

## Properties of Myhill Nerode Class-

1. ~~every myhill~~  $|E_i| \geq 1$  i.e each myhill-nerode ~~can~~ ~~have~~ have atleast one string.

2. Each myhill-nerode class must be unique.

3. ~~Each~~ two equivalence classes must have intersection as  $E_i \cap E_j = \phi$

(it must be disjoint)

because of min. dfa as in this, one string can perform single function only.

4. the Union of all the equivalence class must be  $\Sigma^*$  ( $\cup E_i = \Sigma^*$ ) because in NFA, no Dead Configuration are there.

Ques  $0^*1^*2^* \cap 1^*0^*2^*$

(a)  $2^*$

(b)  $(0^*+1^*)2^*$

(c)  $\phi$

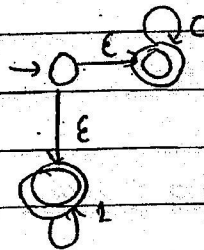
(d)  $(0+1+2)^*$

$$[0^*1^* \cap 1^*0^* = 0^*1^*]$$

$$0^*1^*2^* \cap 1^*0^*2^* = \phi$$

$0^*+1^*$

→ Min. NFA

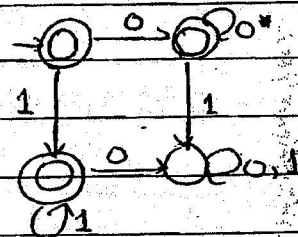


Min. DFA

$0^*+1^*$

$\epsilon + 00^* + \epsilon + 11^*$

$\epsilon + 00^* + 11^*$



To generate a regular language, we only need  $V \Rightarrow TV | \epsilon$  because for RG, there is dfa if grammar are constructed from dfa. It will only of the form  $V \Rightarrow TV | \epsilon$ .

• Unit production are luxury in Regular Grammar

The class of Grammar like

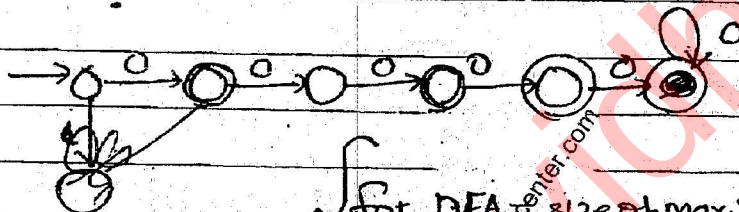
$$V \rightarrow T^* + \emptyset T^* V$$

$\emptyset$  and

$V \rightarrow TV \mid \epsilon$  have same powers.

## Unary Design

1.  $\{0, 000, 0000\}^*$

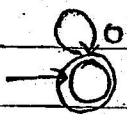


for DFA size of max string + 2

$$NFA = |W_{max}| + 1$$

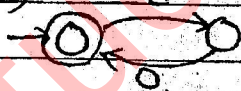
1 for start  
1 for trap

2.  $0^*$



Minimal DFA = 1

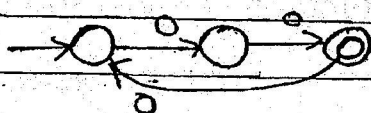
3.  $(00)^*$  - even no of zero



4.  $(00)^*0$  - odd no of zero



5.  $(000)^*00$  -  $n \bmod 3 = 2$



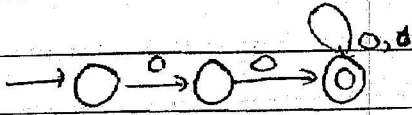
$$\equiv 0(000)^*0$$

$$\equiv 00(000)^*$$

EXCELLENT



4. atleast 2 zero's



Ques -  $\{0^{7n+4} \mid n \geq 0\}$  - how Many states in min. DFA?

Minimum state = 7 as mod 7 machine

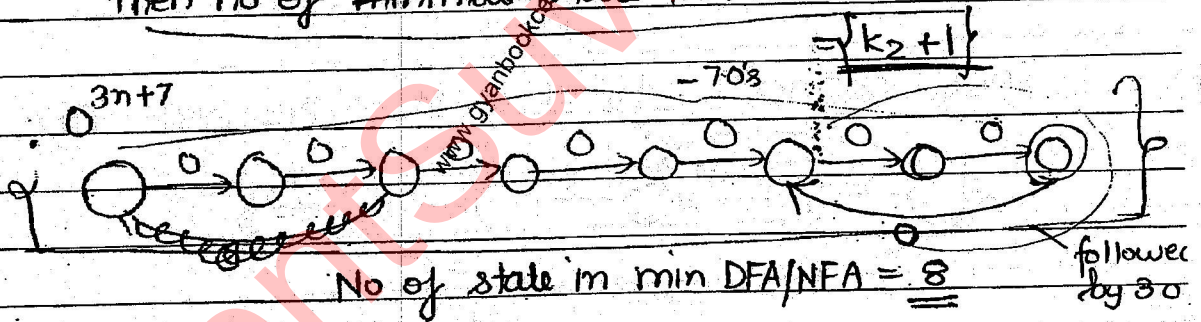
Ques -  $\{0^{k_1n+k_2} \mid n \geq 0\}$

Case I -  $k_1 > k_2$

then no of minimal state =  $k_1$  as it become mod  $k_1$  machine

II -  $k_2 \geq k_1$

then no of minimal state in M-DFA or NFA



Ques -  $\{0^{3n+7} \mid n \geq 0\}$

if  $n \geq 1$

$$\downarrow k_2 + 2k_1 + 1$$

Case II -  $k_2 > k_1$  &  $n \geq 0$

No of states =  $k_2 + k_1 + 1$  in case when  $n \geq 0$

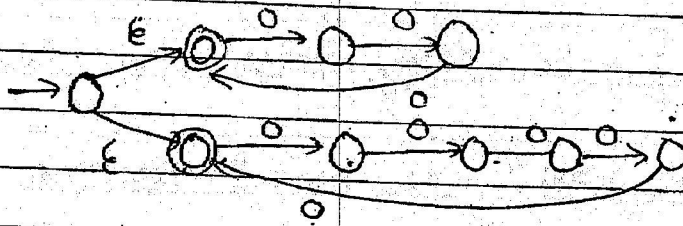
or  $\{ \text{minimum string} \} + 1 = \text{No of states}$

Here  $n=1$   $\rightarrow$  minimum string = 10

$\therefore$  No of state = 11



Ques  $(000)^* + (00000)^*$   
 $0^{3n} \cup 0^{5n}$



it is difficult to design DFA

No of states = in min NFA = 9  
 $= 3 + 5 + 1$

if  $(000)^* + (000000)^*$

A

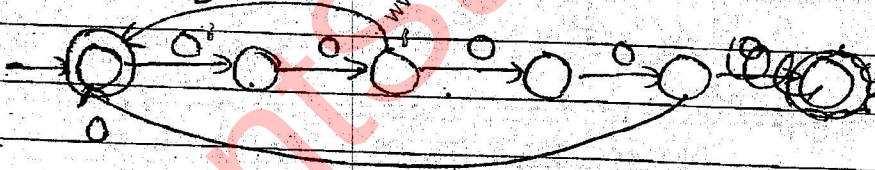
B

$B \subset A = (000)^*$  No of states = 3

Ques -

$(000 + 00000)^*$

No of state in Min state DFA



↓ Min. NFA = 5

- Take the design of larger string.
- Take to accumulate in it

(if the string are not multiple of each other smaller one will control the loop)

but if  $(000 + 000000)$

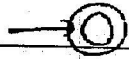
↳ so  $(000) = 3$  states are required.



Minimal DFA - Method-

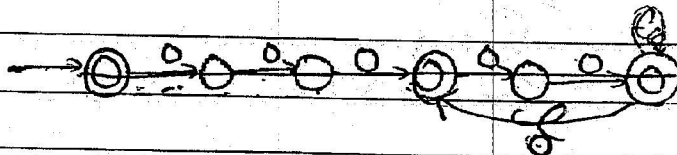
if  $\epsilon$  belong to language, accept it.

↳ goes on increasing the string accept it

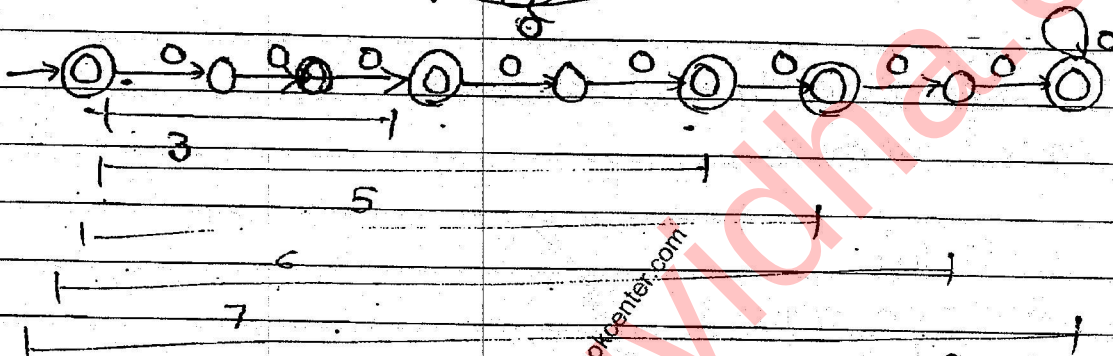


if belong to language

but it goes to infinite.

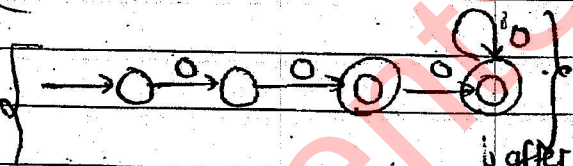


Minimum state = 9



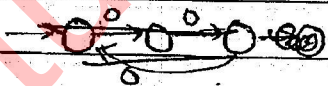
$(00+000)^*$

8 and so on  
after 8 no need to put more  
becoz each string is  
accepted

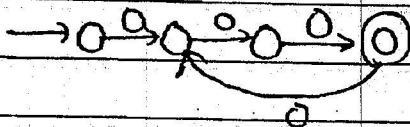


after 3 it accept everything.

Ques -  $\{0^{nk} \mid k \geq 0, n \text{ is fixed integer}\}$  no of states in min FA = ?



if  $n=3$  it  $0^{3k}$  mod machine  
with residue 0.

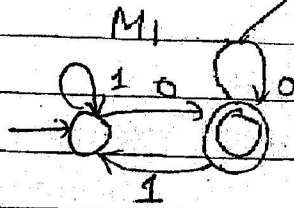


it would be  $n+1$

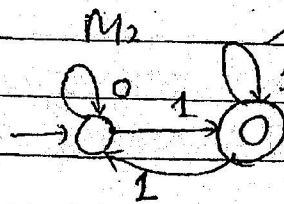
if it is  $k \geq 0$  then it would be  $n$   
↳ as it also accept null.

EXCELLENT

Ques -



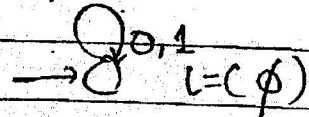
ending with 0



ending with 1

$$L(M_1) \cap L(M_2) = \emptyset$$

so  $M_1 \cap M_2$  require 1 state



89

- (a) as it can result to less breakdown.  
(b) more breakdown

(c)

(d)  $L = \{10101\}$

$$h(a) = 00 \quad h(b) = 11$$

$$h^{-1}(L) = \emptyset$$

$$h(h^{-1}(L)) = \emptyset$$

www.gyanbookcenter.com

90

$$L_1 \in (ab+ba)^*$$

$$h(0) = aa$$

$$h(1) = bb$$

$$h^{-1}(L_1) = \{ \emptyset \} = \{ 1 \} \text{ for } (*) \text{ as 1 need no breakdown}$$

$$h(L_2) = \{ bbaa + aabb \}^*$$

$$h^{-1}(L_3) = \{ 0^n 1^n \mid n \geq 0 \}$$

91



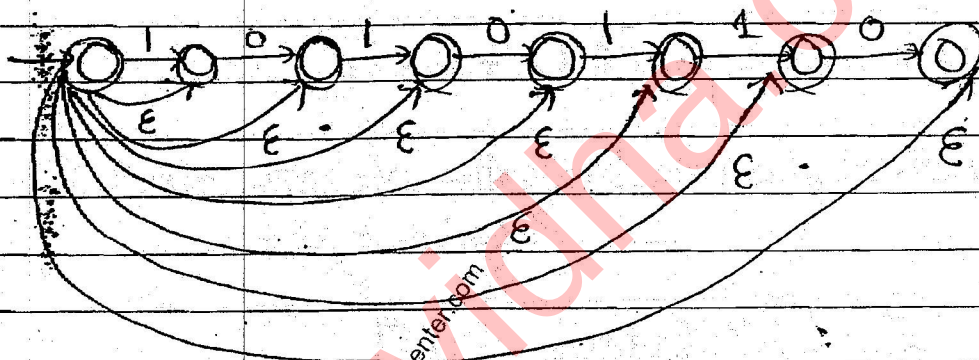
Linear  $f^n$  machine usually do not have trap

Page No.

Date: / /

Ques - Given a string "1010110" design NFA which accept every substring of this string.

$$\text{No of substring} = \frac{7(7+1)}{2} + 1 = 29$$



Here, firstly, all states are final state, so it can accept every prefix & then, add  $\epsilon$  move from initial state to starting & every state so that they can act as initial state.

• So  $n$  length string, no of states for accepting every string =  $n+1$

for  $n$  length string, no of state for accepting every prefix =  $n+1$

for AP find  $a$  &  $d$

$a + (n-1)d$  & then find the equation as

$$\text{If } a=3 \quad d=2$$

$$\left\{ \begin{array}{l} 2n+1 \\ 0 \end{array} \right\}$$

EXCELLENT

## Right Quotient & Left Quotient

Right Quotient -  $L_1/L_2$

Right Quotient of  $L_1$  with  $L_2$

$L_2/L_1$  - Left Quotient of  $L_1$  with  $L_2$

$$L_1/L_2 = \{ u \mid uv \in L_1 \text{ \& } v \in L_2 \}$$

$$L_2/L_1 = \{ v \mid uv \in L_1 \text{ \& } u \in L_2 \}$$

for Quotient, Closure properties are only applicable for Regular & CFL.

| $\varnothing$          | REGULAR | CFL |
|------------------------|---------|-----|
| INIT(L)                | ✓       | ✓   |
| $L_1/L_2$<br>$L_2/L_1$ | ✓       | ✓   |
| CYCLE(L)               | ✓       | ✓   |
| Suffix(L)              | ✓       | ✓   |

$L_1/L_2 \rightarrow$  if  $L_1 = \text{regular}$   
 $L_2 = \text{CFL}$

$\therefore L_1/L_2 \rightarrow \text{CFL}$

INIT(L)  $\leftarrow$  Initial or prefix of L  
SUFFIX(L)  $\rightarrow$  last of L

both are closed

CYCLE(L) = if  $L = \{010\}$  • put each rotation of  
CYCLE of  $L = \{0101, 1010\}$  given language

if  $L = \{1234\}$

$\therefore \text{CYCLE}(L) = \{1234, 2341, 3421, 4321\}$

EXCELLENT



1.  $L/\epsilon^* = \text{Prefix of } L$  i.e right quotient of  $L$  with  $\epsilon^*$ , will produce prefix of  $L$
- $\epsilon^*/L = \text{Suffix of } L$

Example  $L = \{ab, ba\}$

$\text{INIT of } L = \{\epsilon, a, ab, b, ba, baa\}$

$\text{INIT of } L$  contains all the prefix possible of the given string in language.

$|\text{INIT}(L)| = 6$

$\text{Suffix}(L) = \{\epsilon, a, b, ab, ba, a\}$

for Infinite language

- $\text{INIT}(L)$  will contain  $L, \epsilon$  (always)
- $\text{INIT}(L)$  will always contain  $\epsilon, \epsilon$

for  $a^*$   $\text{INIT}(L) = a^*$  take everything possible

| $L$                     | $\text{INIT}(L)$               | $\text{Suffix}(L)$      |
|-------------------------|--------------------------------|-------------------------|
| $a^*$                   | $a^*$                          | $a^*$                   |
| $(ab)^*$                | $(ab)^* + (ab)a$               | $(ab)^* + b(ab)^*$      |
| $a^*b^*$                | $a^*b^*$                       | $a^*b^*$                |
| $a^n b^n \mid n \geq 0$ | $a^m b^n \mid n \geq m \geq n$ | $a^m b^n \mid n \geq m$ |
| $na = nb$               | $(a+b)^*$                      | $(a+b)^*$               |

as it can have aba as prefix

$a^m b^n \mid n \geq m$

as it can have equal a & b as well a followed by b

as no equality is required in prefix



$$\sqrt{a^*/a = a^*}$$

$$\{ \epsilon, aa, a, aaaa, \dots \} / a = \{ a, \epsilon, aa, \dots \}$$

Page No.

Date: 1/1

Que  $L_1 = \{ 100, 0100, 11 \}$

$$L_2 = \{ 1, 01 \}$$

$L_1/L_2$  = take every string of  $L_1$  & divide it by  $L_2$ .

$$\begin{array}{ccccccc} 100 & \times & 100 & \times & 0100 & \times & 0100 & 11 & \checkmark & 011 \\ | & & 01 & & | & & 01 & | & & 01 \end{array}$$

$$L_1/L_2 = \{ 1 \}$$

$$L_2/L_1 = \frac{1}{100} \{ 00, 1 \} \text{ as } \frac{100}{100} = 1 \quad \frac{11}{11} = 1$$

$$\boxed{L_1/L_2 \neq L_2/L_1}$$

Que  $L_1/L_2 = 0^*1/1^*0$  expand  $L_2$

$$= 0^*1 \{ 0, 10, 100, \dots \}$$

$$= \phi$$

$$L_2/L_1 = 1^*0/0^*1 = \phi$$

$$L_1 = a^*ba^*$$

$$L_2 = ba^*$$

$$L_1/L_2 = a^*ba^* / \{ \epsilon, ba, baa, baaa, \dots \}$$

$$\frac{a^*ba^*}{a} = a^*ba^*$$

$$\frac{a^*ba^*}{ba^*} = a^* \text{ as } baa \text{ can never happen in end.}$$

$$L_1/L_2 = \{ a^*ba^* + a^* \}$$

EXCELLENT



$$\{ L/\epsilon = L \}$$

Page No.

Date: / /

$$L_1 = a^*b$$

$$L_2 = b^*$$

$$L_1/L_2 = a^*b/(e, b, bb, \dots)$$

$$\frac{a^*b}{\epsilon} = a^*b$$

$$\frac{a^*b}{b} = a^*$$

$$\frac{a^*b}{bb} = \text{Not possible}$$

$$L_1/L_2 = \{a^*b + a^*\}$$

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

$$baa^*/a$$

$$a^*/a = a^*$$

$$baa^*/a = a^*$$

$$aaa^*/a = aa^*$$

$$\{a, aa, aaa, \dots\} = a^*$$

$$\{aa, aaaa, \dots\} = a^2$$

$$aa\{e, a, aa, aaa, \dots\}$$

$$= aa^*$$

if  $L/\epsilon^* \rightarrow$  prefix is obtained as  $\epsilon^*$  contains all the string.

$$\text{Let } \{aba, baa\}/\epsilon^*$$

$$= \{a, ab, aba, e, b, ba, baa\}$$

$\downarrow$  because  $\epsilon^*$  has each string for division.

$$\{ |L/\epsilon^*| = |\text{INIT}(L)| = \text{no of distinct prefix in } L \}$$

|      | REGULAR | CAL |
|------|---------|-----|
| MIN  | ✓       | ✗   |
| MAX  | ✓       | ✗   |
| HALF | ✓       | ✗   |
| ALT  | ✓       | ✗   |

$$\rightarrow \text{Half}(1010) = 10$$

$$\text{if } L = \{101011, 100\}$$

$$\text{Half}(L) = \{101\}$$

$\downarrow$  take only even

length & take

front half part

## Pumping Lemma for Regular Languages-

$L$  is regular  $\Rightarrow L$  is satisfying pumping lemma for regular languages

There exist some language satisfy PL for RL which are not regular  $\Rightarrow$  True.

if  $L$  don't satisfy PL, then  $L$  is not Regular - True.

if  $L$  is regular then it has FA (NFA) with  $n$  states, let it accept  $|w| \geq n$

then PL says

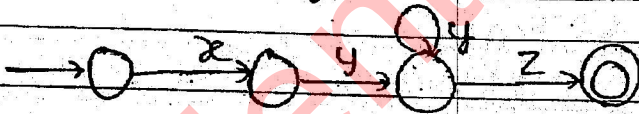
$$w = xyz \quad \> \quad |y| \geq 1$$

$$\therefore xyz \in L(M)$$

if it accepts  $xyz$  it must accept  $xy^*z$

$$xy^*z \in L(M)$$

i.e  $y$  can be pumped.



this the major weakness of FA that it cannot control the loops.

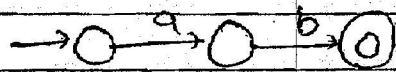
the pumping lemma says that if nfa accept a string of length having atleast equal to no of states ( $n$ )

i.e  $|w| \geq n$  then language is infinite

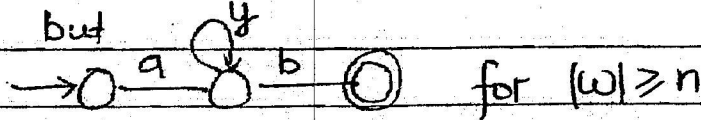
as it says  $xy^*z$  also belong to  $L(M)$



proof - An state NFA, with <sup>out</sup> loop can accept  $n-1$  length string only



but



for  $|w| \geq n$

Applying PL

Ques - 5 state fa accept a string of length 4.

the fa is

(a) finite

(b) infinite

☒ (c) finite or infinite (bcoz from given info, we can comment whether it has loop or not)

- 5 state fa accept a string of length 5 or 6

the language is

1) finite

☒ 2) infinite

- 5 state fa accepts a string of length 4 only

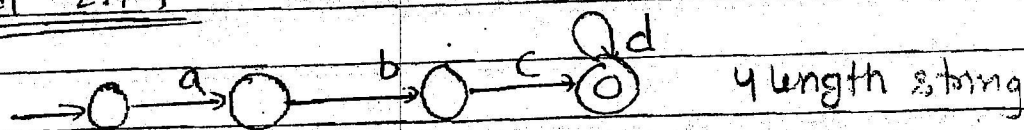
1) finite

\* A FA to be infinite of  $n$  state, it must accept a string of length  $n$  (atleast)

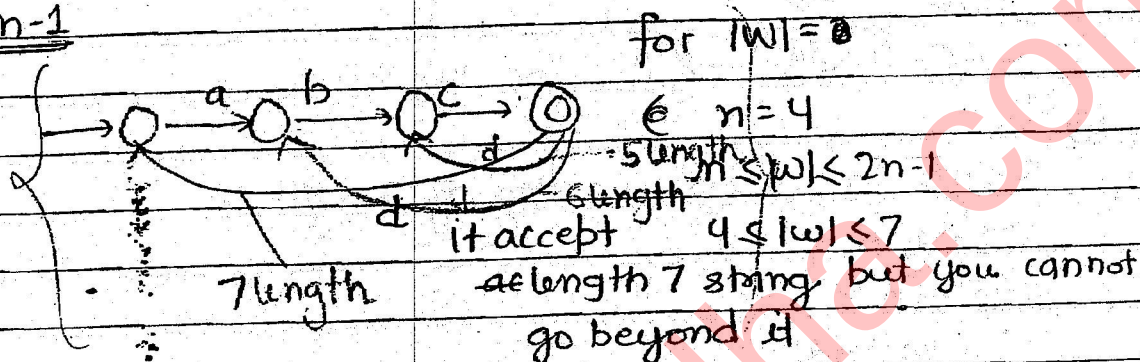
If  $n$  state FA is accepting infinite language, it surely accept the string b/w length  $n$  &  $2n-1$

$$\begin{array}{ccc} & \downarrow & \\ |w| \geq n & & |w| \leq 2n-1 \end{array}$$

proof for  $2n-1$



for  $2n-1$



- I at least one string above  $n$   
 II at most one string of  $2n-1 \leq |w| \leq 2n-1$   
 If it is an infinite language

To check whether a lang is infinite,  
 if any string in the range  $n \leq |w| \leq 2n-1$   
 then the language is infinite otherwise finite.

The transition graph of infinite language surely contains a cycle.

If  $\mathcal{L}$  contains  $w$  of at most  $2n-1$ . is it finite or infinite?

No because it also cover the range below  $n$ . so it is

possible if & only if

$$n < |w| \leq 2n-1$$



● Myhill-Nerode Relation - It is applicable on a Minimal DFA

$$\left\{ \begin{array}{l} \emptyset \cup Rv \Leftrightarrow \delta^*(q_0, u) = \delta^*(q_0, v) \\ \text{then } u \& v \text{ are Related.} \end{array} \right\} \begin{array}{l} \text{Reflexive } uRu \\ \text{Symmetric } uRv \Rightarrow vRu \\ \text{Transitive } uRv, vRw \Rightarrow uRw \end{array}$$

● if  $u \& v$  are related, they belong to same equivalence class.  
in a given fa. (ie which get pushed to same state)

● No of equivalence class = No of states in minimal dfa.

● 97 (b) Union of some of equivalence class.

because in DFA if  $u$  of all class =  $\Sigma^*$

but  $L = \emptyset$  state

● Myhill Nerode is right invariant equivalence relation as

$$\text{if } uRv \Rightarrow \delta^*(q_0, uz) = \delta^*(q_0, vz)$$

$uz \equiv vz$  (Right Variant)

● index of relation = no of classes = it has to be finite as no of state in FA can be finite only.

● For a non regular language, No of class is infinite.

Ques  $1^* \neq 1^*01^* + \dots = (0+1)^*$   
 $\{ \dots = 1^*01^*0(1+0)^* \}$

Ques I  $(000)^*(\epsilon+0)$

II  $(000)^*(\epsilon+0+00)$

III  $(000)^*$

IV  $(0)^*$

II & IV are equivalent.

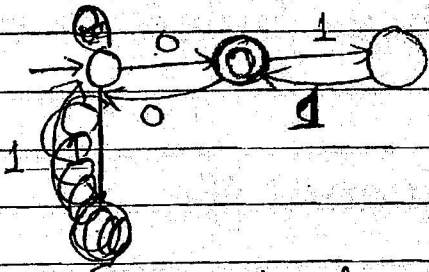


105

Regular  $0^n 1^m$  :  $n+m$  is odd

case 1-  $0+E$   
 $E+0$

Odd no of 0 followed by even no of  
 1



(c)  $pq$  by finding its complement

(d)  $0^p 1^q$   $p+q=10 \rightarrow R$   
 $p \cdot q = 10 \rightarrow R$   
 $p - q = 10 \rightarrow NR$   
 $p/q = 10 \rightarrow NR$

these are regular with all the operators  $(\leq, \geq, =, >, < \neq)$   
 by taking complement.

for  $0^p 1^q$ ;  $p \geq 0, q \geq 0, pq \geq 4$   
 $\downarrow$  Do complement of machine  
 $pq \leq 4$

Let  $L = \{0^p 1^q; p+q \geq 10\}$

the complement is

$\bar{L} = \{0^p 1^q; p+q < 10\}$

all the 1 must not follow

zero &  $p+q < 9$

$\therefore (0+1)^* 10(0+1)^* + \{0^p 1^q \mid p+q < 9\}$

$\downarrow$   
 for all the 1 not  
 following 0

$\downarrow$   
 for less than 9

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

$L$  is regular.

EXCELLENT



for  $(0^p 1^q \mid pq \geq 4)$

$$0^4 1^1 + 0^3 1^1 + 0^2 1^1$$

$$0^4 1^1 + 0^2 1^2 + 0^1 1^4 \quad pq = 4$$

$$(0^4 0^1 1^1 + 0^2 0^1 1^2 + 0^1 0^1 1^4) \quad pq \geq 4$$

but only  $(0^* 1^*)$

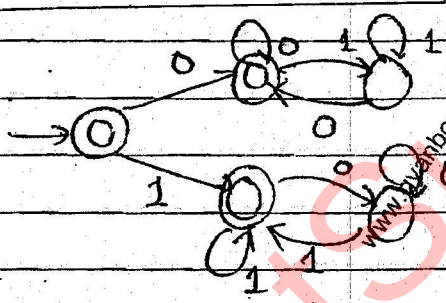
\* In Regular language, it is not allow

$$T^* V T^* V$$

as it allow comparison

$T^* V / V T^*$  - it also allow comparison

Ques (2) -



$$\Rightarrow \{n_{01}(w) = n_{10}(w)\}$$

It is basically not possible to  
because it ← compare no of 1 & 0.  
require infinite M/M (guideline violator)  
but possible to comp. 10 & 01  
bit require finite M/M.

Ques  $n_{000}(w) = n_{111}(w) \rightarrow$  NR as we can have infinite 000

$n_{011}(w) = n_{110}(w) \rightarrow$  Regular but we cannot have even

$$\underline{0110110} \text{ (equal)}$$

अगर 011 ayega to possible nhi

hai ki fir se 011 aaye kyunki

It will lead to 110 110 so it is not

possible to have infinite memory case

for 0  
1

(only two  
states are  
required)

comparison

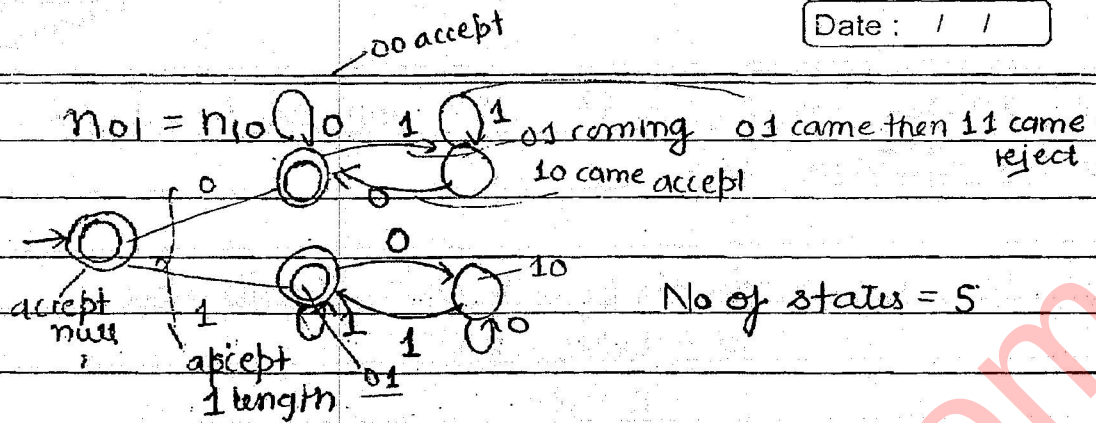
Page No.

Date : / /

www.gyanbookcenter.com

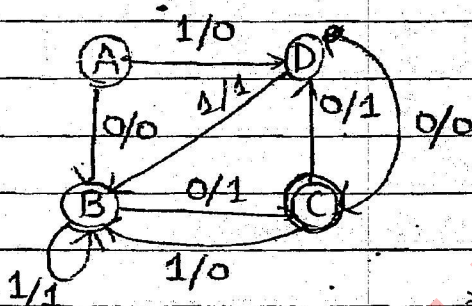
EXCELLENT





Unknown initial state -

133



to find out starting state to reach C?

00 for A  
0 for B

(a) 01 for A → O/P B  
10 for A → O/P C

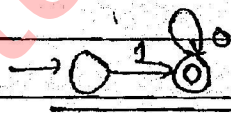
B → O/P C

D → O/P C

Shortest path must work from each node A, B, D

139- 1, 2, 4, 8, ...  $2^n$

1, 10, 100, 1000, ... 100...



Re  $\rightarrow 10^*$

1, 3, 9, 27, ...  $3^n$  is not Regular in binary  
but in ternary

1, 10, 100, 1000, ...  $k, k^2, \dots, k^n$   
1 3 9  $\downarrow$  it is  
Regular