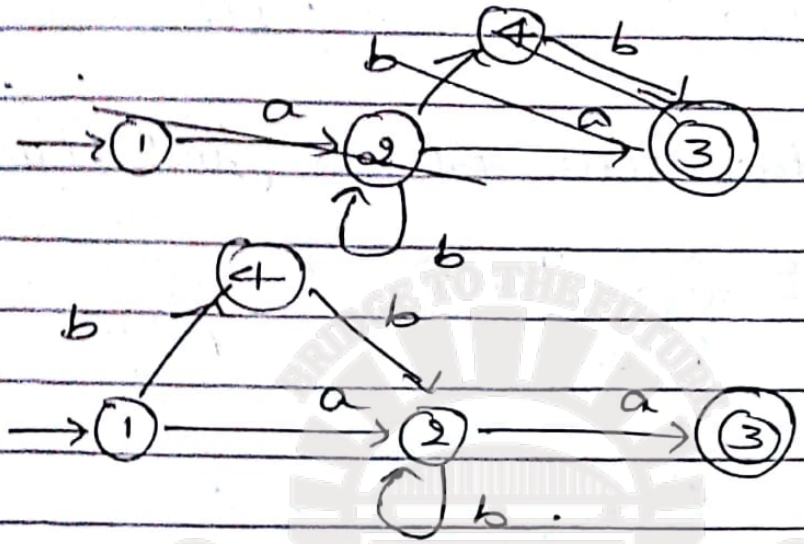


$R'(S,A) = R(S,A) \cup R(S,1)\,(R(1,1))^*.R(1,A)$

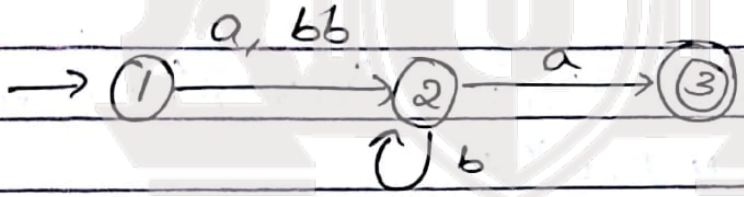
$= \phi \cup \varepsilon(\phi)^*.(a \cup bb)\,b^*a$

$R'(S,A) = .(a \cup bb)\,b^*a.$

using state elimination method.

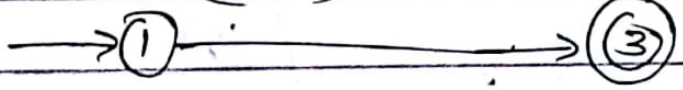


Introduce eliminate (4),

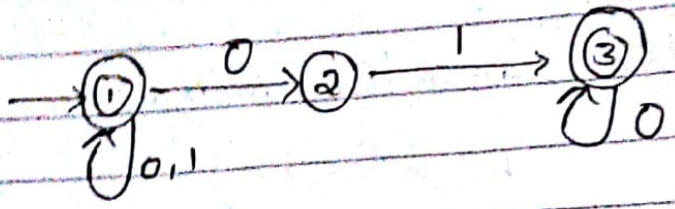


eliminate (2),

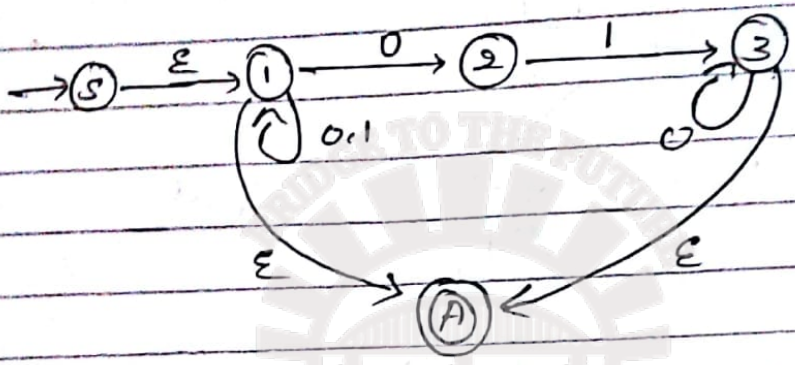
$(a \vee bb)(b^*a)$.



(3)

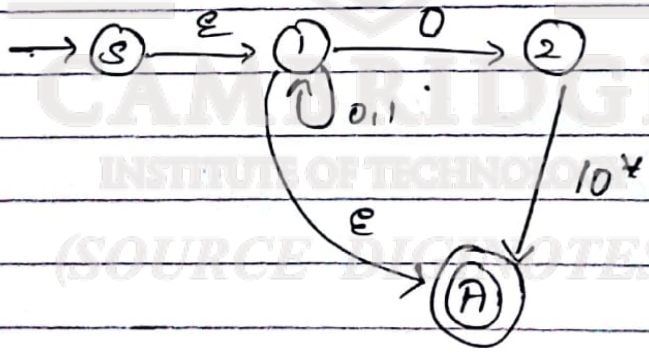


Introduce start state and final state.



let rip state be 3 2-3-A

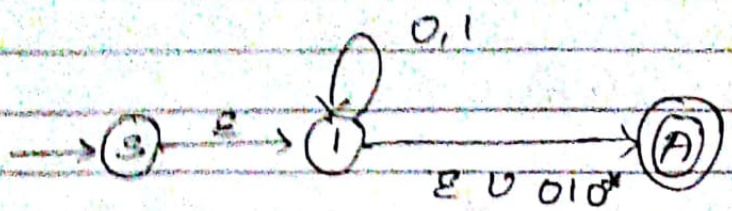
$$R'(2,A) = R(2,A) \cup R(2,3)(R(3,3))^* \cdot R(3,A)$$
$$= \phi \cup 1(0)^* \varepsilon$$
$$R'(2,A) = 1(0)^*$$



rip = 2 1-2-A

$$R'(1,A) = R(1,A) \cup R(1,2)(R(2,2))^* R(2,A)$$
$$= \varepsilon \cup 0(\phi)^* 10^*.$$

$$R'(1,A) = \varepsilon \cup 010^*$$

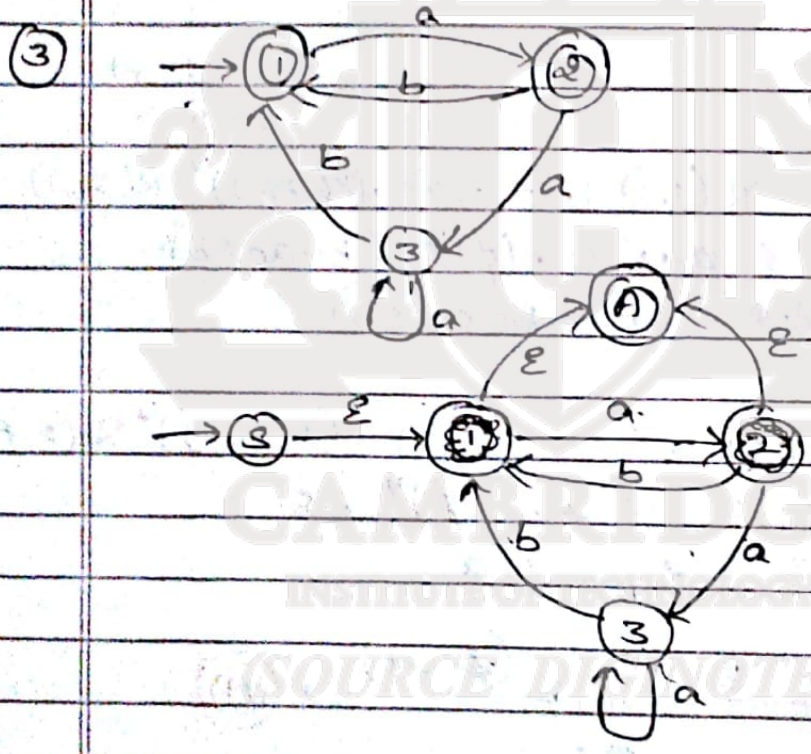


Elp = 1 S - 1 - A

$R'(S,A) = R(S,A) \cup R(S,1)(R(1,1))^* R(1,A)$.

$= \phi \cup \varepsilon (0 \cup 1)^* \varepsilon \cup 010^*$.

$\boxed{R'(S,A) = (0 \cup 1)^* (\varepsilon \cup 010^*)}$



Elp = 3 2 - 3 - 1

$R'(2,1) = R(2,1) \cup R(2,3).(R(3,3))^* R(3,1)$

$= a2^*b \cup$

Elp = 2 1 - 2 - A

R'(1,A) 3 - 2 - A

R'(1,3)

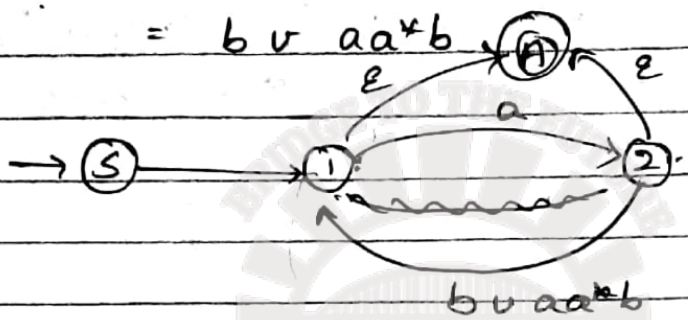
~~$R'(1,A) = R(1,A) \cup \epsilon$~~

let $x_{rip} = 3$, then $2-3-1$

$R'(2,1) = R(2,1) \cup R(2,3) \cdot (R(3,3))^* \cdot R(3,1)$

$= b \cup a \cdot a^* \cdot b$

$= b \cup aa^*b$



$x_{rip} = 2$ $1-2-A$ $1-2-1$

$R'(1,1) = R(1,1) \cup R(1,2) \cdot (R(2,2))^* \cdot R(2,1)$

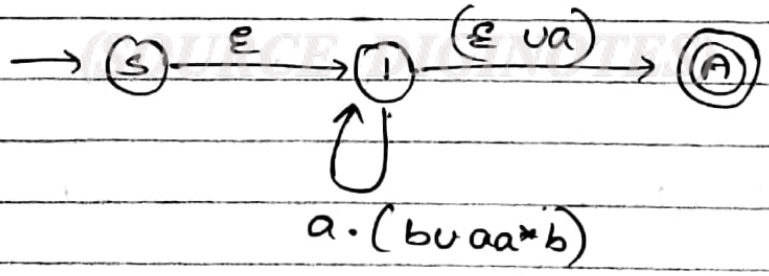
$= \phi \cup a \cdot (\phi)^* (b \cup aa^*b)$

$= a \cdot (b \cup aa^*b)$

$R'(1,A) = R(1,A) \cup R(1,2) \cdot (R(2,2))^* R(2,A)$

$= \epsilon \cup a \cdot (\phi)^* \cdot \epsilon$

$= \epsilon \cup a$

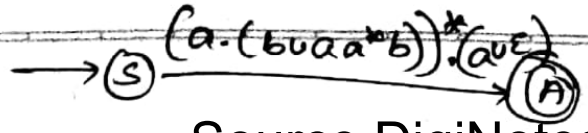


$x_{rip} = 1$ $S-1-A$

$R'(S,A) = R(S,A) \cup R(S,1) \cdot (R(1,1))^* \cdot R(1,A)$

$= \phi \cup \epsilon \cdot (a \cdot (b \cup aa^*b))^* \cdot (\epsilon \cup a)$

$= (a \cdot (b \cup aa^*b))^* \cdot (a \cup \epsilon)$



Simplification of Regular Expression

1. union is Commutative
   $a \cup b = b \cup a$
2. union is associative
   $(a \cup b) \cup c = a \cup (b \cup c)$
3. ε is identify for concatenation $[\varepsilon . a = a . \varepsilon = a]$
4. ϕ is " " union $\phi \cup a = a \cup \phi = a$
5. ϕ is zero for concatenation $\phi . a = a . \phi = \phi$
6. $\phi^* = \varepsilon$
7. $\varepsilon^* = \varepsilon$
8. $(a \cup b)^* = (a^* b^*)^*$
9. $(a^*)^* = a^*$

10. left distributive law :- $\gamma(\alpha \cup \beta) = \gamma\alpha \cup \gamma\beta$
11. Right distributive law :- $(\alpha \cup \beta)\gamma = \alpha\gamma \cup \beta\gamma$
12. Given 2 sets A and sets B, $A \subseteq B$ then
    ~~$L(A) \cup L($~~ $A \cup B = B$