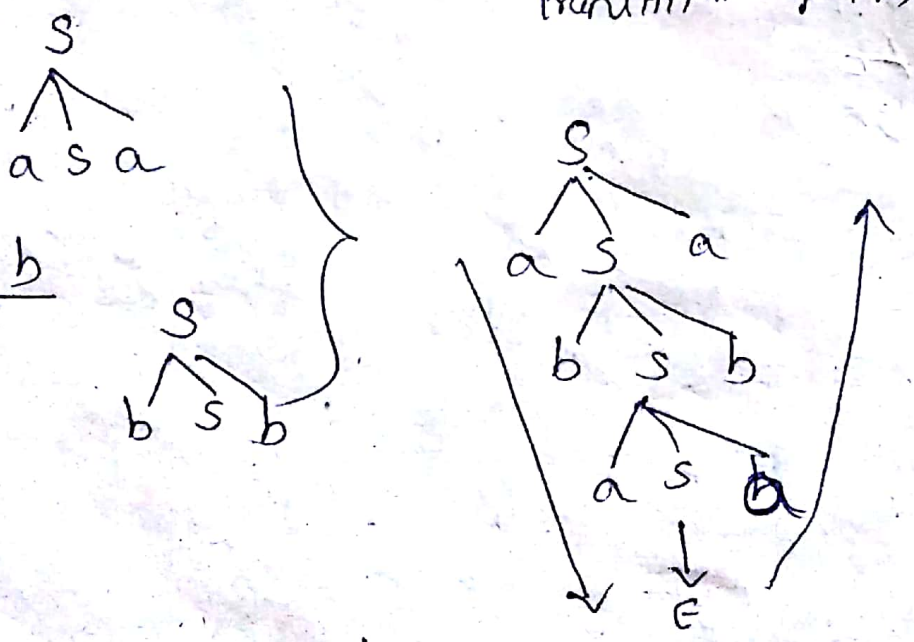
Derivation Trees (DT)

1) Draw a D.T for the string (w) = abaaaba (combination of Terminals) with the given CFG where productions are

P: { S → asa
S → bsb
S → a/b/∈
↓
S → a
S → b
S → ∈

First to start with a & productions as possible

S → asa, S → a
↓ this is not because there is no transition again



w = abaaaba

2) w = aabbabba

P: (Q) S → aB / bA

$A \rightarrow a/as/bAA$

$B \rightarrow b/bs/aBB$

$S \rightarrow aB$

$S \rightarrow bA$

$A \rightarrow a$

$A \rightarrow as$

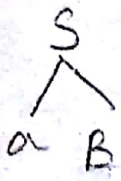
$A \rightarrow bAA$

$B \rightarrow b$

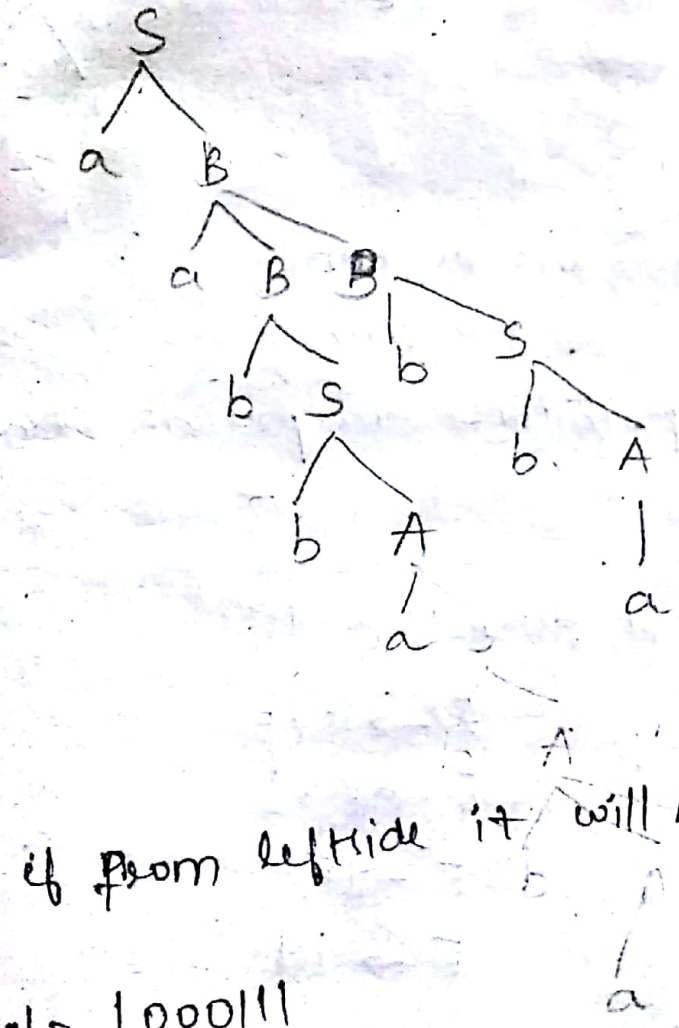
$B \rightarrow bS$

$B \rightarrow aBB$

For a string



For b string



⇒ if from leftside it will be left most derivation

Rightmost derivation

$W = 1000111$

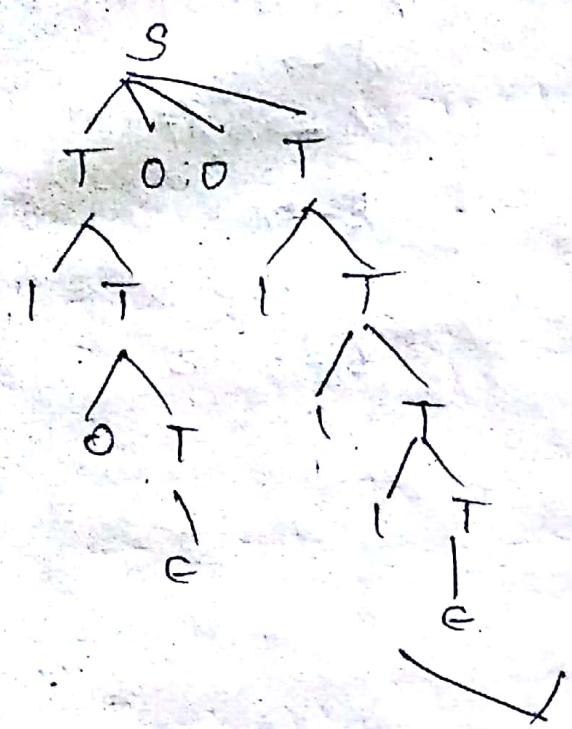
P: $S \rightarrow TOOT$

$T \rightarrow OT$

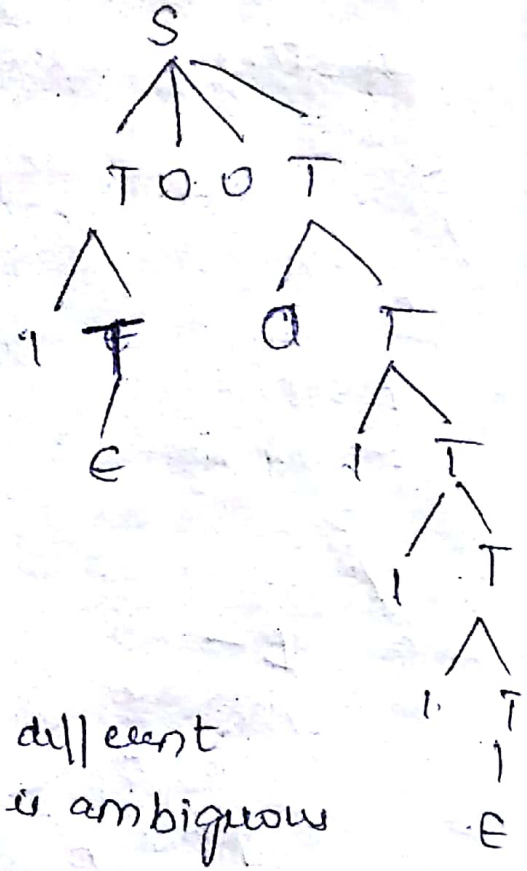
$T \rightarrow 1T$

$T \rightarrow E$

Left most derivation



Right most



two are different
it is ambiguous

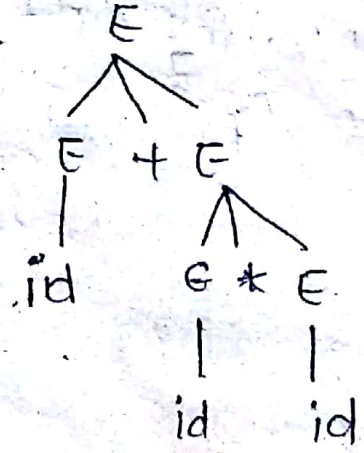
⇒ More than one posibilities are present then it is ambiguous.

2 distinct left most derivation tree

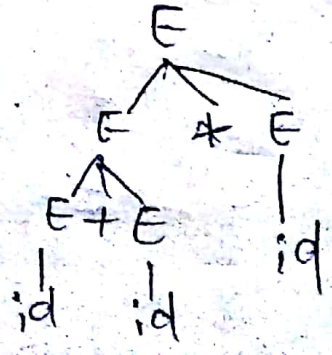
4) W = id + id * id

G: E → E+E
E → E*E
E → (E)
E → id

LMD



RMD



Derivations using grammer There are 2 approaches to check productions of a C.F.G

i) The more convectional approach is to use the rules from body to head called as "Recursive Inference".

ii) on second approach it is used to define the language from head to body called as "Derivation".

example:

Recursive Inference (body to head) :-

W = (a+b)

1) E → E
2) E → E+E
3) E → I
4) I → a
5) I → b

here (I+I) is not then in production so again take a & b & convert it into E

used E → I → (a)

is used to take that inference

story

string Inference	language symbol	production	strings used
(i) a	I	I → a (4)	—
ii) b	I	I → b (5)	—

iii) a	E	E → I	a
iv) b	E	E → I	b
v) a+b	E	E → E+E	a, b (or) (iii), (iv)
vi) (a+b)	E	E → (E)	(v)

Derivation (head to body).

$G = VTPS$

Let $\alpha A \beta$ be a string of terminals & variable with 'A' is a variable

i.e α & β are strings in $(V \cup T)^*$

Let $A \to \gamma$ be a production of G then we say $\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$

$\Rightarrow$ is converting to according to G.P (Grammar production)

$\Rightarrow$ we may extend the transfer symbol for 0, 1 or many no. of transitions

$\Rightarrow^*$ only for zero or many

example:

$w = (a+b)$

1) $E \to E+E$
2) $E \to (E)$
3) $E \to I$
4) $I \to a$
5) $I \to b$

$E \Rightarrow (E) \Rightarrow (E+E) \Rightarrow (I+E) \Rightarrow (a+E) \Rightarrow$

$(a+I) \Rightarrow (a+b)$

leftmost derivation :- & Rightmost.

At each step in deriving a string we replace left most variable by one of its production body, Such derivation is called Left Most derivation and it is indicated by $\underset{lm}{\Rightarrow}$ (either for 1 step),

$\overset{*}{\underset{lm}{\Rightarrow}}$ (multiple transitions)

eg:- $w = (a101 + b1) * (a1+b)$

1 substitute another substitute

P:- $E \to I$ $E \to E+E$ $E \to E*E$ $E \to (E)$ $I \to a$ $I \to b$ $I \to Ia$ $I \to Ib$ $I \to I0$ $I \to I1$

$E \underset{lm}{\Rightarrow} E*E \underset{lm}{\Rightarrow} (E)*E \underset{lm}{\Rightarrow} (E+E)*E \underset{lm}{\Rightarrow}$

$(I+E)*E \underset{lm}{\Rightarrow} (I_1+E)*E \underset{lm}{\Rightarrow} (I01+E)*E$

↓ Converting I_1 to $I0$

$\underset{lm}{\Rightarrow} (I101+E)*E \underset{lm}{\Rightarrow} (a101+E)*E \underset{lm}{\Rightarrow}$

$(a101+I)*E \underset{lm}{\Rightarrow} (a101+I_1)*(E+E)$

$\underset{lm}{\Rightarrow} (a101+b1)*(I+I) \underset{lm}{\Rightarrow} (a101+b1)*(I_1+I)$

$\underset{lm}{\Rightarrow} (a101+b1)*(a1+b)$ //

Rightmost

Similarly it is possible to require that at each step the Rightmost variable is replaced by one of its body such derivation is called Rightmost derivation

Same eg as Leftmost derivation

$E \underset{lm}{\Rightarrow} E*E \underset{lm}{\Rightarrow} E*(E) \underset{lm}{\Rightarrow} E*(E+E)$

$\underset{lm}{\Rightarrow} E*(I+E) \underset{lm}{\Rightarrow} E*(I_1+E) \underset{lm}{\Rightarrow}$

$E*(I_1+b) \underset{lm}{\Rightarrow} E*(a_1+b) \underset{lm}{\Rightarrow}$

$(E)*(a_1+b) \underset{lm}{\Rightarrow} (E+E)*(a_1+b)$

⇒ The language of a Grammer

If $G = V T P S$ is a CFG. The language of G is denoted as $L(G)$ is a set of terminal strings

That have derivations from the stack symbol

i.e $L(G) = \{ w \text{ in } T^* / S \overset{*}{\underset{G}{\Rightarrow}} w \}$

if L is a language of some CFG then

L is said to be a CFL language (context free).

parse tree :-

The tree representation for derivations is known as parse tree when used in compilers.

The tree structure of the source program facilitates the translation of source program into executable code by allowing natural, recursive fn to perform this translation process.

constructing parse tree

$G = VTPS$ the parse tree for 'G' with the following conditions

(i) each interior node is labelled by a variable 'V'.

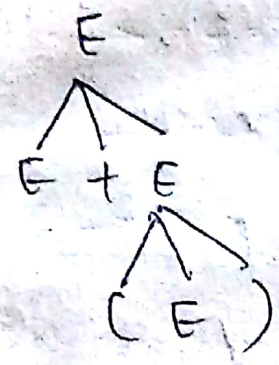
(ii) Each leaf is labelled by either a variable, a Terminal or 'ϵ' (epsilon).

(iii) If the leaf is labelled as 'ϵ' then it must be a only child of its parents

(iv) If an interior node is labelled as 'A' and its children are labelled $x_1, x_2 \dots x_n$ then production on are $A \to x_1 / x_2 / \dots x_n$

eg:- $E \rightarrow E + E$

$E \rightarrow (E)$



(or)

⇒ Yield of parse tree :-

Concadinating the leafs of any parse tree from the left becomes a string. That string is called as yield of a tree

which is always a string i.e derived from root variable

i) All the leafs are labelled either with a terminal or with epsilon (ε).

ii) The root is labelled with start symbol

Applications of CFGs:-

1) CFGs has a way to describe instances of concepts which are recursively defined & Multiplicol

Two of their uses are

i) Grammars are used to describe programming languages. There is a mechanical way of converting the language description into a parse tree

ii) The development of XML is widely predicted and essential part of XML is DTD it is of context free grammer that describes all the Tags

properties of CFG:-

1) eliminate useless productions
2) eliminate Null production (ϵ)

} to get normal form of CFG.

3) eliminate unit production

$A \to B$, $B \to C$, $C \to D$, $D \to i$

$A \to C$, $A \to D$, $\boxed{A \to i}$ → only 1 production can be there now no use of all those productions

Normal form CFG

- always Chomsky: Variable → Variable (two) $A \to BC$; $A \to a$ terminal
- Greibach

Elimination of useless

example:-

1) $S \rightarrow aAa/aB$

$A \rightarrow as/bD$

$B \rightarrow aBa/b$

$C \rightarrow abb/DD$

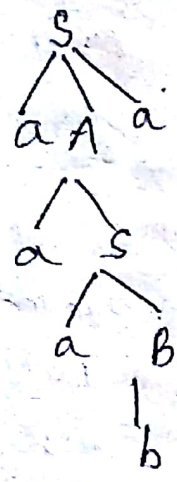
$D \rightarrow aDa$

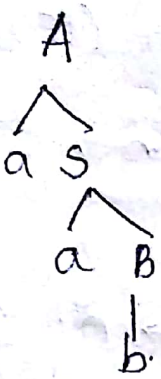
Step1 eliminate useless variables productions.

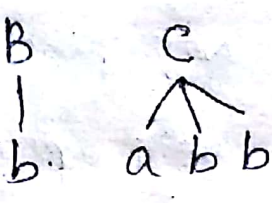
$V: \{S, A, B, C, D\}$

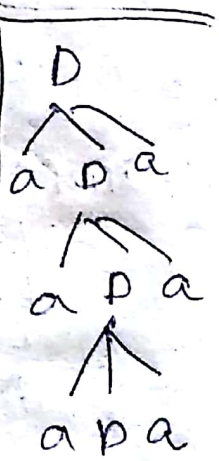
$T \{a, b\}$.

check every variable having a proper terminal state or not









(The variable which do not have terminal state is useless)

one Terminal state

after eliminating useless one 'D'

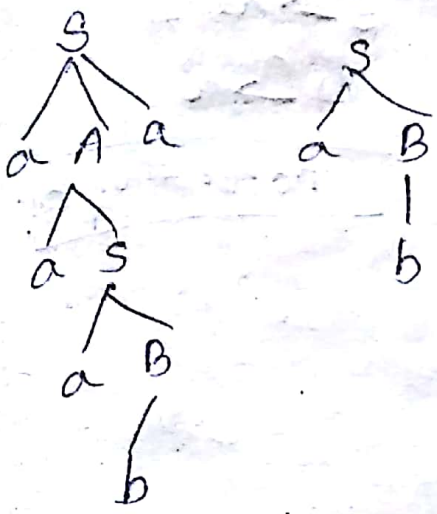
$S \rightarrow aAa/aB$

$A \rightarrow as$

$B \rightarrow aBa/b$

$C \rightarrow abb$ ✗

useless production.



from start to the end state symbol to deriving any string

then is no production of 'C' in this without ε also we can go to a & b so it is a useless production

$S \rightarrow aAa/aB$

$A \rightarrow as$

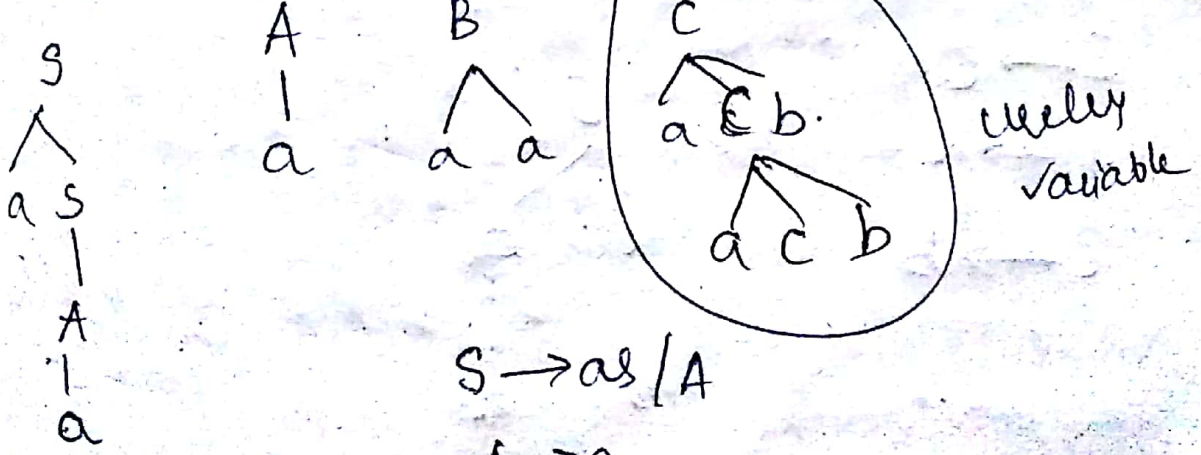
$B \rightarrow aBa/b$

2) $S \rightarrow as/A/C$ $\quad V=\{S, A, B, C\}$

$A \rightarrow a$ $\quad T=\{a, b\}$

$B \rightarrow aa$

$C \rightarrow a\varepsilon b.$

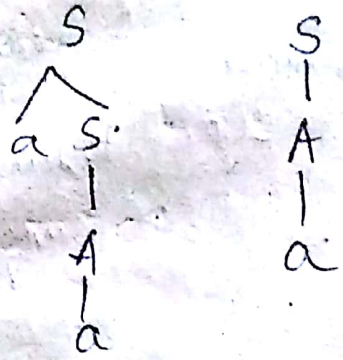


$S \rightarrow as/A$

$A \rightarrow a$

$B \rightarrow aa$

useless production



$S \to aS \mid A$

$A \to a$

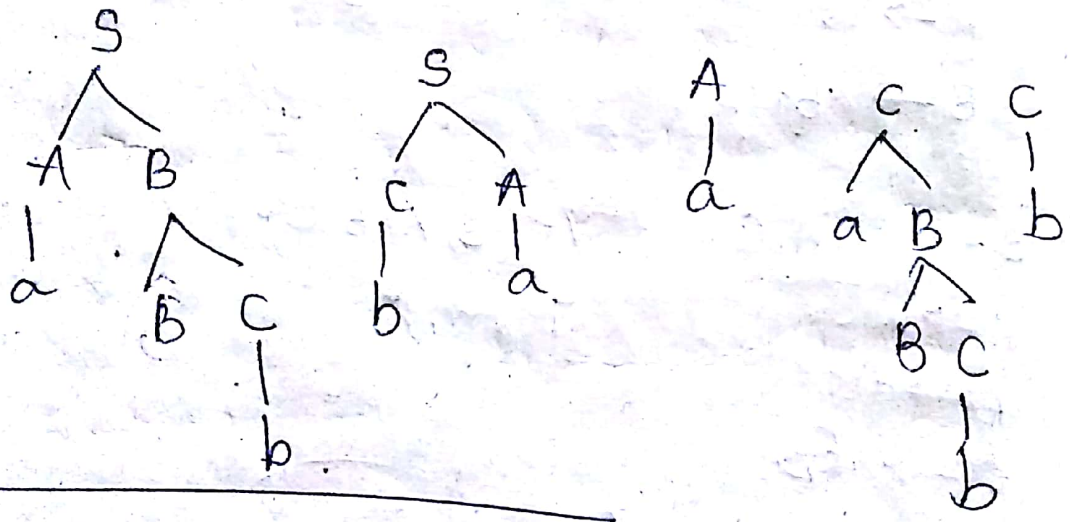
no need of B.

3) $S \to AB \mid CA$

$B \to BC \mid AB$

$A \to a$

$C \to aB \mid b$



$S \to CA$

$A \to a$

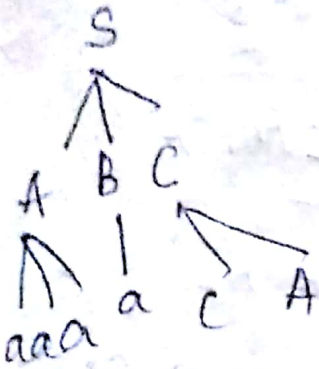
$C \to b$

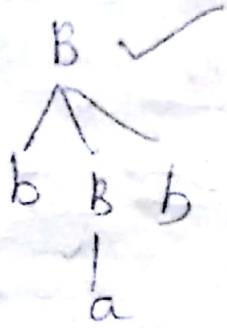
4) S → ABC | BaB

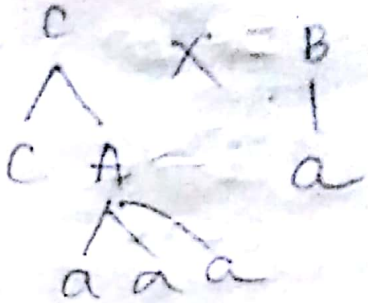
A → aA | BaC | aaa

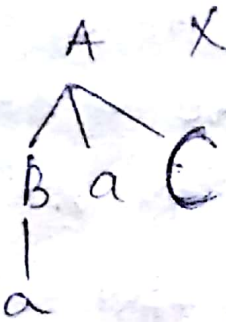
B → bBb | a

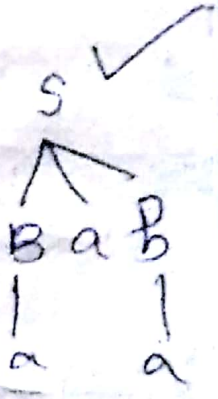
C → CA | Ac









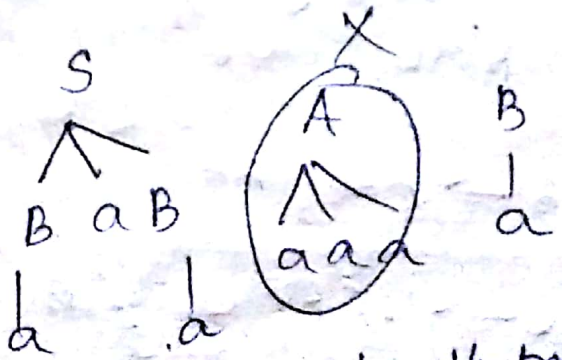


eliminate A

S → BaB

A → aA | aaa

B → bBb | a



directly from start symbol.

S → BaB

B → bBb | a

Elimination of ε

5) S → ABC
A → BC/a
B → bAC/ε
C → cAB/ε

V{S, A, B, C}
T{a, b, c}

Epsilon production Sub 'ε' in 'B'

B → ε
C → ε

S → ABC / AC / AB / A

↓ replace B with ε

ACε ⇒ is a string itself

Sub 'ε' in C ⇒ AC

ABε ⇒ AB

for both 'B & C' 'ε' the A.

A → C / B / a / BC first write the given string

↓ Base ↓ case

B → bAC / bA keep C as 'ε' the bAε = bA

C → cAB / CA / ~~CAB~~

keep B as ε the CAε = CA

6) S → ABaC
A → BC
B → b/ε
C → D/ε D → ∩

V = {S, A, B, C, D}
T = {a, b}

$B \to \epsilon$

$C \to \epsilon$

$S \to ABaC \mid AaC \mid ABa \mid Aa$ → first writing itself; → replace null B as ε; → replace null C as ε; → both B & C as ε

$A \to BC \mid B \mid C \mid$

$b \to b$

$C \to D$

$D \to d$

7) $S \to as \mid bA$ $\quad A \to \epsilon$

$A \to aA \mid \epsilon$

Sub $A \to \epsilon$ $\quad b\epsilon = b$

$S \to as \mid bA \mid b$

$A \to aA \mid a$ → A variable is taking to one single variable

elimination & unit production

8) $I \to a \mid b \mid Ia \mid Ib \mid Io \mid I1$

$F \to I \mid (E)$ $\quad V = \{E, T, F, I\}$

$T \to F \mid T * F$ $\quad T = \{a, b, 0, 1\}$.

$E \to T \mid E + T$

$I \to a \mid b \mid Ia \mid Ib \mid Io \mid I1$

$F \to a \mid b \mid Ia \mid Ib \mid Io \mid I1 \mid (\epsilon)$

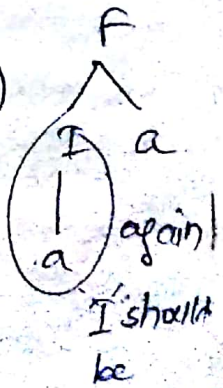
$F \to I$

$T \to F$

$\epsilon \to T$

$T \to a \mid b \mid Ia \mid Ib \mid Io \mid I1 \mid (E) \mid T * F$

$E \to a \mid b \mid Ia \mid Ib \mid Io \mid I1 \mid (E) \mid T * F \mid E + T$



9) S → Aa | B
B → A | bb
A → a | bc | B

S → AB B
B → A
A → B

S → bb | a | bc | Aa
B → a | bc | bb
A → bb | a | bc

11/09/18 :-

1) X is generating - $X \overset{*}{\Rightarrow} \omega$

2) X is reachable - $S \overset{*}{\Rightarrow} \alpha X \beta$

Eliminating ε production:

If Language 'L' has a CFG then L-{ε} has a CFG without epsilon productions (ε).

If Variable 'A' is Nullable if $A \overset{*}{\Rightarrow} \epsilon$ then whenever A appears in a production body then drive ε for A.

We make 2 versions

i) 'A' would have be used to drive a 'ε'

ii) 'A' still present in the production

Eliminating unit productions:

A unit production is of the form A→B where A, B are Variables.

To eliminate this unit production it expands the Variable body into the Variable

Chomsky Normal form:

After generating CFG it is possible to convert it into Chomsky normal form

eg: $E \to EPT/TMF/LER/a/b/IA/IB/IZ/IO$

$T \to TMF/LER/a/b/IA/IB/IZ/IO$

$F \to LER/a/b/IA/IB/IZ/IO$

$I \to a/b/IA/IB/IZ/IO$ chomsky NF

$A \to a$

$B \to b$

$Z \to 0$

$O \to 1$

$P \to +$

$M \to *$

$L \to ($

$R \to)$

(Vaiable) → terminal

Step1: write Variables

$V = \{E, T, F, I, A, B, Z, O, P, M, L, R\}$

terminals

$T = \{a, b, 0, 1, +, *, (,)\}$

TO check chomsky NF there are 2 cases

Vaiable → two Vaible

$A \to BC$

$A \to a$

Vaiable → Terminal

Not in chomsky NF are

$E \to EPT/TMF/LER$ → (T) because Vaiable → (X) false

$T \to TMF/LER$ $F \to LER$ 3 vaibles it has

Take a new Variable

Let $S_1 \rightarrow EP$; $E \rightarrow ST$ → EP

it became a chomsky model

Let $W \rightarrow EP$; $\Rightarrow E \rightarrow WT$

$X \rightarrow TM \Rightarrow E \rightarrow XF$

$Y \rightarrow LE \Rightarrow E \rightarrow YR$

$T \rightarrow XF$

$T \rightarrow YR$

$F \rightarrow YR$

all these should be written in the final grammer

↳ to make it a chomsky Normal form. here we have taken new variable & assign two variable to 1 variable and substitute in eqn ①

we need to take a variable which is not present in the variable list

final grammer should be compulsory

eg

⇒ Begin with a grammer eliminate ϵ, unit, useless productions and convert into CNF

$S \rightarrow ABC \mid BaB$

$A \rightarrow aA \mid BaC \mid aaa$

$B \rightarrow bBb \mid a \mid D$

$C \rightarrow CA \mid AC$

$D \rightarrow \epsilon$

(i) eliminate 'ϵ'

$D \rightarrow \epsilon$

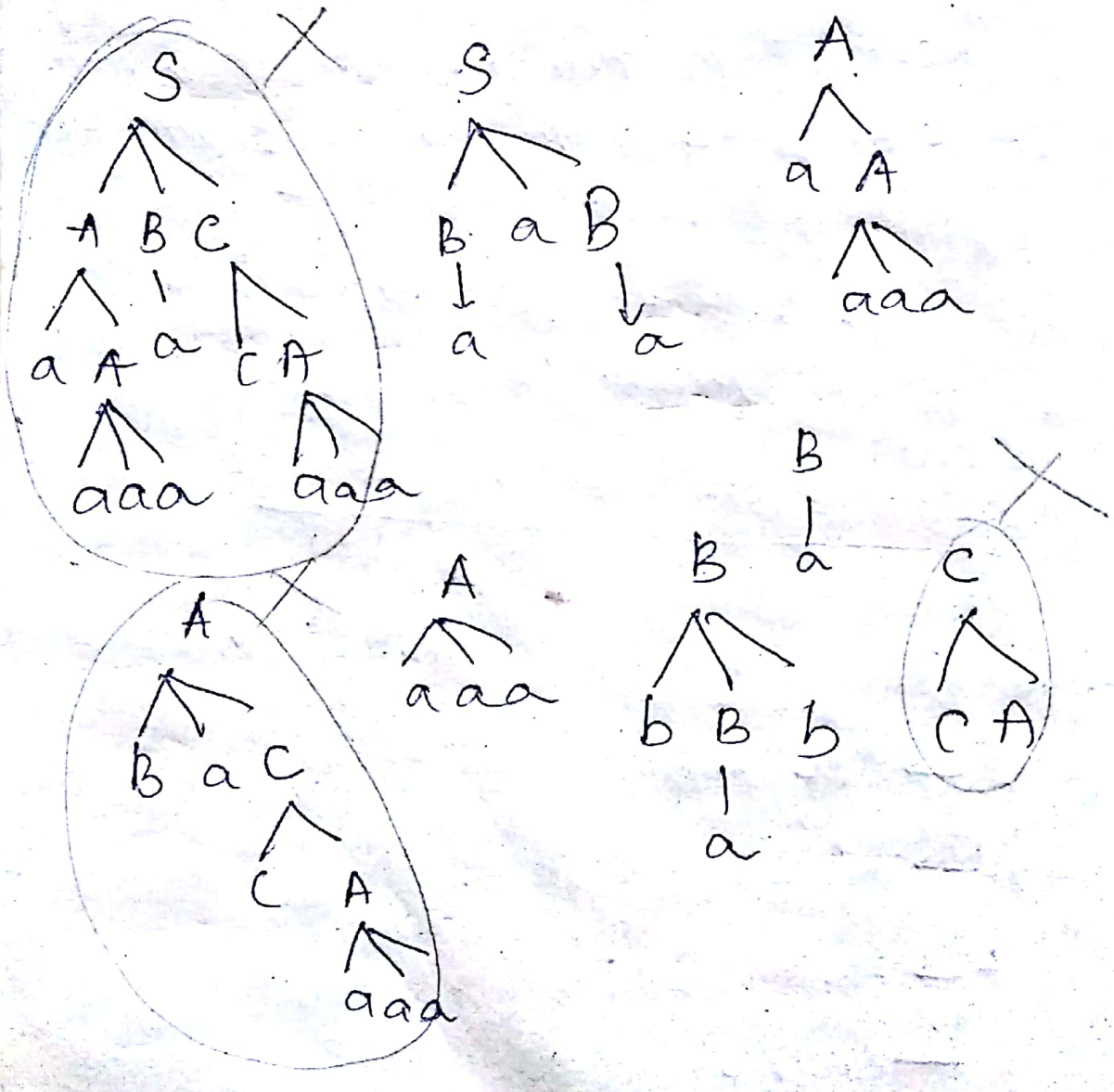
$S \rightarrow ABC \mid BaB$

$A \rightarrow aA \mid BaC \mid aaa$

$B \rightarrow bBb \mid a \mid D$ (write symy as it is. and substitute (ϵ) so nothing)

$C \rightarrow CA \mid AC$

(ii) eliminate ~~unit~~ useless

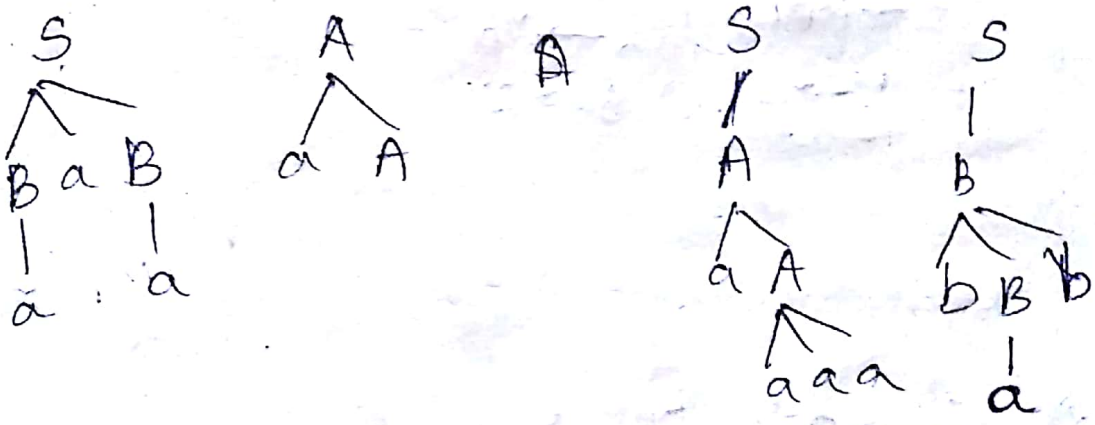


$S \rightarrow BaB$

$A \rightarrow aA \mid aaa$ ✗

$B \rightarrow bBb \mid a$

$C \rightarrow CA \mid AC$ ✗ gone (useless production)



$S \rightarrow BaB$

$B \rightarrow bBb \mid a$

chomsky form is $B \rightarrow a$

~~useless~~ unit production :-

$B \rightarrow D$ → since it has no 'D' production then no substitution

$S \rightarrow ABC \mid BaB$.

$S \rightarrow aA \mid Bac \mid aaa$

$B \rightarrow bBb \mid a$

$C \rightarrow CA \mid AC$

Sub 'B' in the place of $B \to a$

$S \to BBB$

$X \to BB$

$S \to BX$

$B \to bBb$

let $Y \to b$

① $Z \to YB$

② $B \to ZY$

---

2) $S \to aAa \mid bBb \mid \epsilon$

$A \to C \mid a$

$B \to C \mid b$

$C \to CDE \mid \epsilon$

$D \to A \mid B \mid ab$

$S \to \epsilon$

$C \to \epsilon$

1) eliminate 'ϵ'

$S \to aAa \mid bBb$

$A \to C \mid a$

$B \to C \mid b$

$C \to CDE \mid DE$

$D \to A \mid B \mid ab$

(ii) eliminate unit production

$A \to C$

$B \to C$

$D \to A$

$D \to B$

$S \to aAa/bBb$

$A \to C/a$

$B \to C/b$

$C \to CDE/DE$

$D \to A/B/ab$

(iii) eliminating useless production

Since there is no $\in$ production so it is useless.

$S \to aAa/bBb$

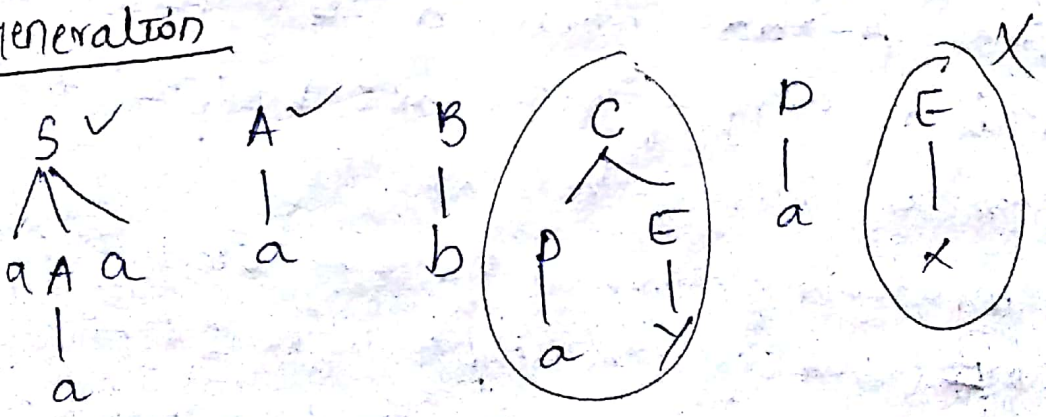
$A \to CDE/DE/a$

$B \to CDE/DE/b$

$C \to CDE/DE$

$D \to CDE/DE/a/b/ab$

Generation



$S \to aAa/bBb$

$A \to a$

$B \to b$

$D \to a/b/ab$

for removing useless productions

$S \rightarrow aAa / bBb$

$A \rightarrow a$

$B \rightarrow b$

Sub 'a' as 'A' b as 'B'

$S \rightarrow AAA / BBB$

① $X \rightarrow AA$

② $Y \rightarrow BB$

③ $S \rightarrow AX / BY$ ④

⑤ $A \rightarrow a$

⑥ $B \rightarrow b$

Greibach N.F :-

Every non-empty language without ϵ is $L(G)$ for some grammer 'G' each of whose productions are in the form $A \rightarrow a\alpha$ where 'a' is a terminal & α is a string of 0(or more Variable This form is called Greibach NF

Converting a grammer to this form starting with CNF (chomsky) and expanding the 1st variable of each production until it will get terminated

→But there can be cycles, where we can never reach terminal state it is neccesary to