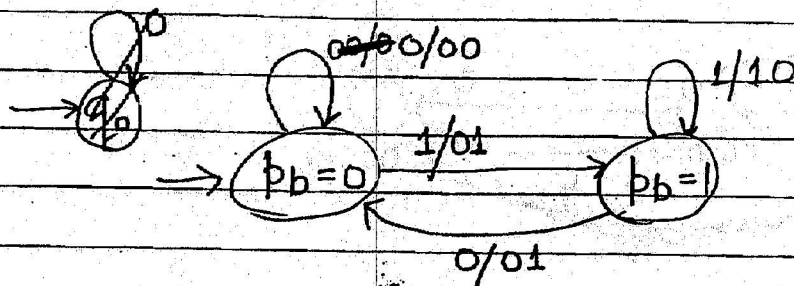


Sum of present & previous bit -



We have taken 2 states here as we have to remember only previous bit & it may be 0 or 1.
 ∴ only two states required

Context Free Languages-

Every Mealy Machine is Moore machine - False
 but for each Mealy, there exist an equivalent Moore - true

① $L \rightarrow G$ & Types of CFL's (Standard CFL & grammar)

② $G \rightarrow L$

② Derivations: LMD, RMD, Derivation tree, partial DT

↳ Right Most Derivation

↳ Left Most Derivation

Ambiguity of Grammar

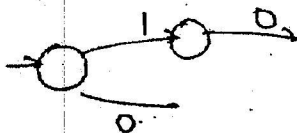
Ambiguity of Language

③ Membership algorithm - CYK Algorithm - complexity - $O(n^3)$

• Brute Force parsing Algorithm
 ↳ complexity - $O(k^n)$

• Properties of LL(k) & LR(k) Grammars.

↳ It will take $O(n)$ time to check for membership.



~~Handwritten~~
(10+01)*

Page No.

Date: / /

④ Algorithm in CFG:

- Removal of ϵ -productions.
- Removal of unit production.
- Removal of useless production.
- Removal of left recursion. } both causes ambige
- Removal of left factoring. }
- Conversion of CFG to CNF (Chomsky Normal Form)
- Conversion of CFG to GNF (Greibach Normal Form)

• Properties of CNF & GNF

⑤ PDA programming

- DPDA & NPDA working
- Machine to Language conversion.

OR

Language to Machine conversion.

• Every Regular language is a CFL.

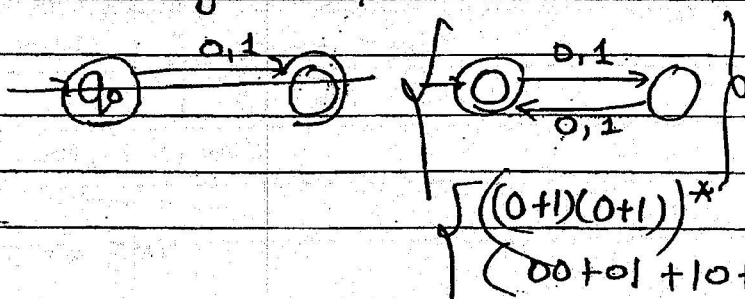
• PDA can do

• counting comparison

• string matching, but ~~states~~ FA cannot do both function

ques I $L = \{ w_1 w_2 \mid |w_1| = |w_2| \}$ — Regular language
 II $L = \{ w_1 \# w_2 \mid |w_1| = |w_2| \}$ — Not Regular but CFL as length of w_1 has to be stored.

I - as it is string of even length. if I/P is even length then it get accepted.



Ques - All CFL's but not regular

$$1. \{0^n 1^n \mid n \geq 0\}$$

$$S \rightarrow 0S1 \mid \epsilon$$

$$\{ (01)^n (23)^n \mid n \geq 0 \} \quad S \rightarrow 01S23 \mid \epsilon$$

$$\bullet \{0^n 1^n \mid n \geq 1\}$$

$$N- \left\{ \begin{array}{l} S \rightarrow 0A1 \\ A \rightarrow 0A1 \mid \epsilon \end{array} \right\}$$

or

$$S \rightarrow 0S1 \mid 01$$

$$\bullet \{0^n 1^n \mid n \geq 2\}$$

$$S \rightarrow 0S1 \mid 0011$$

Ques $G_1 \rightarrow S \rightarrow 0S1 \mid \epsilon$

$$G_2 \rightarrow S \rightarrow 0S1 \mid 01$$

$$G_3 \rightarrow S \rightarrow 0S1 \mid 01 \mid 0011$$

$$G_4 \rightarrow S \rightarrow 0S1 \mid 0011$$

} G_2, G_3 are equivalent

Whenever check for equivalence of grammar initially check for equivalence of minimal string.

Ques - a) $(0+1)^* 1 (0+1)^*$

b) $(0+1)^* 11 (0+1)^*$

c) $(0+1)^* 11 (0+1)^* + 1$

None is equivalent

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

b) $(0+1)^* 11$

c) $(0+1)^* 1 + (0+1)^* 11$

a, c are equivalent

To check for equivalence check for minimal & then start

as 0, 1

00, 10, 11, 10

and so on - EXCELLENT

Ques - $(a+b)^*(b+ab) \equiv (a+b)^*b$

$\{0^n 1^n\}$

In CFL's, same language can be written in two ways -

conditional comparison - $\{0^m 1^n \mid m=n\}$

Direct comparison $\{0^n 1^n \mid n \geq 0\}$

for $\{ww^R \mid w \in \{0,1\}^*\}$

$\forall$ condⁿ comparison - $\{w_1 w_2 \mid w_1 = w_2^R, w_1, w_2 \in \{0,1\}^*\}$

$\{0^n 1^{2n} \mid n \geq 0\}$

$S \rightarrow 00S11 \mid \epsilon$

$\{0^n 1^{3n} \mid n \geq 0\}$

$S \rightarrow 0S11 \mid 00S111 \mid \epsilon$

$S \rightarrow 000S111 \mid \epsilon$

You can push any no of digit or symbol at a time in PDA but you can pop only one symbol at a time.

$\{0^n 1^{2n} \mid n \geq 2\}$ $S \rightarrow 00S1 \mid 001111$

if $S \rightarrow 0^n S 1^{2n} \mid 00111$

$\downarrow$
 $\{0^{n+2} 1^{2n+3} \mid n \geq 0\}$

then let $n+2=k$ $n=k-2$

$\Rightarrow \{0^k 1^{2n+3} \mid n \geq 0\}$

$\Rightarrow \{0^k 1^{2k-1} \mid k \geq 2\}$

$\{0^n 1^{2n-1} \mid n \geq 2\}$

to do it normally
start checking for
minimal string
only do for two
function.

$$* \{ 0^{2n} 1^n \mid n \geq 0 \}$$

$$S \rightarrow 00S1 \mid \epsilon$$

$$* \{ 0^{2n} 1^n \mid n \geq 2 \}$$

$$S \rightarrow 00S1 \mid 000011$$

$$* \{ 0^{2n} 1^{3n} \mid n \geq 0 \}$$

$$S \rightarrow 00S111 \mid \epsilon$$

↳ It must be linear 0^{2n^2} will be CSL

$$* \{ 0^{2n+1} 1^{3n+2} \mid n \geq 0 \}$$

$$= \{ 0^{2n} \textcircled{011} 1^{3n} \mid n \geq 0 \}$$

↳ stopper, just separate the const. powers.

$$S \rightarrow 00S111 \mid 011 \quad \text{and make constant a stopper}$$

if given

$$S \rightarrow 00S111 \mid 011$$

solve it

$$0^{2n} S 1^{3n} \mid 011$$

$$0^{2n} 0111 1^{3n}$$

$$\{ 0^{2n+1} 1^{3n+2} \mid n \geq 0 \}$$

↳ PDA can solve linear function.

if given $\{ 0^m 1^n \mid m = \frac{2}{3}n \}$

$$\downarrow$$

$$3m = 2n \quad \text{Yes it is a CFL.}$$

$$\{ 0^m 1^n \mid m = 0.1n \} - \text{CFL}$$

$$10m = n$$

$$\{ 0^m 1^n \mid m = \sqrt{2}n \} - \text{cannot be written as it must be integer.}$$

$m = 2^n$ cannot be done as it is not regular.

Part - PDA can solve $\geq, \leq, <, >, \neq, =$

$$m \geq 2n \quad m \neq 2n$$

$$m \leq 2n \quad m < 2n$$

Ques - $\{0^m 1^n \mid m \geq n\}$

$$S \rightarrow 0S1 \mid 0 \mid \epsilon$$

$\{0^m 1^n \mid m \leq n\}$

$$S \rightarrow 0S1 \mid 1 \mid \epsilon$$

$\{0^m 1^n \mid m > n\}$

$$S \rightarrow 0S1 \mid 0$$

$\{0^m 1^n \mid m < n\}$

$$S \rightarrow 0S1 \mid 1$$

Ques - $\{0^m 1^n \mid m = n\} = \{0^n 1^n \mid n \geq 0\}$

$$S \rightarrow 0S1 \mid \epsilon$$

$\{0^m 1^n \mid m \geq n\} = \{0^{m+n} 1^n \mid n \geq 0\}$

$$S \rightarrow 0S1 \mid AS1 \mid \epsilon \quad A \rightarrow 0A \mid \epsilon$$

$\{0^m 1^n \mid m \leq n\} = \{0^m 1^{n+m} \mid m, n \geq 0\}$

$$S \rightarrow 0S1 \mid SA \mid \epsilon \quad A \rightarrow A1 \mid \epsilon$$

$\{0^m 1^n \mid m > n\} = \{0^{m+n} 1^n \mid m \geq 0, n \geq 0, m \geq 1\}$

$$S \rightarrow 0S1 \mid AS \mid \epsilon \quad A \rightarrow 0A \mid 1$$

$\{0^m 1^n \mid m < n\} = \{0^m 1^{m+n} \mid m \geq 0, n > 0\}$

$$S \rightarrow 0S1 \mid SA \mid \epsilon \quad A \rightarrow A1 \mid 1 \quad n \geq 1$$

$\{0^m 1^n \mid m \neq n\} = \{0^m 1^n \mid m > n \text{ or } n > m\}$

$$\left\{ \begin{array}{l} S_1 \rightarrow 0S_11 \mid AS_1 \mid \epsilon \\ S_2 \rightarrow 0S_21 \mid S_2A \mid \epsilon \\ A \rightarrow 0A \mid 1 \\ B \rightarrow B1 \mid 1 \end{array} \right\}$$

$\Downarrow$

$\hookrightarrow$ DCFL

$\{ \text{Union of two language for } m > n \text{ and } n > m \}$

as it $m > n$ or $n > m$ is DCFL as it has only one comparison there is no need of double comparison.

but if $m > n$ or $n > p$, it is CFL as it require two comp.

$$I = \{0^m 1^n \mid m > n \text{ or } m < n\} \rightarrow \text{DCFL}$$

$$II. \{0^m 1^n 2^p \mid m > n \ \& \ m < p\} - \text{CSL}$$

$$III \{0^m 1^n 2^p \mid m > n \text{ or } m < p\} - \text{CFL}$$

$$IV \{0^m 1^n \mid m > n \ \& \ n > m\} - \text{Regular}$$

as it is $\emptyset$

Ques

$$S \rightarrow AS$$

$$S \rightarrow 000S1/\epsilon \rightarrow 0^{3n} 1^n$$

$$A \rightarrow 0A/\epsilon \quad 0^m 1^n \mid m \geq 3n$$

$$S \rightarrow AS$$

$$S \rightarrow 0S1/01 \rightarrow 0^m 1^n \mid m \geq n \geq 1$$

$$A \rightarrow 0A/\epsilon$$

$$\{0^m 1^n \mid m \neq n\} \rightarrow \left. \begin{array}{l} S \rightarrow S_1/S_2 \\ A \rightarrow 0A1/\epsilon \\ S_1 \rightarrow XA \\ X \rightarrow 0X/0 \\ S_2 \rightarrow AY \\ Y \rightarrow Y1/1 \end{array} \right\}$$

$$\text{or } \left\{ \begin{array}{l} S \rightarrow XA/AY \\ X \rightarrow 0X/0 \\ Y \rightarrow Y1/1 \\ A \rightarrow 0A1/\epsilon \end{array} \right\}$$

$$\left\{ \begin{array}{l} S \rightarrow AX/AY \\ X \rightarrow 0X/0/\epsilon \\ Y \rightarrow Y1/1 \\ A \rightarrow 0A1/\epsilon \end{array} \right\} \begin{array}{l} m \geq n \\ (0+1)^3 \\ m < n \end{array}$$

$m \leq n \quad m \geq n$
 $S \rightarrow xA|AY$
 $A \rightarrow 0A1|\epsilon$
 $X \rightarrow 0X|\epsilon$
 $Y \rightarrow 1Y|\epsilon$

$(0+1)^*$ but Now the grammar became ambiguous.

{ Ambiguity of a CFL is undecidable }

Primary CFL

$\{ 0^m 1^n 2^p \mid m, n, p \geq 0 \}$

— Regular

$S \rightarrow 0AB$
 $A \rightarrow 0A|\epsilon$
 $C \rightarrow 1C|\epsilon$
 $B \rightarrow 2B|\epsilon$

$\{ 0^m 1^n 2^p \mid m=n, p \geq 0 \}$

— DCFL

$S \rightarrow AB$

$A \rightarrow 0A1|\epsilon$

$B \rightarrow 2B|\epsilon$

$\{ 0^m 1^n 2^n \mid m, n \geq 0 \}$

$S \rightarrow AB$
 $A \rightarrow 0A|\epsilon$
 $B \rightarrow 1B2|\epsilon$

— DCFL

$\{ 0^m 1^n 2^m \mid m, n \geq 0 \}$

$S \rightarrow AB$
 $S \rightarrow 0S2|1A|\epsilon$
 $A \rightarrow 1A|\epsilon$

— DCFL

$\{ 0^m 1^n 2^m \mid m \geq 1, n \geq 2 \}$

$S \rightarrow 0S1|0A1$
 $A \rightarrow 1A|11$

$\{0^m 1^n 2^p \mid m=n \text{ or } m=p\}$ CFL but not DCFL
as one variable comp. with or

 $S \rightarrow S_1 \mid S_2$
 $S_1 \rightarrow OS_1 1 \mid \epsilon$
 $S_2 \rightarrow OS_2 AB$
 $A \rightarrow OA 1 \mid \epsilon$
 $B \rightarrow 2B \mid \epsilon$
 $S_2 \rightarrow C$
 $C \rightarrow OC 2 \mid D$
 $D \rightarrow 1D \mid \epsilon$
 $S \rightarrow S_1 \mid S_2$
 $\Rightarrow S_1 \rightarrow AB$
 $A \rightarrow OA 1 \mid \epsilon$
 $B \rightarrow 2B \mid \epsilon$
 $S_2 \rightarrow OS_2 2 \mid C$
 $C \rightarrow 1C \mid \epsilon$
 $m=p$
 $\Downarrow$
 ~~$S \rightarrow AB \mid OS_2$~~
 $\{0^n 1^n\} \cup \{0^n 1^{2n}\} \mid n \geq 0$
 $\Downarrow$
 $\{0^m 1^n \mid m=n \text{ or } m=2n\}$ - Not a DCFL but a CFL

 $\left\{ \begin{array}{l} S \rightarrow S_1 \mid S_2 \\ S_1 \rightarrow OS_1 1 \mid \epsilon \\ S_2 \rightarrow OS_2 11 \mid \epsilon \end{array} \right\}$

CFG has power equivalent
to NPDA

 $\{0^m 1^n \mid m \leq n \leq 2m\}$ $\rightarrow$ it will accept 0011

00111 as well
001111

 $\{0^m 1^n \mid m \leq n \text{ or } n \leq 2m\}$
 ~~$S \rightarrow S_1 \mid S_2$~~
 ~~$S_1 \rightarrow OS_1 1 \mid \epsilon$~~
 ~~$S_2 \rightarrow 1S_2 \mid \epsilon$~~
 ~~$S_2 \rightarrow$~~
 $\left\{ \begin{array}{l} S \rightarrow OS_1 \mid OS_1 1 \mid \epsilon \\ \text{ } \end{array} \right\}$

$$\{0^m 1^n \mid m \leq n \leq 3m\}$$

$$\{S \rightarrow OS1 \mid OS11 \mid OS111 \mid \epsilon\}$$

You have to take all 1 to 3
because you can't generate
any other from the
given

$$\{0^m 1^n \mid m \leq n \leq 4m\}$$

$$\{S \rightarrow OS1 \mid OS11 \mid OS111 \mid OS1111 \mid \epsilon\}$$

Ques - $\{w \mid n_0(w) = n_1(w)\}$

$$\{0^n 1^n \mid n \geq 0\} \quad \{w \mid n_0(w) = n_1(w)\}$$

$$\{0^n 1^n \mid n \geq 1\} \quad \{w \mid n_0(w) = n_1(w) \geq 1\}$$

$$\{0^n 1^{2n} \mid n \geq 0\} \quad \{w \mid n_0(w) = 2n_1(w)\}$$

$$\{w \mid n_1(w) = 2n_0(w)\}$$

$$\{0^n 1^{2n+3} \mid n \geq 0\} \quad \{w \mid n_1(w) = 2n_0(w) + 3\}$$

$$\{0^n 1^m 2^p \mid m=n, p \geq 0\} \quad \{w \mid n_0(w) = n_1(w), n_2(w) \geq 0\}$$

$$\{0^m 1^n 2^p \mid m=n \text{ or } m=p\} \quad \{w \mid n_0(w) = n_1(w), \text{ or } n_0(w) = n_2(w)\}$$

$$\{0^m 1^n \text{ or } 0^n 1^{2n}\} \quad w \mid n_0 = n_1 \text{ or } n_0 = 2n_1$$

$$\bullet \{w \mid n_0(w) = n_1(w)\}$$

$$\text{if } G: S \rightarrow 0S1 \mid 1S0 \mid \epsilon$$

$$\downarrow$$

$$\{0110 \notin G\}$$

So we can not use it.

$$S \rightarrow 0S1 \mid 1S0 \mid \cancel{00S11} \mid \cancel{01S10}$$

$$S \rightarrow \cancel{0S11S0} \mid \cancel{01S10}$$

this grammar will not generate

0110

1001

001100

$\therefore$ the c/p grammar is

$$S \rightarrow 0S1 \mid 1S0 \mid SS \mid \epsilon$$

to allow the production side by side.

for equal
no. of zero
and 1.

OR

$$S \rightarrow 0S1S \mid 1S0S \mid \epsilon$$

OR

$$S \rightarrow S0S1 \mid S1S0 \mid \epsilon$$

equivalent
to each
other

but if $S \rightarrow 0S1S \mid S1S0 \mid \epsilon$

it will miss the string ending with 0 & start with 1

1001

$$\bullet \{w \mid n_0(w) = 2n_1(w)\}$$

do min. combination possible-

$$S \rightarrow 0S0S1 \mid 0S1S0 \mid 1S0S0 \mid SS \mid \epsilon$$

001

010

100

↓
S is putted to allow all

for side by side

00110

EXCELLENT

$$\left\{ \begin{array}{l} \text{OR} \\ S \rightarrow SOSISO \mid SISOISO \mid SOSOSI \mid \epsilon \\ \text{OR} \\ S \rightarrow OSISOI \mid IOSOSI \mid OSOSIS \mid \epsilon \end{array} \right\}$$

$$\cdot \{ w \mid n_o(w) = n_i(w) \} \quad \text{for } O^n I^n$$

$$S \rightarrow OSI \mid ISO \mid SS \mid \epsilon$$

$$S \rightarrow OSI \mid \epsilon$$

Counting comparisons are useful for calculation of parenthesized expression for matching parenthesis. Matching of parenthesis comes with every looping structure.

$O^n I^n$ $\rightarrow$ balanced
allow $(^n)^n$ $((((($)))

$$n_o(w) = n_i(w)$$

it will allow everything
()) (

but we want left one before right one bracket so we need a grammar b/w these two grammar.

So $\rightarrow$

$$S \rightarrow (S) \mid SS \mid \epsilon$$

$\rightarrow$ properly balanced parenthesis structure.

$$S \rightarrow (S) \mid \epsilon \rightarrow \{ \}$$

$$S \rightarrow (S) \mid)S(\mid SS \mid \epsilon \rightarrow (())) () \text{ everything}$$

$$S \rightarrow (S) \mid SS \mid \epsilon \rightarrow (() ()) - \text{correct properly parenthesized.}$$

$G \rightarrow S \rightarrow (S) \mid () \mid SS \rightarrow ((())()) \rightarrow$ properly balanced
but null never came.

• Instead of null it will give $()$

1. $((()))$
2. $((()()))$
3. $((())())$ - properly balanced
4. $(())()$

for balanced $\{w \mid n_L(w) = n_R(w)\}$

for properly balanced - $\{w \mid n_L(w) = n_R(w) \ \& \ n_L(v) \geq n_R(v) \mid v \text{ is any prefix of } w\}$

as $((())) \rightarrow$ prefix condition satisfied.

~~$((()))$~~ $((()(($ $\rightarrow$ not satisfied prefix condition

for prefix $((()(($ $L < R$

Grammar for balanced one without null string. balanced.

$\{S \rightarrow (S) \mid)SL \mid SS \mid () \mid \epsilon\}$

$\Downarrow$
 $\{w \mid n_L(w) = n_R(w) \ \& \ w \notin \epsilon\}$

III family

Even palindrome - $S \rightarrow 0S0 \mid 1S1 \mid \epsilon$ $\rightarrow$ CFL $\{ww^R \mid w \in (0,1)^*\}$

odd palindrome - $S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1$ $\{wxw^R \mid w \in (0,1)^*, x \in \{0,1\}\}$

All palindrome $S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$

$$\{ww^R \cup wxw^R\}$$

OR

$$\{w \mid w = w^R\}$$

palindrome -

$S \rightarrow 0S0 \mid 1S1 \mid \#$ $\rightarrow$ DCFL

$$L(G) = \{w\#w^R \mid w \in (0,1)^*\}$$

24-Apr

Workbook chapter-1

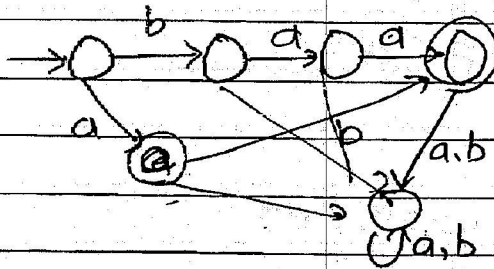
Page No.

Date : / /

- d)
- 2) c
 - 3) b
 - 4) c
 - 5) d
 - 6) c
 - 7) b
 - 8) a
 - 9) (b)
 - 10) (b) (2×3) states
 - 11) (d)
 - 12) (b)
 - 13) (b)
 - 14) (c)
 - 15) (a)
 - 16) (a)
 - 17) (d)
 - 18) (c)
 - 19) (a)
 - 20) (d)
 - 21) (d) $\rightarrow$ (c) is not there as $L(R_1) \cap L(R_2) = \emptyset$
 - 22) (d)
 - 23) (c)
 - 24) (a)
 - 25) (d)
 - 26) (b)
 - 27) (a)
 - 28) (a)
 - 29) (d)
 - 30) b
 - 31) b
 - 32) (b)
 - 33) (d)
 - 34) (d)
 - 35) (d)
 - 36) (a)
 - 37) (d)
 - 38) (c)
 - 39) (d)
 - 40) (d)
 - 41) (a)
 - 42) (d) (Start checking with smaller strings.)
 - 43) (c) ~~it is (c)~~
 - 44) (a)
 - 45) (b) ~~it is (a)~~
 - 46) (d)
 - 47) (b)
 - 48) (d)
 - 49) (d)
 - 50) (c)
 - 51) (b)
 - 52) (b)
 - 53) (c)
 - 54) b
 - 55) c
 - 56) b
 - 57) a
 - 58) b
 - 59) c
 - 60) b
 - 61) a
 - 62) c
 - 63) (d)
 - 64) a
 - 65) a
 - 66) b
 - 67) d
 - 68) a
 - 69) b
 - 70) a
 - 71) d
 - 72) (c)
 - 73) c
 - 74) a
 - 75) (d)
 - 76) (e)
 - 77) (c)
 - 78) (d)
 - 79) (a)
 - 80) (c)
 - 81) (c)
 - 82) (a)
 - 83) (c)
 - 84) (e)
 - 85) (d)
 - 86) (c)
 - 87) (d)
 - 88) (d)
 - 89) (d)
 - 90) (b)
 - 91) a
 - 92) d
 - 93) a
 - 94) b
 - 95) d
 - 96) d
 - 97) e
 - 98) d
 - 99) d
 - 100) a
 - 101) c
 - 102) d
 - 103) b
 - 104) c
 - 105) e
 - 106) (a)
 - 107) (d)
 - 108) (c)
 - 109) c
 - 110) d
 - 111) d
 - 112) b
 - 113) b
 - 114)
 - 115)
 - 116)
 - 117)
 - 118)
 - 119)
 - 120)

EXCELLENT

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



a no of q less than 2

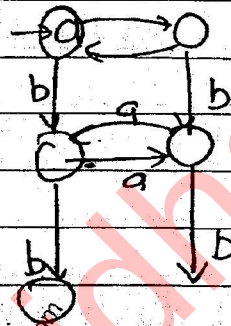
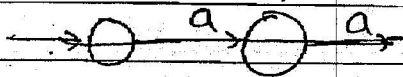
no of a less than 4

↓ Union

no of a less than 4

as $A \cup B = A$

If $B \subseteq A$



In an ordinary DFA, one way to accept & one way to reject a string

- 2DFA, one way to accept but two ways to reject a string

nonfinal state

infinite loop

if a machine hangs, string is considered to be rejected

2-NFA to reject a string

Non final state

DC

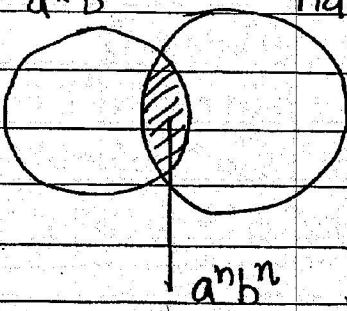
Hang

if $\delta^*(q_0, 101001) = \infty$ (hang)

Basically

a^*b^*

$n_a(w) = n_b(w)$



if given

$R_1 = a^*b^*$

$R_2 = n_a(w) = n_b(w)$

$R_1 \cap R_2 = a^n b^n$

$L_1 = a^n b^n$

$L_2 = n_a(w) = n_b(w)$

$L_1 \cap L_2 = a^n b^n$
 $L_1 \cup L_2 = L_2$

EXCELLENT

if $a \times 2 \leq a \times 4$

↓ then n is a multiple of 2

as $A \cap B = B$ if $B \subseteq A$

Ques - $\epsilon + (11^*0)(11^*0 + 0)^*$

$$= (11^*0 + 0)^* \quad \text{as } (\epsilon + 11^*11^*) = 11^*$$

$$= ((11^* + \epsilon)0)^*$$

$$= (1^*0)^* \quad - \{ \epsilon, 0, 10, 110, \dots \}$$

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

↓ 1101 does not belong to it

Ques $(a+ba)^*(a+ba)$ (as if it is ending with a it's always

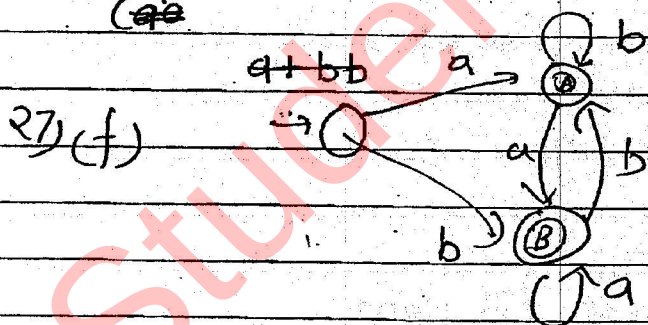
$$= (a+ba)^*a$$

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

a having b before a as we are provided with $(a+b)^*$)

When a π is given, check for its minimization.

(a)



$$\pi_A = a(b+aa^*b)^* + b(a+bb^*a)^*bb^*$$

$$\pi_B = a(b+aa^*b)^*aa^* + b(a+bb^*a)^*$$

$$\pi_A + \pi_B = a(b+aa^*b)^*(\epsilon + aa^*)$$

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

$$\left\{ \begin{aligned} \pi_A + \pi_B &= a(b+aa^*b)^*a^* + b(a+bb^*a)^*b^* \\ &= a(a^*b)^*a^* + b(b^*a)^*b^* \end{aligned} \right\}$$

if $a^*b^*a(a+b)^* \equiv (a+b)^*$ - false

it has (a)

$a^*b^*a(a+b)^* \equiv a(a+b)^*$ - false it force it to start with

Page No.

Date

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

2. $c^*ab^*(a+b)^* = c^*a(a+b)^*$

3. $a^*ac^*(a+b)^* = a^*a(cc^*)(a+b)^*$

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

5. $(a^*b^*)^* = (b^*a^*)^*$

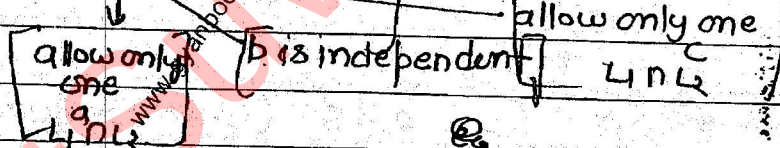
6. $(0^*1^*2^*)^* = (0+1+2)^*$

if $L_1 = \text{Non-regular}$, $L_3 = L_1 L_2 = \text{Regular}$ $L_2 = ?$

↓ Regular

* $L_1 = \{a^i b^j c^k \mid i, j \geq 0\}$
 $L_2 = \{a^i b^j c^k \mid i, j \geq 0\}$

$L_1 \cap L_2 = \{a^i b^j c^k \mid i, j \geq 0\}$



* $L_1 = (ab)^* \cap (a^*b^*)$
 $= \epsilon, ab$
 $|L_1| = 2$

EXCELLENT

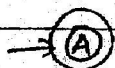
Conversion of Grammar to Machine-

- to convert Right Linear GR to Machine

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

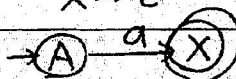
for T^* -

for $A \rightarrow \epsilon$



for $A \rightarrow qx$ put an arbitrary state

$$x \rightarrow \epsilon$$



$$A \rightarrow qbc$$

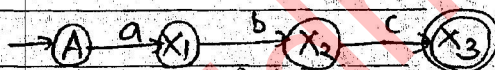
$\Downarrow$

$$A \rightarrow qX_1$$

$$X_1 \rightarrow bX_2$$

$$X_2 \rightarrow cX_3$$

$$X_3 \rightarrow \epsilon$$



for $T^* V$ -

for -

$$S \rightarrow qB$$

$$B \rightarrow \epsilon$$

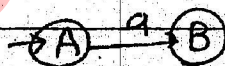
• $A \rightarrow B$ (Unit Production)

as $A \rightarrow \epsilon.B$



Every unit production corresponds to null move.

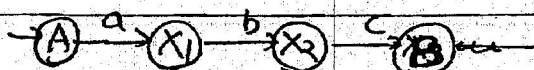
• $A \rightarrow qB$



Every null production corresponds to final state.

$$A \rightarrow abcb$$

$$A \rightarrow qX_1, X_1 \rightarrow bX_2, X_2 \rightarrow cB$$



Ques 38 -

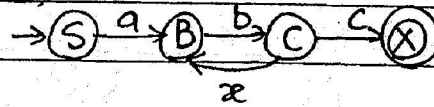
$$S \rightarrow aB$$

$$B \rightarrow bc$$

$$C \rightarrow xB$$

$$C \rightarrow C' \rightarrow C \rightarrow CX$$

$$X \rightarrow \epsilon$$



$$a(bx)^*bc$$

or

$$ab(xb)^*bc$$

$$\text{as } \{(pq)^*p = p(qp)^*\}$$

- $\gamma + \phi = \gamma$
- $\gamma + \epsilon = \gamma + \epsilon$
- $\gamma \cdot \epsilon = \gamma$
- $\gamma \cdot \phi = \phi$

if $\gamma + \epsilon = \epsilon$ if γ contains ϵ

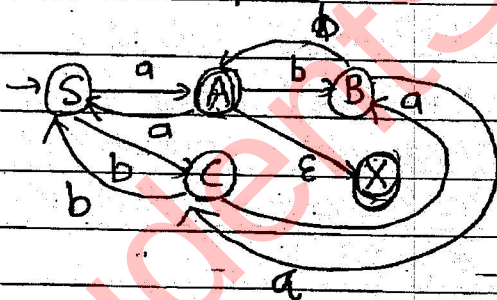
Ques - "for RLGR try to draw the machine."

$$S \rightarrow aA | bC$$

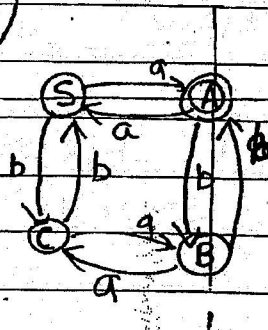
$$A \rightarrow aS | bB | \epsilon$$

$$B \rightarrow bA | aC$$

$$C \rightarrow aB | bS$$



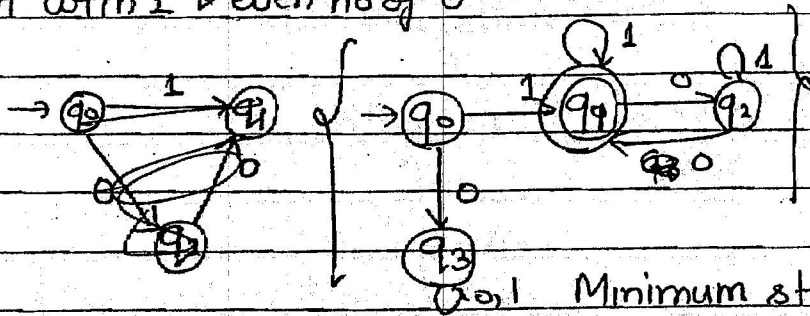
$$a^* + b^*a^* +$$



odd a	even b	odd a
-------	--------	-------

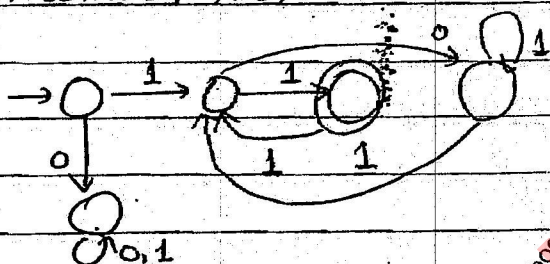
$$\{x^+x^+x^+\}$$

w/ start with 1 & even no of 0



Minimum state for DFA = 4
NFA = 3

• start with 1 & $d(w) \bmod 3 = 0$



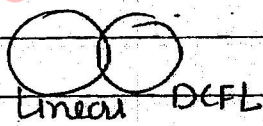
ques

53. 3 possible answers-

1. $a(abbb)^*ab + ba(bab)^*$ loop at A
 2. $aa(bba)^*b + ba(bab)^*$ loop at B
 3. $aaab(bab)^* + ba(bab)^*$ directly at C
- all are equiv $\downarrow$ bcoz loop can be resolved at 3 point A, B, C.

RL = LL < LINEAR < CFL < CSL

RL = LL < DCFL < CFL < CSL



Both P & NP problem are in REC