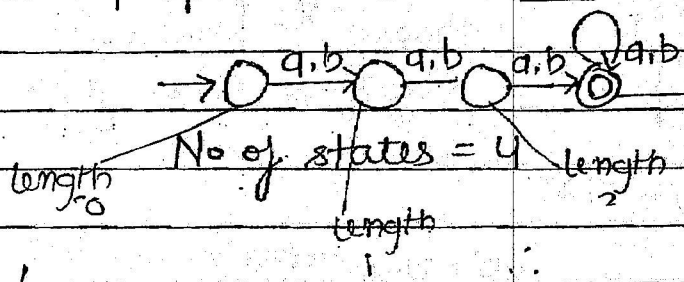
Page No.
Date : / /

Length Machine-

21. $L = \{ w \mid |w| \geq 3 \}$ Re - $(a+b)^3 (a+b)^*$

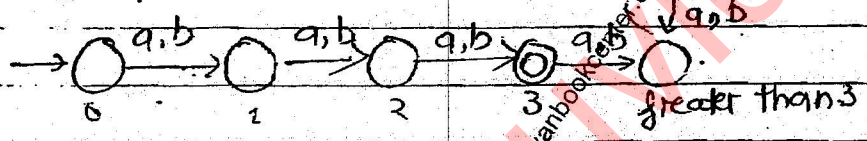
(at least 3 length) (for more than 3 length)



No of states = 4

{ for n length machine, No of states = n+1 }

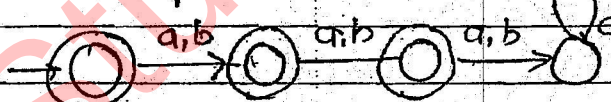
22. $L = \{ w \mid |w| \neq 3 \}$ RE - $(a+b)(a+b)(a+b)$



No of states = 5

{ No of states in DFA = n+2
No of states in NFA = n+1 }

23. $L = \{ w \mid |w| \leq 2 \}$ RE - $\epsilon + (a+b) + (a+b)(a+b)$



No of state in NFA = n+1
DFA = n+2

for $|w| \leq 2$

$RF = \epsilon + (a+b) + (a+b)^2 + (a+b)^3$

$= \epsilon + (a+b) + (a+b)^2 (\epsilon + (a+b))$

$= (\epsilon + (a+b)(\epsilon + (a+b))^2)$

$= (\epsilon + a + b)$

$= (\epsilon + a + b)^3$

(atmost 3 bit)

EXCELLENT

If $((0+1)^*)$ is given ie it is the string of language is having $w \bmod 100 = 0$

ie it will accept the string mu ble of 100 (in length)

Page No.
Date: / /

If $(\varepsilon+a+b)^{100}$ – atmost 100 bits

as $\varepsilon ab\ \varepsilon a,b \quad \cdots \quad \varepsilon$

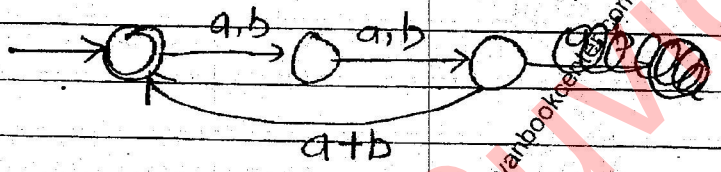
If $(\varepsilon+a+b)^{100}(a+b)^*$ – atleast 100 bits

If ~~$\varepsilon+$~~ $(a+b)^{100}$ – exactly 100 bits

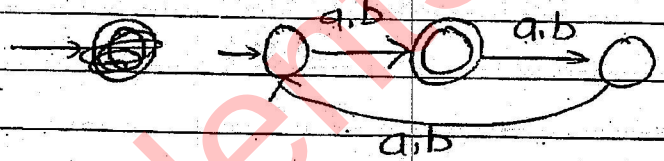
24. $L = \{w \mid |w| \bmod 3 = 0\}$

$RE = ((a+b)(a+b)(a+b))^*$ ie 3x

↳ as it must have 3 length



for 1 as residue 3x+1



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

for 3x+2 : $(a+b)((a+b)(a+b)(a+b))^*(a+b)$

Ex- $(aa+bb)^*$

I- Every string created is of even length ✓

II- Every string is even length string ✓

~~$\varepsilon+$~~

III- It will generate all the string of even length

↳ false

$(aa+ab+ba+bb)^*$

it will not create ab ba

Page No.
Date : / /

Ex - $((0+1)^{100})^* (\epsilon+0+1)^{23}$

$L = \{w \mid w \bmod 100 \leq 23\}$

if $\left((0+1)^{100}\right)^* (0+1)^{23} \Rightarrow L = \{w \mid w \bmod 100 = 23\}$

as for $L = \{w \mid w \bmod 100 \geq 23\}$

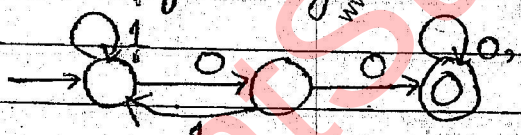
~~$(0+100)$~~ $\left((0+1)^{100}\right)^* \left((0+1)^{23} + (0+1)^{24} + \cdots + (0+1)^{99}\right)$

$(0+1)^{100})^* (0+1)^{23} (\epsilon + (0+1) + (0+1)^2 + \cdots (0+1)^{76})$

$(0+1)^{100})^* (0+1)^{23} (\epsilon+0+1)^{76}$

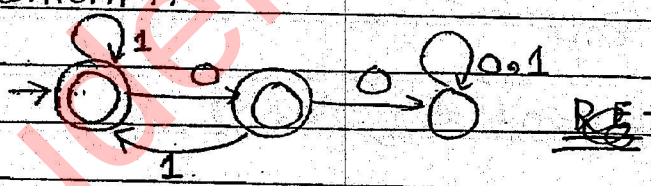
$\Sigma = \{0,1\}$

25) All the binary string with nd two consecutive zeros



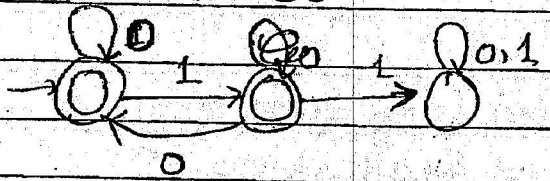
(for with consecutive zero)

Now complement it



RE

26) No two consecutive 1's



EXCELLENT

~~0+10+~~

Page No.
Date : / /

Regular expression -

* No two consecutive zero -

1. take 0 blocked with 1

~~(01+1)~~ $(01+1)^*$

~~(10+01+1~~ and then put 1 zero

$RE = (01+1)^*(\epsilon+0) = (\text{~~10+1)*(ε~~})$

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

$(\epsilon+0)$ is placed so that string can end with zero as well

* No two consecutive one.

1. block 1 with zero

$RE = (0+10)^*(\epsilon+1)$

It can also be written ~~(0+01)~~ as

$\{(\epsilon+1)(0+01)^*\}$ // string can start with 1 as well

$(01+1)^*$ - Not two consecutive zero but string ending with z

$(0+01)^*(0+011)^*$ → No two consecutive zero and string with 1 & with no null

~~(0+01)~~ = Every 1 0 is

$(0+10)^*$ = every 1 is followed by atleast 1 0.

26. No three consecutive '000's

$RE = (1+01)^*(\epsilon+0)$ (for no two consecutive zero)

~~(1+00~~

$RE = (1+0(1+01))^*(\epsilon+0+00)$

091

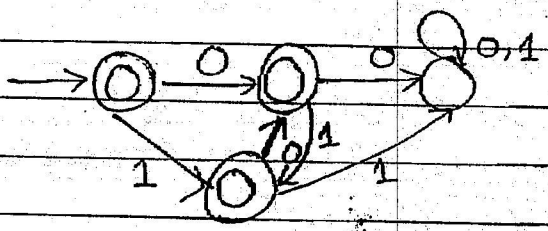
$RE = (\epsilon+0+00)(1+(1+01)0)^*(\text{~~ε~~})$

EXCELLENT

Page No.
Date : / /

Ex - Neither consecutive 0 nor consecutive 1 (two)

$(\epsilon+0)(101+01)^*(\epsilon+0)$



RE $(\epsilon+1)(010+0$

RE $(\epsilon+0)(101)^*(\epsilon+0)$

RE = $(\epsilon+0)(101)^*(\epsilon+0)$

RE = $(\epsilon+1)(010)^*(\epsilon+1)$

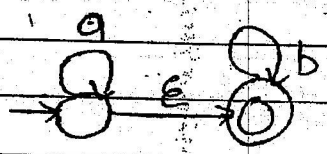
$(\epsilon+0)$ is added as $(101)^*$ will only generate the string starting & as well as ending with 1.

Design for Regular expression-

29. If a Regular expression is given directly design NFA it is easy to convert design NFA by given RE.

29. a^*b^*

NFA is:



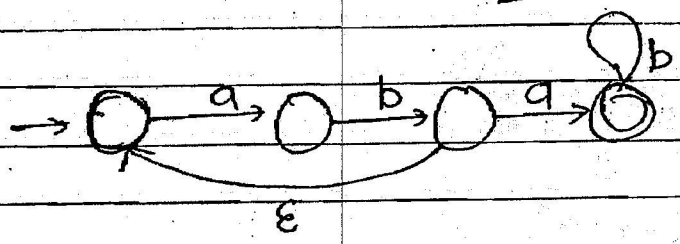
Minimum state = 2

EXCELLENT

Page No.
Date : / /

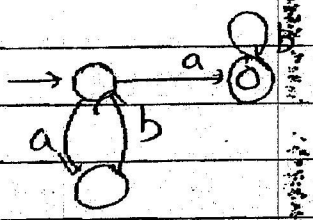
30) $(ab)^*ab^*$

(*) when it came go on serial ie change of sto



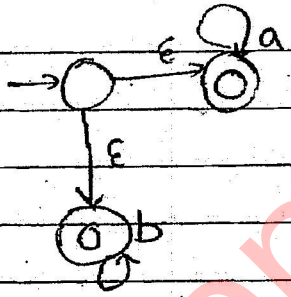
+ - parallel branch
• concatenated.

its minimal NFA is—



31) - $a^* + b^*$

it (+) just do parallel branching use ε move to go on.



for Regular to DFA
1. convert into NFA
2. ~~convert~~ if already DFA, occurs
else if it is not DFA, 3 reason
1. choice are multiple
2. DC
3. Null Move.
2. if DC only, but that move into trap directly no need of algorithm.

EXCELLENT

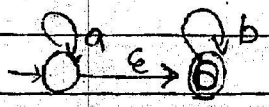
DFA to NFA conversion algorithm will give a unminimized DFA.

Page No.
Date : / /

- If Null Move exist, algorithm is applied but it take time so use shortcut

as Ex -

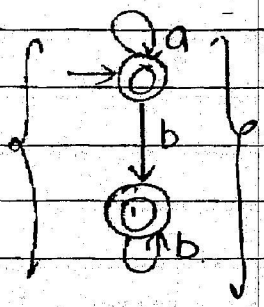
if a^*b^*



$\Downarrow$

$b^* = (\varepsilon + bb^*)$

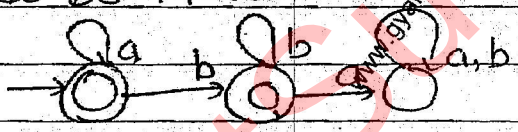
$\therefore a^*(\varepsilon + bb^*) = a^* + a^*bb^*$



Theorem :

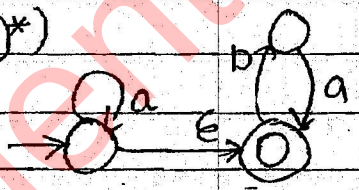
1. $r^* = (\varepsilon + rr^*)$

Now do DC Moves



Ex 2 $(a^*(ba)^*)$

NFA -

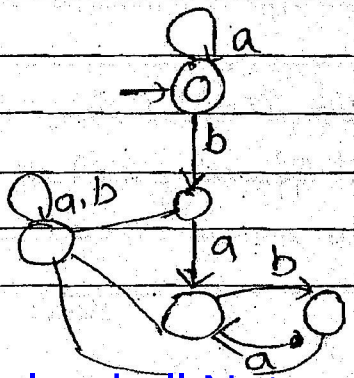


use like $r^* = \varepsilon + rr^*$

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

$\therefore (a^*(ba)^*) = a^*(\varepsilon + (ba)(ba)^*)$

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



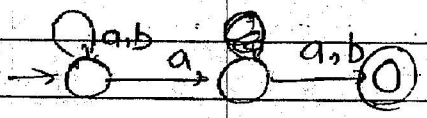
EXCELLENT

Page No.
Date : / /

• if choice comes, you can not directly remove them.

[it basically arise when setting is done in right side so if possible, avoid them.]

as $(a+b)^*a(a+b)$



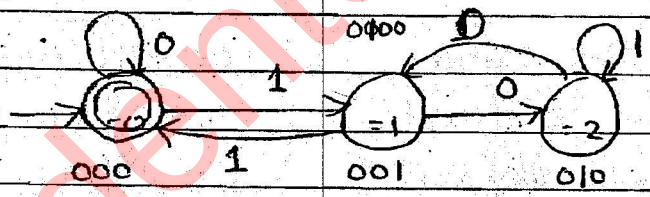
↳ You cannot remove directly the choice problem.

33 - $\{w \mid d(w) \bmod 3 = 0\}$ ie binary string of a decima value must be multiple of ~~three~~ three.

• Mod is always regular.

• since Mod 3, just you will have 3 states.

these three state has 0, 1, 2 as O/P (i.e residue)

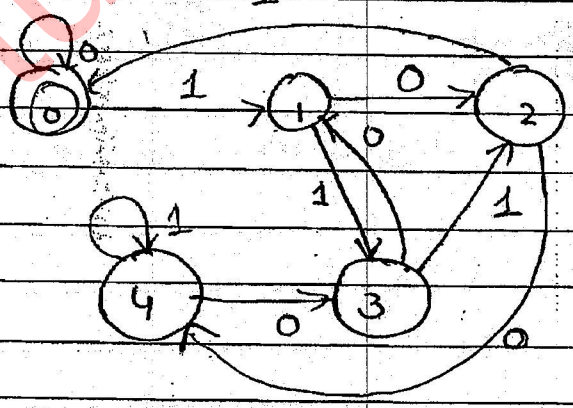


• if only 0000 come must give 0 only

• when 1st 1 com it go to 1 o/p st

• when 0 comes at 1 it became 2 and so on.

for mod 5



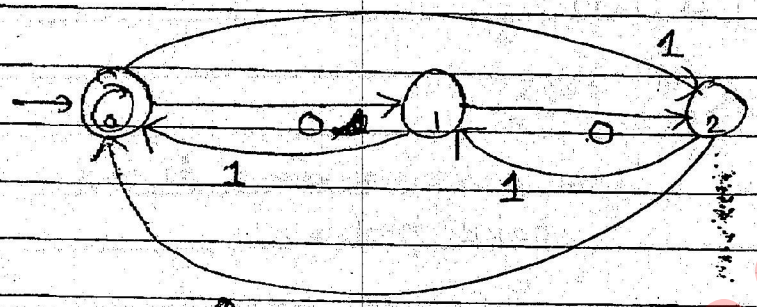
EXCELLENT

Page No.
Date : / /

a If you want to do $n_0(\omega) - n_1(\omega) = 2$ not a regular language

but

33 $L = \{ \omega \mid n_0(\omega) - n_1(\omega) \bmod 3 = 0 \}$



If only 0 comes
= 1
If only 1 comes
-1 mod 3
= 2
as 7 mod 3
7-6 = 1
-1 mod 3
-1 -(-3) = 2

just took some bits & proceed.

for mod n machine of any type,
only n state will exist

Language with more than one minimal string.

34. First two symbols are same — (00+11)(0+1)*
35 first ——— different — (10+01)(0+1)*
36 last ——— different — (0+1)*(10+01)
37. last ——— same — (0+1)*(11+00)
38. first ~~two~~ symbol are same & last ~~two~~ are also same
39. first & last symbol are different — 0(0+1)*0 + 1(0+1)*1 + 1 + 0

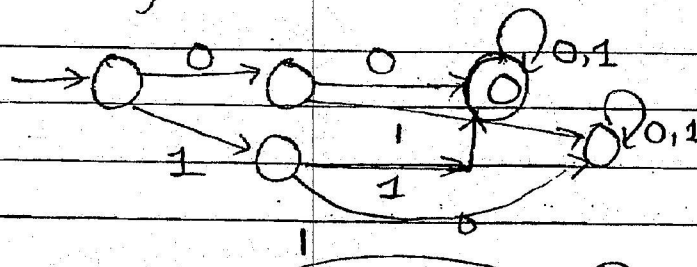
Min state in NFA = 4
——— DFA = 5

0(0+1)*1 + 1(0+1)*0

Page No.
Date : / /

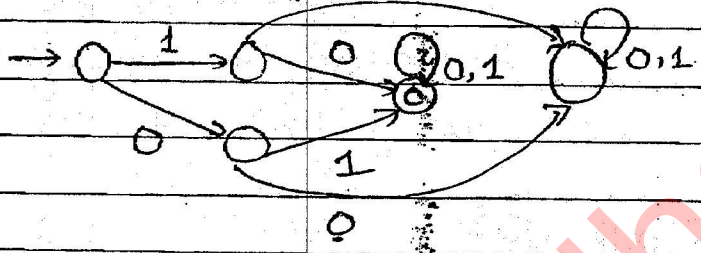
$\Sigma = \{0,1\}$

34. first two symbols are same



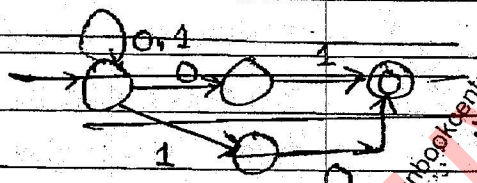
(DFA)

35 -



(DFA)

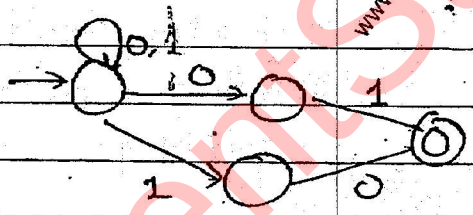
36.



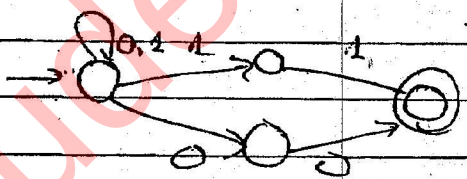
NFA

~~When~~

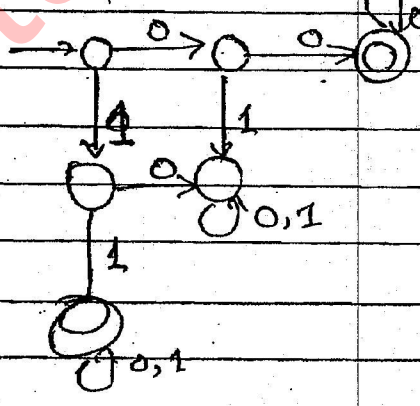
36.



37



for 34 -



⇒ then Merge bcoz both have same behave.

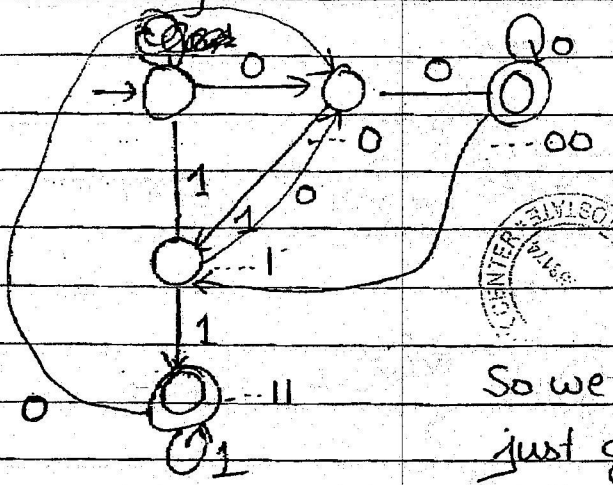
whenever you are doing for two minir string, in starting itself donot merge the final state, if those are permanen accept, then merge. otherwise leave ther i.e it is done so th no mistake can ever happen.

→ procedure

EXCELLENT

Page No.
Date : / /

3] Last two symbol are same



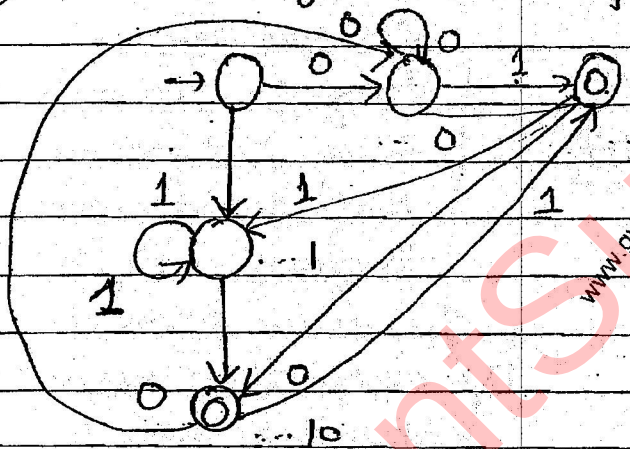
(00+11)(0

(0+1)*(00+11)

(0+1)*- ε +(00+1)(0+1)*

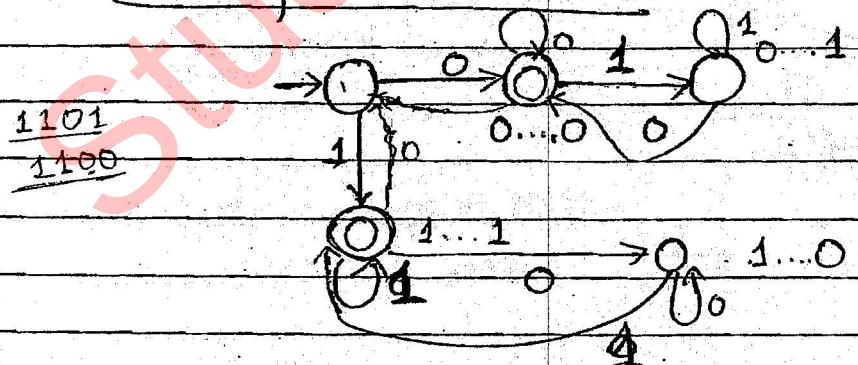
So we use the arrow filling ie just go with arrows

3] Last two symbol are diff.



- Initially Draw for minimal states. note down underneath
- then fillup the arrows
- if you not down under neath no chances of mistake

38 - last & first are same-



EXCELLENT

Page No.
Date : / /

Regular Expressions-

- Arideen's Theorem - for conversion of machine to RE.

1. Conversion of Machine to RE.
2. Equivalance of two RE & properties of RE

Ex - $L_1 = \{\epsilon\}$ $L_2 = \{a\}$, $L_3 = \phi$

$L_1 \cup L_2 \cdot L_3^*$

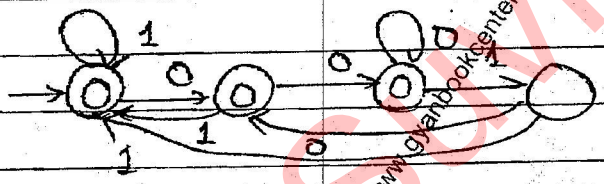
$\epsilon + a \cdot \phi^*$

$\epsilon + a \cdot \epsilon$

$= \epsilon(\epsilon + a)$

$\{\phi^* = \epsilon\}$

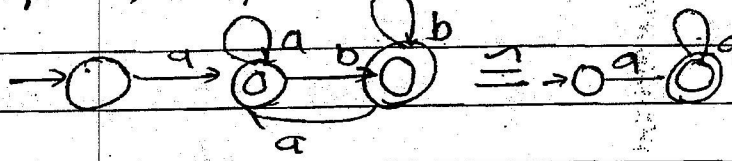
3. Arideen's Theorem

Ques 

not end with 001

1. Machine to RE -

1. You should not bother whether it is Nfa or Dfa.
2. Before you convert it, simplify the machine.
   1. Remove any trap state
   2. Remove any non-reachable state
   3. In case of multiple final state, if they can be collapsed, collapsed them.



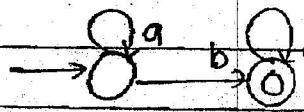
   4. Remove all states which do not serves anything for final state.
3. If you have final state which can not be collapsed, write re for each final state & add them

EXCELLENT

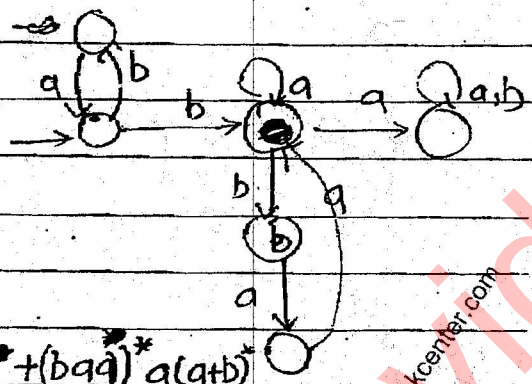
Page No.
Date : / /

5) try to write r.e for one final state into term of another.

6) the r.e for final state is the longest path from initial to final state



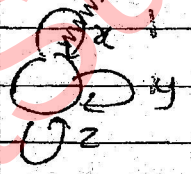
$a^*b(a+b)^*$



$(ab)^* b(a^* + (baa)^* a(a+b)^*$

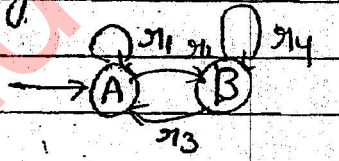
$= (ab)^* b(a + baa)^* a(a+b)^*$

7. When you have



$\Rightarrow (x+y+z)^*$

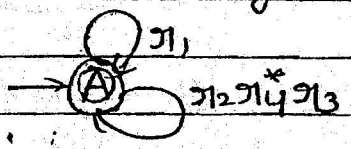
8. Try to reduce the machine to 2 state machine & can directly solve it.



r_1, r_2, r_3, r_4 = any expression

Let A be the final state,

$r.e = r_A$ reduce to a single state



$\therefore$

$r_e = (r_1 + r_2 r_4^* r_3)^*$

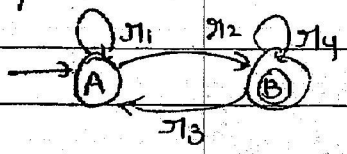
EXCELLENT

Page No.
Date : / /

To simplify the machine -

1. In beginning, try to remove not initial & final states of those where min actions are done.

if B is the final state.



It will have two soln has it has loop. You can generate answ for left arrow & for right arrow known as left & right resolution

$B \cdot r_B = r_4^* r_1^* r_2 r_4^* r_3 (r_1^* r_2 r_4^*)$

~~kno~~

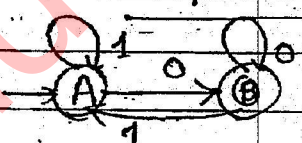
from A to B & B to A — A to B

LR, $r_B = (r_1 + r_2 r_4^* r_3)^* r_2 r_4^*$

RR, $r_B = r_1^* r_2 (r_3 r_1^* r_2 + r_4^*)$

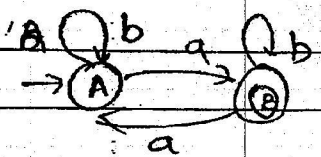
A to B — B to A — B to A

in left, initially solve the left part by A when can have r_1 or $r_2 r_4^* r_3$

for 

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

RR = ~~$(00^*1$~~ $1^*0(0 + 11^*0)^*$

for 

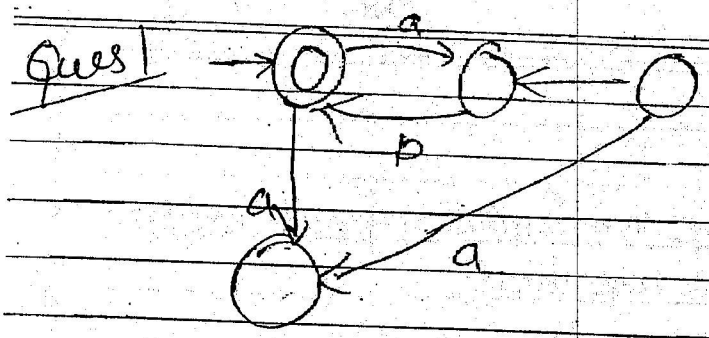
~~As~~ LR = $(b + baab^*a)ab^*$

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

EXCELLENT

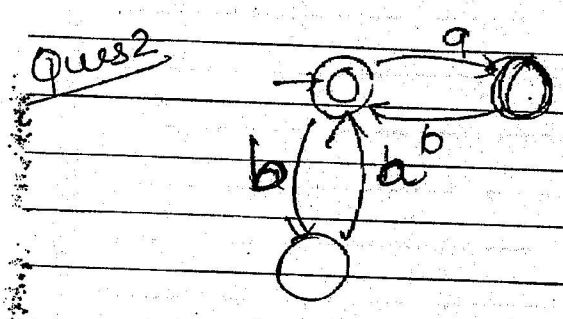
Page No.
Date : / /

Ques 1



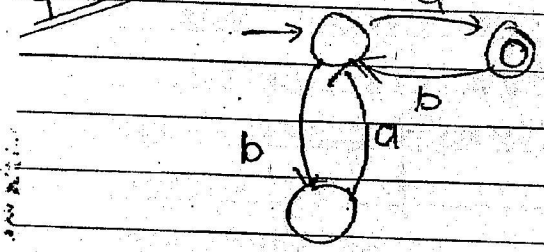
R.E = $(ab)^*$

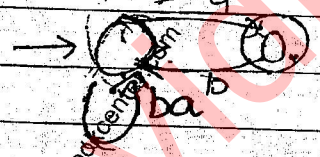
Ques 2



RE = $(ab + ba)^*$

Ques 3

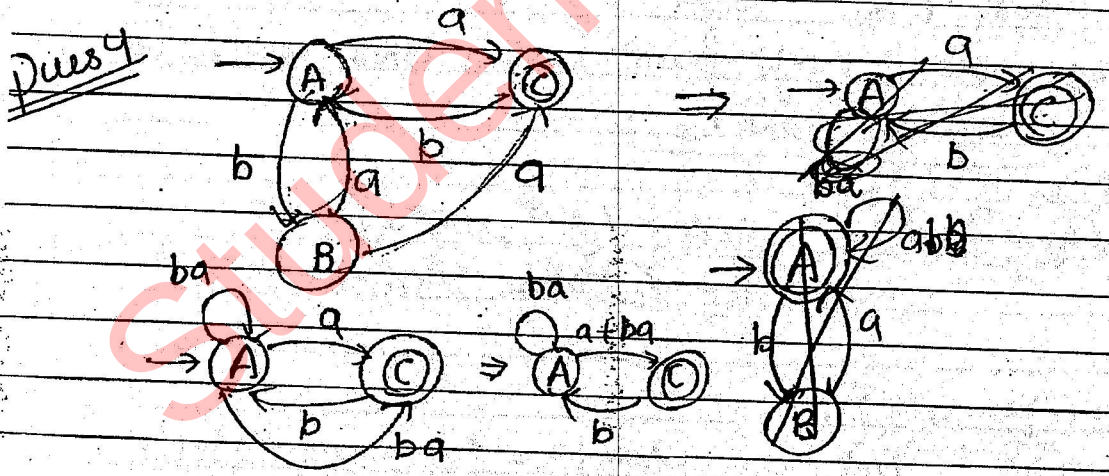




LR = $ba + (ab)^*a$

RR = $(ba)^*a\,(b(ba)^*a)^*$

Ques 4



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

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

EXCELLENT

= LR = Write A & then B followed b

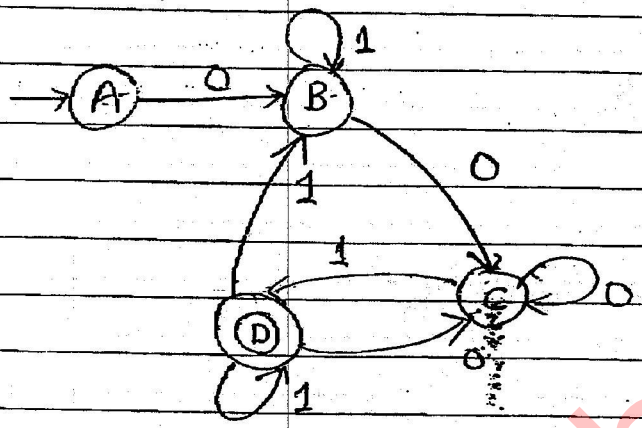
RR = Write A followed by B

Page No.

Date : / /

LR means B in terms of A

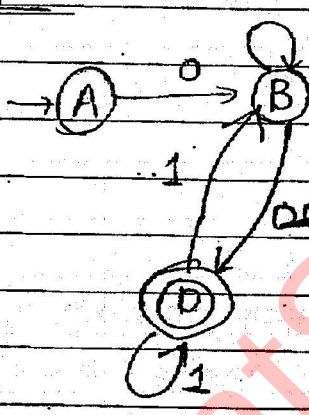
Ques -



~~(1+0~~

0(0+11*0)1

Delete C



~~00*(10)~~ 0(0+11*0)*1 = ε (let)

D

LR - 0[(1+ε1*1)* ε1*]

RR - 0[ε1*ε(1+11*ε)*]

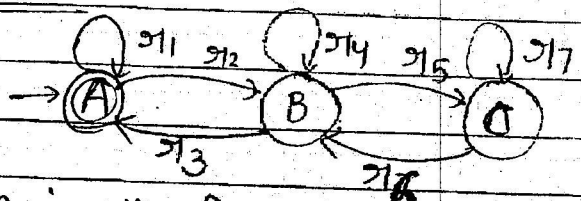
LR = 0[(1+(0(0+11*0)*1*1))* + (0(0+11*0)*11*]

So many soln are possible

EXCELLENT

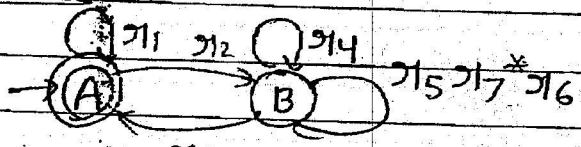
Page No.
Date : / /

General -



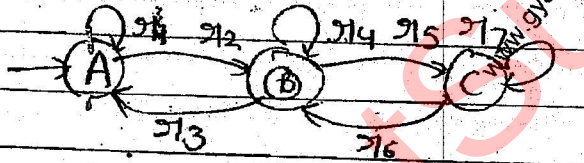
Let A is the final state

1. Delete C :

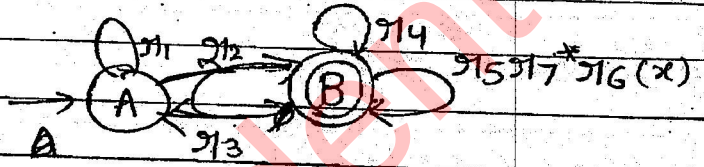


$$rA = \left(r_1 + r_2 (r_4 + r_5 r_7^* r_6)^* r_3\right)^*$$

Let B is the final state.



For best answer resolve the loop at B only



$x = r_4 + r_5 r_7^* r_6$

LR ~~$(r_1 + r_2 (r_4 + $~~

$LR = (r_1 + r_2 x^* r_3)^* r_2 x^*$ (A get resolved.

$RR = r_1^* r_2 (x + r_3 r_1^* r_2)^*$ (when resolve is done at B only)

(Here RR is better)

∴ final answer = $r_1^* r_2 ((r_4 + r_5 r_7^* r_6) + r_3 r_1^* r_2)^*$

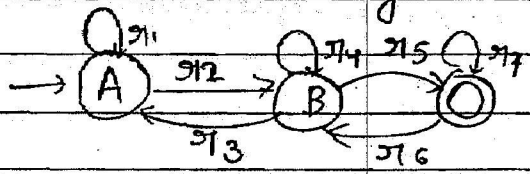
(You can also get it by directly taking loop only)

EXCELLENT

Page No.
Date : / /

for C as final state, you have four solutions as for loop

so use best answer of B & then followed by C.



$r_C = (RR \text{ of } B)\, r_5 r_7^*$

Properties of Regular Expression-

If r & s are two regular expression on given alphabet Σ

1. Commutative - $r + s = s + r$

$r \cdot s \neq s \cdot r$

Ex : $a + b = b + a$

$a \cdot b \neq b \cdot a$

2. Associativity - $r + (s + t) = (r + s) + t$

$r \cdot (s \cdot t) = (r \cdot s) \cdot t$

Ex - I. $a + (b^*c^* + d^*)$

II $d^* + (a + c^*b^*)$ I, III are equivalent.

III $d^*(a + b^*c^*)$

3. Distributive - $r \cdot (s + t) = r \cdot s + r \cdot t$

$r + (st) \neq (r + s) \cdot (r + t)$

Ex I = $a(b^* + c^*)$

II $ab^* + ac^*$ } equivalent

III $ab^* + c^*a$

EXCELLENT

1. $a + \varepsilon = \{a, \varepsilon\}$
2. $a + \phi = a$

Page No.
Date : / /

$(s + t)r = sr + tr$

Ex - I $(ab^*) + c^*$
II $(a + b^*)(b^* + c^*)$ > Not equivalent

4. Identity - 1. $r + \phi = \phi + r = r$ 3. $r \cdot \phi = \phi$
2. $r \cdot \varepsilon = \varepsilon \cdot r \neq \varepsilon$

Ex - $a^* = a^* \cdot \varepsilon + \phi = a^* + \phi$

5. properties of * - 1) $\varepsilon^* = \varepsilon$
3) $\varepsilon^+ = \varepsilon$
3) $\phi^* = \varepsilon$
4) $\phi^+ = \phi$

1. $\varepsilon^* = \varepsilon^+$
2. $\phi^* \neq \phi^+$

5. $(r^*)^* = r^*$

6) $r^* \cdot r^* = r^*$
7) $r^* + r^* = r^*$
8) $r^* r^+ = r^+ = r r^*$
9) $(r^*)^+ = r^*$
10) $(r^+)^* = r^*$

6) Union & $r + r = r$
Concatenation - $r \cdot r = r^2$

7 $(r + s)^* = (r^* + s^*)^*$

EXCELLENT

Page No.
Date : / /

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

$(a+b)^*$ as $\{\epsilon+b+b^2\ldots\}$

$\Downarrow$

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

$=(a+b)$

Go

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

$(b, ab, bab, \ldots + b+\epsilon+bb)^*$

↓ No we not getting $(a+b)^*$

$(r+s)^* = (r^*+s^*)^* = (r^*s^*)^*$

↓ because we are getting r & s

to check $(\text{——})^*$ is $(a+b)^*$ check whether you are getting ~~r~~ a & b seperately or not-

1) $(a^*b^*)^* = (b^*a^*)^*$

2) $\epsilon + rr^* = r^*$

3) if r, s, t are r.e.,

12) $p(q^*p)^* = (pq^*)^*p$

~~$p(q^*r)^*$~~ 14) $p(qp)^* = (pq)^*p$

Proof → (A) p (B) q

LR- $(pq)^*p$ } equal

RR $p(qp)^*$

Manual- $p(pq)^n = p(pq)^n$

$p(qp)^n = (pq)^n p$

$pqpqpqp = pqpqpqp$

$n=3$

$(pq)^3 p = p(qp)^3$ EXCELLENT

Page No.
Date : / /

Ex I) $(ab)^*(c^*(ab)^*)^*$ (p q p)
II) $((ab)^*c^*)^*(ab)^*$ (p q p)
} equal

I) $(ab)^*(c^*(ab)^*)^*$
II) $(ab)c^*)^*(ab)^*$
} not equal

15. $(r+s)^*r^* = (r+s)^*$
$(r+s)^*s^* = (r+s)^*$

$(r+s)^*(\epsilon + r + rr + \dots) = (r+s)^*$ Hence proved

Ex - $(a+b)^*b^*a = (a+b)^*a$

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

Arden theorem - if $r = q + rp$
then $r = qp^*$
($r = r$ is dependent on r if there is cycle or loop; p cannot contain ϵ)

If $X = Xz + XTC^*$
$X = YZ(TC^*)^*$

EXCELLENT

Page No.
Date : / /

if eqn is given as $r = q + rp$ → Left Recursion

$\{r = qp^*\}$

but if equn is $r = q + pr$

$\{r = p^*q\}$

as if $r = q + rp$

$= q + qp^*p = q(\varepsilon + p^*p) = qp^*$

if

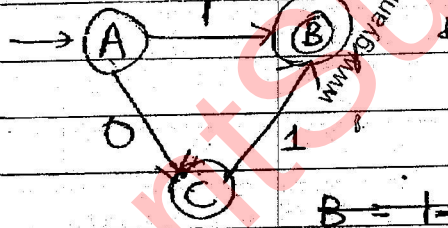
$r = q + pr$

$= q + pp^*q$

$= (\varepsilon + pp^*)q = p^*q$

but if ~~r~~

Ques -



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

~~B = 1+00 1+01~~

for applying Arden theorem

- each state will store incoming array with state followed by input

$A = \varepsilon$

$B = A1 + B1 + C1$

$C = A0$

$B = \varepsilon \cdot 1 + B1 + C1$

$C = 0$

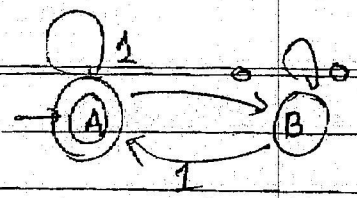
$B = 1 + 01 + B1$

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

EXCELLENT

Page No.
Date : / /

Ques-



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

$A = B1 + A1 + \epsilon$

$B = A0 + B0$

$B = A00^*$

$A = A00^*1 + A1 + \epsilon$

$A = \epsilon + A(00^*1 + 1)$

$A = \epsilon(00^*1 + 1)^*$

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

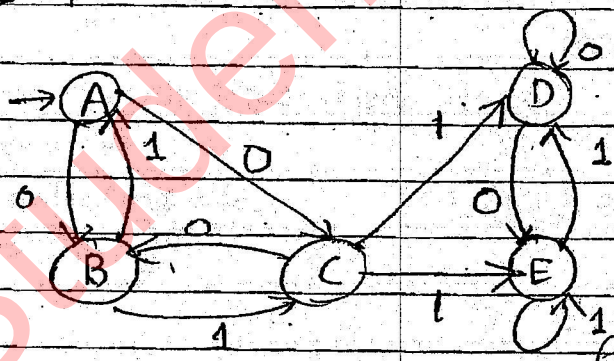
Ques → if the final state is D, L is L(D) if E is final state L is L(E)

(a) $L(D) = L(E)$

$L(D) \subseteq L(E)$

$L_E \subseteq L_D$

$L_D \neq L_E$



$D = C1 + D0 + E1$

$E = E1 + C\phi + D0$

$\therefore L_D = L_E$

You can find out the relation directly

EXCELLENT