Unit-3 TAFL Notes
(Part-I)

Context free Grammar: (CFG):

Context Free Grammar also called Type-2 Grammar can be obtained by ~~sabati~~ retaining the restriction on the left side as in regular grammar but permitting any thing on the R.H.S.

i.e. Production rule

$$\alpha \rightarrow \beta$$

α ↓ Single variable (Non-terminal)

β ↓ any combination of terminals or nonterminals

ex:1

$S \rightarrow abB$
$B \rightarrow bB$
$B \rightarrow \lambda$

ex:2

$S \rightarrow AB$
$A \rightarrow aB$
$B \rightarrow c$

Each regular grammar is context-free grammar but vice-versa is not true.

ex:

$S \rightarrow aB$
$B \rightarrow d$

This regular grammar is also CFG.

"but a CFG need not to be regular grammar"

$S \rightarrow abB$
$B \rightarrow bB$
$B \rightarrow \lambda$

This is CFG but not regular grammar

Context free Language (CFL)

The set of strings generated from a CFG forms a Context free Language (CFL)

~~Example~~ In order to generate strings from the CFG, we need to find derivations from it.

example

$$\left.\begin{aligned} S &\rightarrow aSa \\ S &\rightarrow bSb \\ S &\rightarrow \lambda \end{aligned}\right\} \text{CFG}$$

Typical derivation

$$\begin{aligned} S &\Rightarrow aSa \\ &\Rightarrow abSba \quad (S \rightarrow bSb) \\ &\Rightarrow ab\cdot\lambda\cdot ba \quad (S \rightarrow \lambda) \\ &\Rightarrow \underline{abba} \end{aligned}$$

Another derivation

$S \Rightarrow aSa$, $\Rightarrow a\cdot\lambda\cdot a = \underline{aa}$ $(S \rightarrow \lambda)$

$S \Rightarrow bSb \Rightarrow b\cdot\lambda\cdot b \Rightarrow \underline{bb}$

Another derivation

~~$S \Rightarrow bSb$~~

~~$S \Rightarrow bSb$~~

~~$S \Rightarrow baSab$~~

$S \rightarrow bSb$

$S \rightarrow baSab$

$S \rightarrow ba\lambda ab$

$S \rightarrow \underline{baab}$

Strings should always be considered only if they are generated from start variable.

So $L = \{\lambda, aa, bb, abba, baab \dots\}$

$L = \{ w\cdot w^R \; ; \; w \in \{a, b\}^* \}$

Derivation Types:

① Leftmost Derivation (LMD)

A derivation is said to be leftmost derivation if in each step, the left most variable in the sentential form is replaced.

example

$S \rightarrow aAB$

$A \rightarrow bBb$

$B \rightarrow A \mid \lambda \quad (B \rightarrow A,\ B \rightarrow \lambda)$

LMD:

$$
\begin{aligned}
S &\rightarrow aAB \\
&\Rightarrow abBbB \quad (A \rightarrow bBb) \\
&\Rightarrow abAbB \quad (B \rightarrow A) \\
&\Rightarrow abbBbbB \quad (A \rightarrow bBb) \\
&\Rightarrow abb\lambda bbB \quad (B \rightarrow \lambda) \\
&\Rightarrow abbbbB \quad (B \rightarrow \lambda) \\
&\Rightarrow abbbb\lambda \\
&\Rightarrow \underline{abbbb}
\end{aligned}
$$

Right most Derivation (RMD)

A derivation is said to be rightmost derivation if in each step, the rightmost variable is replaced.

example

$S \rightarrow aAS \mid a$ ($S \rightarrow aAS$, $S \rightarrow a$)

$A \rightarrow SbA \mid SS \mid ba$ ($A \rightarrow SbA$, $A \rightarrow SS$, $A \rightarrow ba$)

RMD

$S \rightarrow aAS$

$\Rightarrow aAa$ ($S \rightarrow a$)

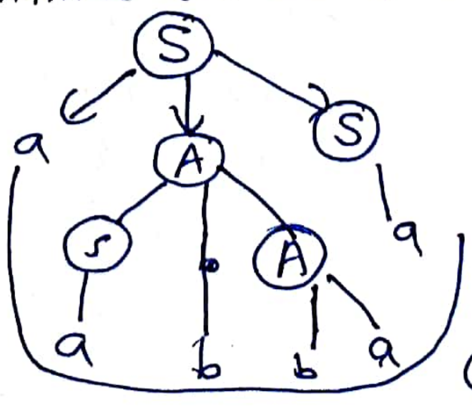
$\Rightarrow aSbAa$ ($A \rightarrow SbA$)

$\Rightarrow aSbbaa$ ($A \rightarrow ba$)

$\Rightarrow aabbaa$

Derivation Tree: (Parse Tree) A derivation tree is the graphical representation of the derivation in Context free Grammar.

In the derivation tree/parse tree, a tree structured form is created during derivation of a string. The left side of production rule forms the parent node and right side forms the children nodes. The process continue till we get all leaf nodes.



= aabbaa

(Derivation tree)

ex: Let G be the Grammar

$S \to 0B \mid 1A$

$A \to 0 \mid 0S \mid 1AA$

$B \to 1 \mid 1S \mid 0BB$

for the string "00110101"

find (a) Leftmost derivation

(b) Rightmost derivation

(c) The derivation tree

Ans (a) Leftmost derivation

$S \to 0B$

$\Rightarrow 00BB \quad (B \to 0BB)$

$\Rightarrow 001B \quad (B \to 1)$

$\Rightarrow 0011S \quad (B \to 1S)$

$\Rightarrow 00110B \quad (S \to 0B)$

$\Rightarrow 001101S \quad (B \to 1S)$

$\Rightarrow 0011010B \quad (S \to 0B)$

$\Rightarrow 00110101 \quad (B \to 1)$

(b) Rightmost Derivation

$S \to 0B$

$\Rightarrow 00BB \quad (B \to 0BB)$

$\Rightarrow 00B1 \quad (B \to 1)$

$\Rightarrow 001S1 \quad (B \to 1S)$

$\Rightarrow 0011A1 \quad (S \to 1A)$

$\Rightarrow 00110S1 \quad (A \to 0S)$

$\Rightarrow 001101A1 \quad (S \to 1A)$

$\Rightarrow 00110101 \quad (A \to 0)$

Derivation Tree

(Using Leftmost Derivation)

LMD

$S \rightarrow 0B$

$\downarrow$

$00BB$

$001B$

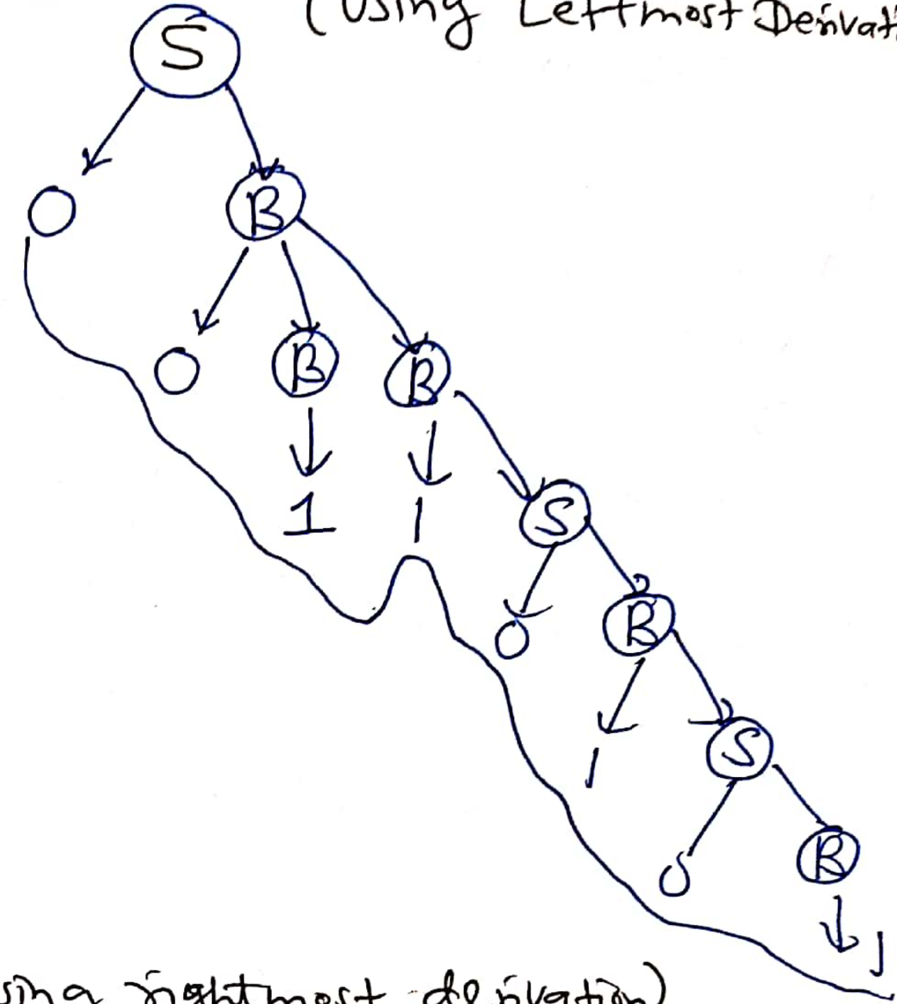
$0011S$

$00110B$

$001101S$

$0011010B$

00110101



Derivation tree (Using rightmost derivation)

RMD

$S \rightarrow 0B$

$\downarrow$

$00BB$

$00B1$

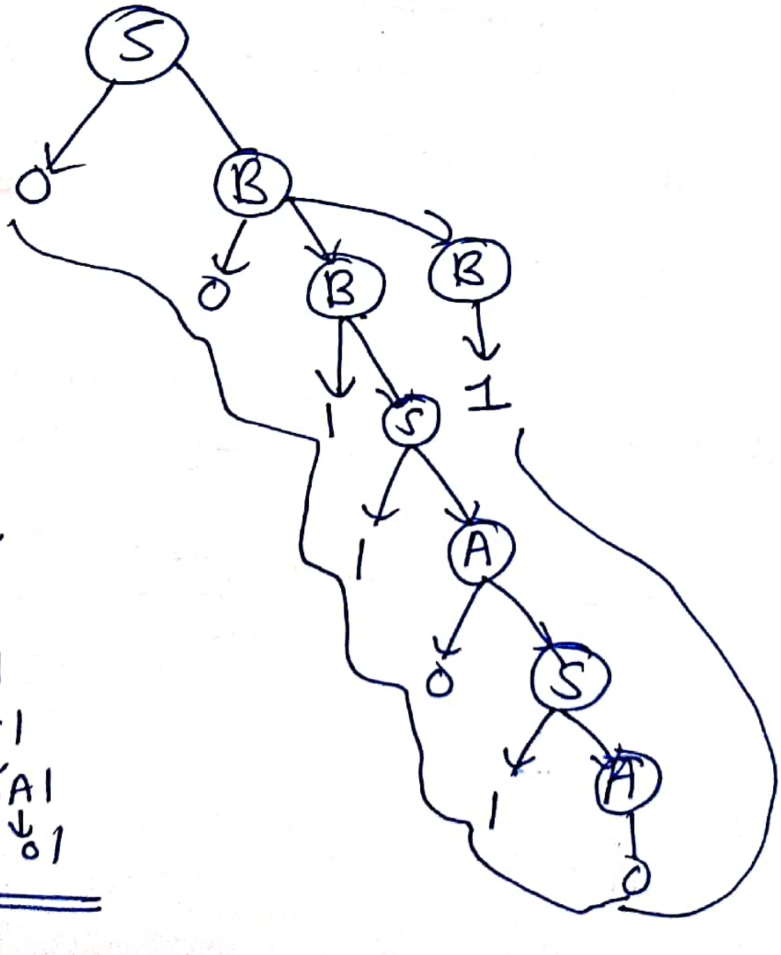
$001S1$

$0011A1$

$00110S1$

$001101A1$

00110101



Ambiguity in Context free Grammar:

Ambiguious Grammar:

A CFG is said to be ambiguous if there are more than one leftmost derivation tree or more than one rightmost derivation tree for a given string.

example-1:

$E \rightarrow E+E$

$E \rightarrow a$

LMD-1

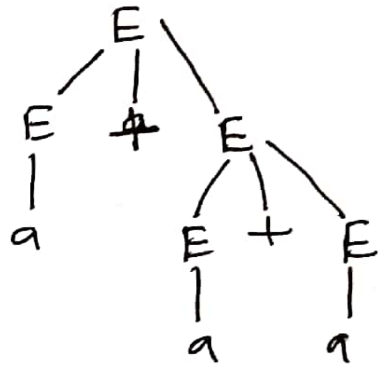
$E \rightarrow E+E$

$E \rightarrow a+E \quad (E \rightarrow a)$

$\rightarrow a+E+E \quad (E \rightarrow E+E)$

$\rightarrow a+a+E \quad (E \rightarrow a)$

$\rightarrow a+a+a \quad (E \rightarrow a)$



LMD-2:

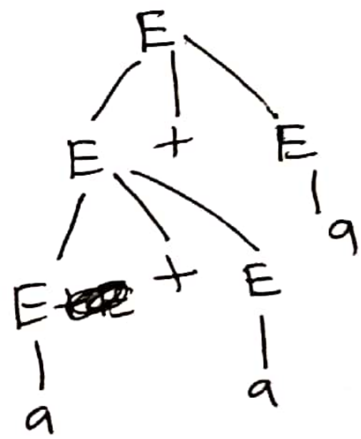
$E \rightarrow E+E$

$E \rightarrow E+E+E \quad (E \rightarrow E+E)$

$E \rightarrow a+E+E \quad (E \rightarrow a)$

$E \rightarrow a+a+E \quad (E \rightarrow a)$

$E \rightarrow a+a+a \quad (E \rightarrow a)$



Two different derivation tree for same string "a+a+a" using LMD. So given CFG is ambiguous.

ex:2 Check whether the following CFG is ambiguous for given string "a+a*b" (8)

$S \to S+S$

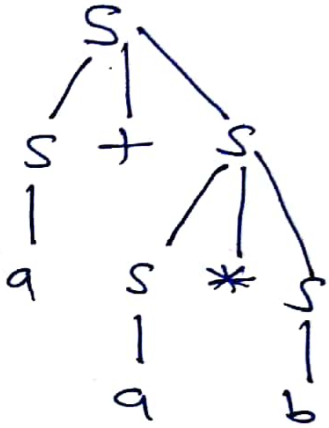
$S \to S*S$

$S \to a$

$S \to b$

Ans **LMD1:** "a+a*b"

$S \to S+S$
$\downarrow$
$a+S$ $(S \to a)$
$\downarrow$
$a+S*S$ $(S \to S*S)$
$\downarrow$ $(S \to a)$
$a+a*S$
$\downarrow$
$a+a*b$ $(S \to b)$



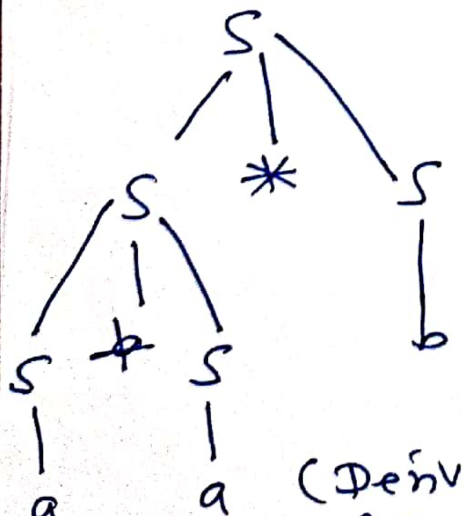
(Derivation tree-1)

~~LMD2:~~ ~~(a+a*b)~~

~~S → S*S~~
~~S → a*S (S→a)~~
~~a*~~

LMD2: string "a+a*b

$S \to S*S$
$\downarrow$
$\to S+S*S$ $(S \to S+S)$
$\downarrow$ $(S \to a)$
$a+S*S$
$\downarrow$
$a+a*S$ $(S \to a)$
$a+a*b$ $(S \to b)$



(Derivation tree-2)

since two different derivation tree for same string so grammar is ambiguous.

Simplification of CFG (Reduction of CFG)

Simplification process of Context free grammar involves elimination of the following in order

(1) Elimination of null (λ) production

$A \rightarrow \lambda$

Nonterminal   null string

(2) Elimination of unit production

$A \rightarrow B$

single Non terminal   single non terminal

(3) Elimination of Useless ~~symbols~~ production

(i) Those production, that are not used in the derivation of strings.

(ii) Those production, from which strings can not be derived. (Not terminating strings).

Questions may be asked in the examination to remove only useless production, only unit production or only null productions.

---

ex:1 Remove the useless production from the grammar

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b$
$E \rightarrow c$

Ans (i) Since $E \rightarrow c$ production is not used in the derivation of strings, so eliminate it.

$$\begin{bmatrix} S \rightarrow AB \\ A \rightarrow a \\ B \rightarrow b \end{bmatrix}$$

Ans

ex:2 Eliminate null production from the CFG

$S \rightarrow aS \mid AB \quad (S \rightarrow aS,\ S \rightarrow AB)$

$A \rightarrow \lambda$

$B \rightarrow \lambda$

$D \rightarrow b$

Ans

$S \rightarrow AB$

$S \rightarrow B$ (put $A \rightarrow \lambda$) ~~[Don't put both A→λ, B→λ same time as create S→λ]~~

$S \rightarrow AB$

$S \rightarrow A$ (put $B \rightarrow \lambda$)

Now if we put both $A \rightarrow \lambda$, $B \rightarrow \lambda$

in $S \rightarrow AB$ then

$S \rightarrow \lambda \cdot \lambda$

X $S \rightarrow \lambda$ (But we need to eliminate it)

So put $S \rightarrow \lambda$ in

$S \rightarrow aS$

$S \rightarrow a \cdot \lambda$

$S \rightarrow a$

So New productions are (final)

$S \rightarrow aS \mid AB$

$D \rightarrow b$

$S \rightarrow B$

$S \rightarrow A$

$S \rightarrow a$

Elimination of unit production

ex Eliminate unit production from the given CFG.

$S \to AB$
$A \to a$
$B \to C \mid b$ $(B \to C,\ B \to b)$
$C \to D$
$D \to E$
$E \to a$

Ans There are 3 unit productions that need to be eliminated

$B \to C$
$C \to D$
$D \to E$

As we know that

$E \to a$, so
$D \to E$ is
$D \to a$

Since $C \to D$, so put value of D
$C \to a$

Now $B \to C$, put value of C
$B \to a$

So New final productions are

$$\begin{bmatrix} S \to AB \\ A \to a \\ B \to a \mid b \\ C \to a \\ D \to a \\ E \to a \end{bmatrix}$$

Simplify (Reduce) the following CFG

$S \rightarrow aA \mid aBB$ $(S \rightarrow aA,\ S \rightarrow aBB)$

$A \rightarrow aaA \mid \lambda$ $(A \rightarrow aaA,\ A \rightarrow \lambda)$

$B \rightarrow bB \mid bbC$ $(B \rightarrow bB,\ B \rightarrow bbC)$

$C \rightarrow B$

Ans Since we have to reduce (simplify grammar), so we will perform simplification in the following order.

(i) Elimination of null (λ) production)

(ii) Elimination of unit production

(iii) Elimination of useless symbols.

(i) Elimination of null production

since it has $A \rightarrow \lambda$ null production, so put $A \rightarrow \lambda$ in production having A

$S \rightarrow aA$

$S \rightarrow a$ $(A \rightarrow \lambda)$

$A \rightarrow aaA$

$A \rightarrow aa$ $(A \rightarrow \lambda)$

After λ-production elimination

$$\boxed{\begin{array}{l} S \rightarrow aA \mid aBB \mid a \\ A \rightarrow aaA \mid aa \\ B \rightarrow bB \mid bbC \\ C \rightarrow B \end{array}}$$

(ii) Elimination of unit production

since it has only one unit production

$C \rightarrow B$

$C \rightarrow bB$ (so put value of B in it to eliminate unit production)

$C \rightarrow B$

$C \rightarrow bbC$

Updated Grammar

$S \rightarrow aA \mid aBB \mid a$

$A \rightarrow aaA \mid aa$

$B \rightarrow bB \mid bbC$

$C \rightarrow bB \mid bbC$

(iii) Elimination of useless production

$$S \to aA \mid aBB \mid a$$
$$A \to aaA \mid aa$$
$$B \to bB \mid bbC$$
$$C \to bB \mid bbC$$

In this, we need to find those useless productions that are not deriving terminal strings, or those productions that are not used in derivation of strings.

$S \to aA$ ✓
$\to a \cdot aa$ (derive string) ($A \to aa$)

$S \to aA$
$\Rightarrow a\,aaA$ ($A \to aaA$) ✓
$\Rightarrow aaa\,aa$

So $S \to aA$, $A \to aaA$, $A \to aa$ are useful

Now $S \to a$ is also useful, as it is driving strings.

Now $S \to aBB$ X
$\to a\,bB\,B$ ($B \to bB$) X
$\to abBB$
$ab\,bbC\,B$ ($B \to bbC$) X
$\to abbbCB$
$\to abbb\,bB\,B$ ($C \to bB$) X
$\to abbbb\,bbC\,B$ ($B \to bbC$) X
$\to abbbbbbbCB$ (~~C→bb~~)
$\to$ Not driving Strings ($C \to bbC$) X

Updated Grammar

$S \to aA \mid a$
$A \to aaA \mid aa$
is simplified Grammar

Conversion from CFG to CNF

CNF (Chomsky Normal Form):

A CFG is said to be in Chomsky Normal Form (CNF) if it consists of productions having following rules.

$\boxed{A \to BC}$ or $\boxed{A \to a}$

- $A \to BC$: A — single Non-terminal; BC — two non terminal
- $A \to a$: A — single Non terminal; a — single terminal

example Convert the following CFG into CNF

$S \to bA \mid aB \quad (S \to bA,\ S \to aB)$

$A \to bAA \mid aS \mid a \quad (A \to bAA,\ A \to aS,\ A \to a)$

$B \to aBB \mid bS \mid b \quad (B \to aBB,\ B \to bS,\ B \to b)$

Ans First we need to check null (λ) production, unit production and useless production, if exist, remove them.

Since in the given CFG, there is no null production, unit production & useless production

So we can convert it now.

$S \to bA,\ S \to aB$, as we know $A \to a,\ B \to b$

$S \to BA,\ S \to AB$ (CNF)

Now $A \to bAA,\ A \to aS,\ A \to a$

$A \to BAA,\ A \to AS \quad A \to a$

$A \to SA,\ A \to AS,\ A \to a$ (CNF)

Now $B \to aBB,\ B \to bS,\ B \to b$

$B \to ABB,\ B \to BS \quad B \to b$

$B \to SB,\ B \to BS,\ B \to b$ (CNF)

$S \to BA$, $S \to AB$
$A \to SA$, $A \to AS$, $A \to a$
$B \to SB$, $B \to BS$, $B \to b$

ex: 2 (2016-17)

Convert the following CFG to CNF

$S \to ASA \mid aB$
$A \to B \mid S$
$B \to b \mid \lambda$

($S \to ASA$, $S \to aB$, $A \to B$, $A \to S$, $B \to b$, $B \to \lambda$)

Ans First need to remove
(i) λ-production
(ii) unit production
(iii) useless production if any

$$\boxed{\begin{array}{l} S \rightarrow BA,\ S \rightarrow AB \\ A \rightarrow SA,\ A \rightarrow AS,\ A \rightarrow a \\ B \rightarrow SB,\ B \rightarrow BS,\ B \rightarrow b \end{array}}$$

ex:2 (2016-17)

Convert the following CFG to CNF

$S \rightarrow ASA \mid aB$ ($S \rightarrow ASA$,
$A \rightarrow B \mid S$ $S \rightarrow aB$
$B \rightarrow b \mid \lambda$ $A \rightarrow B$
$A \rightarrow S$
$B \rightarrow b$
$B \rightarrow \lambda$)

Ans First need to remove
(i) λ-production
(ii) unit production
(iii) useless production if any

(i) Remove λ-production ($B \rightarrow \lambda$)

$S \rightarrow ASA \mid aB \mid a$ (putting $B \rightarrow \lambda$ in $S \rightarrow aB$)

$A \rightarrow B \mid S$

$B \rightarrow b$

(ii) Remove unit production ($A \rightarrow B$, $A \rightarrow S$)

$S \rightarrow ASA \mid aB \mid a$ ✓

$\rightarrow$ ~~A→B~~ $A \rightarrow b \mid ASA \mid aB \mid a$ ✓ putting ($B \rightarrow b$) ($S \rightarrow ASA \mid aB \mid a$)

$B \rightarrow b$ ✓

(iii) Remove useless production.

There is no useless production

$S \rightarrow ASA \mid aB \mid a$

$A \rightarrow b \mid ASA \mid aB \mid a$

$B \rightarrow b$

$S \to ASA \mid aB \mid a$ (16)

$A \to b \mid ASA \mid aB \mid a$

$B \to b$

Now converting into chomsky Normal form (CNF)

$\boxed{S \to ASA, S \to aB}$ & $\boxed{A \to ASA \quad A \to aB}$ are not in CNF, while Other productions are already in CNF.

So Converting them into CNF

$S \to aB$ can be

$S \to SB$ $(S \to a)$

or

$S \to aB$

$S \to AB$ $(A \to a)$

$A \to aB$ can be

$A \to SB$ $(S \to a)$

$A \to aB$

$A \to AB$ $(A \to a)$

$S \to \underbrace{AS}_{X}A$, $A \to \underbrace{AS}_{X}A$

Let $X \to AS$

$S \to XA$ $A \to XA$

$X \to AS$

So Final CNF is

$S \to XA$, $A \to XA$, $X \to AS$

$S \to SB \mid AB \mid a$

$A \to b \mid a \mid SB \mid AB$

$B \to b$

Ex: 3 2016-17 Convert the following into CNF

$S \to XY \mid Xn \mid p$

$X \to mX \mid m$

$Y \to Xn \mid o$

Ans

$S \to XY \mid p$
$X \to m$
$Y \to o$
already in CNF

$S \to Xn$, $Y \to Xn$
$X \to mX$ are not in CNF.

$S \rightarrow Xn$

$Y \rightarrow Xn$

$X \rightarrow mX$

are not in CNF

Let $Z \rightarrow n$ then

$S \rightarrow XZ$
$Y \rightarrow XZ$
$Z \rightarrow n$
} Now in CNF

for $X \rightarrow mX$ (given $X \rightarrow m$)

$X \rightarrow XX$
$X \rightarrow m$
} Now in CNF

So Final CNF is

$$\boxed{\begin{array}{l} S \rightarrow XY/\phi \\ X \rightarrow m \\ Y \rightarrow o \\ S \rightarrow XZ \\ Y \rightarrow XZ \\ Z \rightarrow n \\ X \rightarrow XX \end{array}}$$

Greibach Normal form (GNF)

A CFG is said to be in GNF, if it follows the following rules:

(i) $A \rightarrow a\,BCD\ldots$

- A → single Non-terminal
- a → single terminal
- $BCD\ldots$ → any No. of Nonterminals

or

(ii) $A \rightarrow a$

- A → single Non-terminal
- a → single terminal

There should be no null (λ) production, no unit production and no useless production.

Example: 1: Convert the following CFG to GNF.

$$S \rightarrow ABb \mid a$$
$$A \rightarrow aaA$$
$$B \rightarrow bAb$$

Ans

$S \rightarrow ABb$, $S \rightarrow a$ → already in GNF

$A \rightarrow aaA$, $B \rightarrow bAb$

for ~~Let~~ $S \rightarrow ABb$

$S \rightarrow aaABb$ (putting $A \rightarrow aaA$)

$S \rightarrow aSABX$

Let $X \rightarrow b$

for $A \rightarrow aaA$

$A \rightarrow aSA$ (as $S \rightarrow a$)

for $B \rightarrow bAb$

$B \rightarrow bAX$ (as $X \rightarrow b$)

final GNF
$S \rightarrow aSABX$
$X \rightarrow b$
$A \rightarrow aSA$
$B \rightarrow bAX$
$S \rightarrow a$
Ans

ex:2 Convert following CFG to GNF

$S \to SS$

$S \to 0S1$

$S \to 01$

Ans

for $S \to SS$

$\to 0S1S$ (putting $S \to 0S1$)

~~let X→0~~ Let $Y \to 1$

~~Y→1~~

$$\boxed{\begin{array}{l} S \to 0SYS \\ Y \to 1 \end{array}}$$

for $S \to SS$

$S \to 01S$ ($S \to 01$)

$$\boxed{\begin{array}{l} S \to 0YS \\ Y \to 1 \end{array}}$$

for $S \to 0S1$

$$\boxed{\begin{array}{l} S \to 0SY \\ Y \to 1 \end{array}}$$

for $S \to 01$

$$\boxed{\begin{array}{l} S \to 0Y \\ Y \to 1 \end{array}}$$

So final GNF =

$$\boxed{\begin{array}{l} S \to 0SYS \\ Y \to 1 \\ S \to 0YS \\ S \to 0SY \\ S \to 0Y \end{array}}$$

Ans

Scanned with CamS

Conversion from CFL to CFG

examples Convert the following into CFG

(i) $L = wcw^R \mid w \in \{a+b\}^*$

(ii) $L = ww^R \mid w \in \{a,b\}^*$

(iii) $\{L = a^n b^{2n} \mid n \geq 1\}$

(iv) $\{L = a^{2n} b^n \mid n \geq 3\}$

(v) $\{L = a^n b^n \mid n \geq 0\}$

Ans

(i) $\boxed{L = wcw^R \mid w \in \{a,b\}^*}$

$= \{c, aca, bcb, abcba, aacaa, bacab, bbcbb, aabcbaa, \ldots\}$

$S \to c$
$S \to aSa$
$S \to bSb$

(ii) $\boxed{L = ww^R \mid w \in \{a,b\}^*}$

$= \{\lambda, aa, bb, abba, baab, aaaa, bbbb, aabbaa, \ldots\}$

$S \to \lambda$
$S \to aSa$
$S \to bSb$

(iii) $\boxed{L = a^n b^{2n} \mid n \geq 1}$

$= \{abb, aabbbb, aaabbbbbb, \ldots\}$

$S \to abb$
$S \to aSbb$

(iv) $\boxed{L = a^{2n} b^n \mid n \geq 3}$

$= \{aaaaaabbb, aaaaaaaabbbb, \ldots\}$

$S \to aaaaaabbb$
$S \to aaSb$

(v) $\boxed{\{L = a^n b^n \mid n \geq 0\}}$

$= \{\lambda, ab, aabb, aaabbb, \text{------}\}$

$$\left[\begin{array}{l} S \to \lambda \\ S \to aSb \end{array}\right]$$

example 2

(i) $\boxed{L = a^n b^n \mid n \geq 1}$

Ans

$= \{ab, aabb, aaabbb, \text{------}\}$

$$\left[\begin{array}{l} S \to ab \\ S \to aSb \end{array}\right]$$

(ii) $L = \underbrace{a^n b^n}_{A} \underbrace{c^m d^m}_{B} \mid m, n \geq 1$

Ans

$$\left[\begin{array}{l} S \to AB \\ \cancel{A \to aAb} \\ A \to ab \\ A \to aAb \\ B \to cd \\ B \to cBd \end{array}\right]$$

Imp

example: 3

$\boxed{\{L = a^i b^j c^k \mid i = j \text{ or } j = k\}}$ $i, j, k \geq 1$

Ans

If $i = j = n$

$L_1 = \underbrace{a^n b^n}_{A} \underbrace{c^k}_{B}$

$$\boxed{\begin{array}{l} S_1 \to AB \\ A \to ab \\ A \to aAb \\ B \to c \\ B \to Bc \end{array}}$$ ②

If $j = k = m$

$L_2 = \underbrace{a^i}_{C} \underbrace{b^m c^m}_{D}$

$$\boxed{\begin{array}{l} S_2 \to CD \\ C \to a \\ C \to aC \\ D \to bc \\ D \to bDc \end{array}}$$ ③

$\boxed{S \to S_1 \mid S_2}$ ①

Convert the following CFG into GNF (22)

$$S \rightarrow AA \mid a$$
$$A \rightarrow SS \mid b$$

Ans First, we need to check, whether there is any useless, null or unit productions.

since in this problem, there is no null, unit or useless production, so goto next step.

as we know that in GNF

Nonterminal (NT) $\rightarrow$ Terminal (T)

OR

NT $\rightarrow$ T $NT_1 NT_2 NT_3$ ----

as we have $S \rightarrow AA$

$S \rightarrow a$ ✓ already in GNF

$A \rightarrow SS$

$A \rightarrow b$ ✓ already in GNF

So we need to convert $S \rightarrow AA$ and $A \rightarrow SS$ in GNF

first

$S \rightarrow AA$

✓ $S \rightarrow bA$ (putting $A \rightarrow b$)

$S \rightarrow AA$

$S \rightarrow SSA$ ($A \rightarrow SSA$)

✓ $S \rightarrow aSA$ ($S \rightarrow a$)

$S \rightarrow AA$

$S \rightarrow ASS$ ($A \rightarrow SS$)

~~$S \rightarrow A$~~

✓ $S \rightarrow bSS$ ($A \rightarrow b$)

Now

$A \rightarrow SS$ ($S \rightarrow a$)

✓ $A \rightarrow aS$

$A \rightarrow SS$

$A \rightarrow AAS$ ($S \rightarrow AA$)

✓ $A \rightarrow bAS$ ($A \rightarrow b$)

$A \rightarrow SS$

$A \rightarrow SAA$ ($S \rightarrow AA$)

✓ $A \rightarrow aAA$ ($S \rightarrow a$)

Final GNF

$S \rightarrow a$

$A \rightarrow b$

$S \rightarrow bA$

$S \rightarrow aSA$

$S \rightarrow bSS$

$A \rightarrow aS$

$A \rightarrow bAS$

$A \rightarrow aAA$