Example 4.1 : Consider the grammar G whose productions are

$$S \rightarrow aAS \mid a$$
$$A \rightarrow SbA \mid SS \mid ba.$$

Show that S derives *aabbaa* and construct the derivation tree whose yield is *aabbaa*.

Solution : Given :

$$S \rightarrow aAS \mid a$$
$$A \rightarrow SbA \mid SS \mid ba$$

$$\begin{aligned} S &\rightarrow aAS & & \\ &\rightarrow aSbAS & & (A \rightarrow SbA) \\ &\rightarrow aabAS & & (S \rightarrow a) \\ &\rightarrow aabbaS & & (A \rightarrow ba) \\ &\rightarrow aabbaa & & (S \rightarrow a) \end{aligned} \qquad \text{....(4.1)}$$

Hence, S derives aabbaa.

The derivation tree is given below :

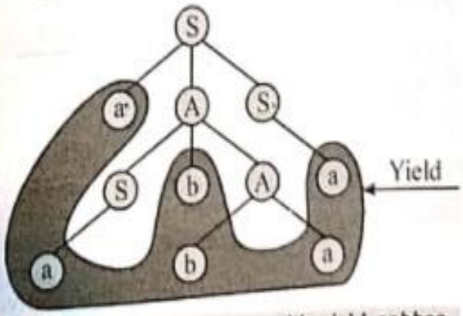


Fig. 4.2 : Derivation tree with yield *aabbaa*.

Another derivation for ***aabbaa*** is :

$$\begin{aligned} S &\rightarrow aAS & & \\ &\rightarrow aAa & & (S \rightarrow a) \\ &\rightarrow aSbAa & & (A \rightarrow SbA) \\ &\rightarrow aSbbaa & & (A \rightarrow ba) \\ &\rightarrow aabbaa & & (S \rightarrow a) \end{aligned} \qquad \text{....(4.2)}$$

The **derivation tree** corresponding to the above productions is :

The **derivation tree** corresponding to the above productions is :

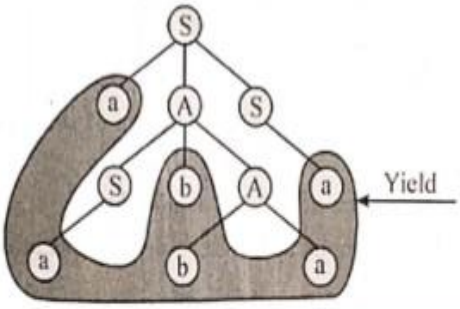


Fig. 4.3 : Derivation tree with yield *aabbaa*.

One more derivation for the string **aabbaa** is :

$$S \rightarrow aAS$$
$$\rightarrow aSbAS \qquad (A \rightarrow SbA)$$
$$\rightarrow aSbAa \qquad (S \rightarrow a)$$
$$\rightarrow aabAa \qquad (S \rightarrow a)$$
$$\rightarrow aabbaa \qquad (A \rightarrow ba) \qquad ....(4.3)$$

The **derivation tree** for the productions is :

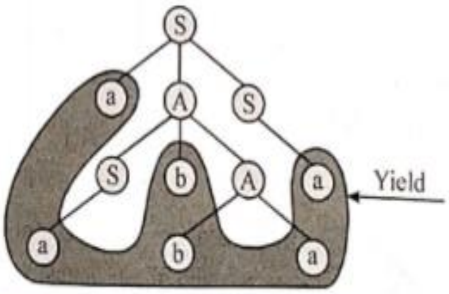


Fig. 4.4 : Derivation tree with yield *aabbaa*.

Example 4.2 : Consider the grammar G with set of production rules P given below:

$$S \rightarrow 0B / 1A$$
$$A \rightarrow 0 / 0S / 1AA$$
$$B \rightarrow 1 / 1S / 0BB$$

Find :

(a) Leftmost derivation

(b) Rightmost derivation

(c) Derivation Tree

For the string $W = 00110101$.

Solution :

(a) Leftmost derivation :

$$
\begin{aligned}
S &\rightarrow 0B & \\
&\rightarrow 00BB & (B \rightarrow 0BB) \\
&\rightarrow 001B & (B \rightarrow 1) \\
&\rightarrow 0011S & (B \rightarrow 1S) \\
&\rightarrow 00110B & (S \rightarrow 0B) \\
&\rightarrow 001101S & (B \rightarrow 1S) \\
&\rightarrow 0011010B & (S \rightarrow 0B) \\
&\rightarrow 00110101 & (B \rightarrow 1) \qquad \text{....(4.4)}
\end{aligned}
$$

(b) Rightmost Derivation :

$$
\begin{aligned}
S &\rightarrow 0B & \\
&\rightarrow 00BB & (B \rightarrow 0BB) \\
&\rightarrow 00B1S & (B \rightarrow 1S) \\
&\rightarrow 00B10B & (S \rightarrow 0B) \\
&\rightarrow 00B101S & (B \rightarrow 1S)
\end{aligned}
$$

$$
\begin{aligned}
&\rightarrow 00B1010B & (S \rightarrow 0B) \\
&\rightarrow 00B10101 & (B \rightarrow 1)
\end{aligned}
$$

$\rightarrow 00110101 \qquad (B \rightarrow 1)$

(c) Derivation Tree : ...(5.5)

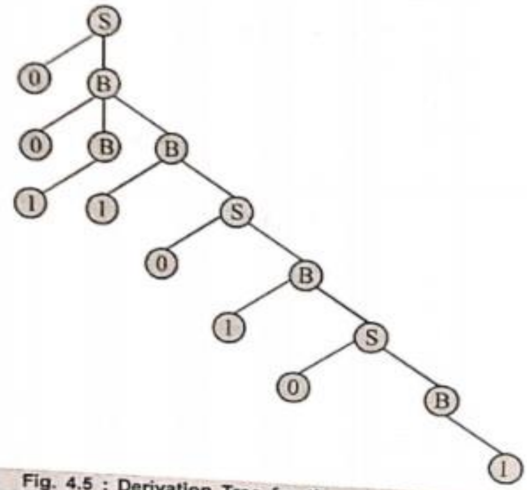


Fig. 4.5 : Derivation Tree for the string 00110101.

4.4.5 Ambiguity in Context Free G

Example 4.3 : Consider the Context Free Grammar (CFG) G whose productions are $S \rightarrow SbS/a$. Show that the grammar is ambiguous.

Solution : To prove that the given Grammar G is ambiguous. Let us consider the string w = abababa We can derive this string by applying production rules as follows :

$$
\begin{aligned}
S &\rightarrow SbS \\
&\rightarrow abS && (S \rightarrow a) \\
&\rightarrow abSbS && (S \rightarrow SbS) \\
&\rightarrow ababS && (S \rightarrow a) \\
&\rightarrow ababSbS && (S \rightarrow SbS) \\
&\rightarrow abababS && (S \rightarrow a) \\
&\rightarrow abababa && (S \rightarrow a)
\end{aligned}
$$

The derivation tree for the string w = abababa by using above productions is shown in Fig. 4.6.

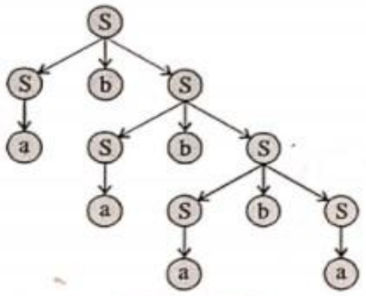


Fig. 4.6

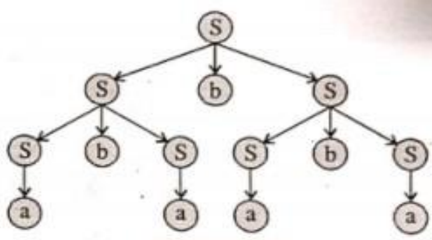


Fig. 4.7

We can derive this string by applying the productions in this way also.

$$
\begin{aligned}
S &\rightarrow SbS \\
&\rightarrow SbSbS && (S \rightarrow SbS) \\
&\rightarrow abSbS && (S \rightarrow a) \\
&\rightarrow ababS && (S \rightarrow a) \\
&\rightarrow ababSbS && (S \rightarrow SbS) \\
&\rightarrow abababS && (S \rightarrow a) \\
&\rightarrow abababa && (S \rightarrow a)
\end{aligned}
$$

The derivation tree corrosponding to the above production is shown in Fig. 4.7, so there exists more than one derivation trees for the string hence, the grammar is ambiguous.

Example 4.4 : Show that the following *CFG* is ambiguous.

$$S \rightarrow aSbS \mid bSaS \mid \wedge$$

Solution :

Given : $S \rightarrow aSbS \mid bSaS \mid \wedge$

To prove : Grammar is ambiguous.

Let us consider the string $w = abab$. We can derive the string by applying the productions as follows :

$$\begin{aligned} S &\rightarrow aSbS & \\ &\rightarrow abSaSbS & (S \rightarrow bSaS) \\ &\rightarrow abaSbS & (S \rightarrow \wedge) \\ &\rightarrow ababS & (S \rightarrow \wedge) \\ &\rightarrow abab & (S \rightarrow \wedge) \end{aligned}$$

The corrosponding, derivation tree is :

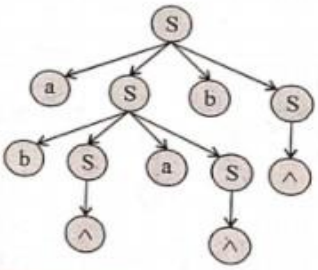


Fig. 4.8 : Derivation tree.

We can also derive the string $w = abab$ by applying the productions in this way.

$$\begin{aligned} S &\rightarrow aSbS & \\ &\rightarrow aSbaSbS & (S \rightarrow aSbS) \\ &\rightarrow abaSbS & (S \rightarrow \wedge) \\ &\rightarrow ababS & (S \rightarrow \wedge) \\ &\rightarrow abab & (S \rightarrow \wedge) \end{aligned}$$

The derivation tree by using above production is :

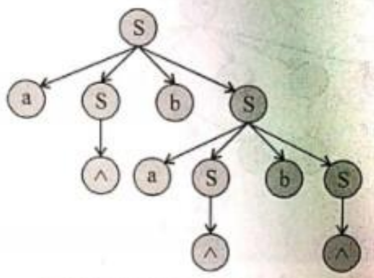


Fig. 4.9 : Derivation tree.

Since, there are more than one deviation tree for the string, w = abab, hence, the grammar is **ambiguous**.

Example 4.5 : Show that the grammar given below is ambiguous.

$$S \rightarrow iCtS$$
$$S \rightarrow iCtSeS$$
$$S \rightarrow a$$
$$C \rightarrow b$$

Solution : **Given :**

$$S \rightarrow iCtS$$
$$S \rightarrow iCtSeS$$
$$S \rightarrow a$$
$$C \rightarrow b$$

To prove : The grammar is ambiguous.

Let us consider the string w = ***ibtibtibtaea***

We can derive the string $w = ibtibtibtaea$ as follows:

(i) $S \rightarrow iCtSeS$

$\rightarrow ibt\mathbf{S}eS$	$(C \rightarrow b)$
$\rightarrow ibti\mathbf{CtS}es$	$(S \rightarrow iCtS)$
$\rightarrow ibtibt\mathbf{S}eS$	$(C \rightarrow b)$
$\rightarrow ibtibti\mathbf{C}tSeS$	$(S \rightarrow iCtS)$
$\rightarrow ibtibtibt\mathbf{S}eS$	$(C \rightarrow b)$
$\rightarrow ibtibtibtae\mathbf{S}$	$(S \rightarrow a)$
$\rightarrow ibtibtibtaea$	$(S \rightarrow a)$

Derivation tree for the above productions :

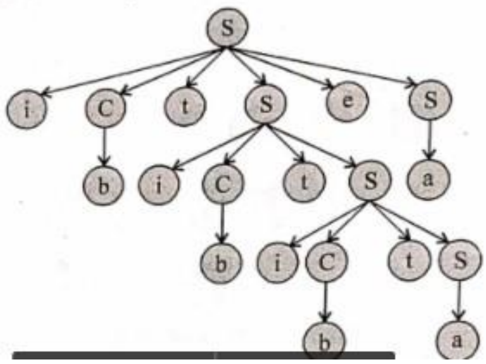


Fig. 4.10 : Derivation tree.

(ii)

$$\begin{aligned} S &\rightarrow iCtS \\ &\rightarrow ibt\boldsymbol{S} && (C \rightarrow b) \\ &\rightarrow ibti\boldsymbol{C}tS && (S \rightarrow iCtS) \\ &\rightarrow ibtibt\boldsymbol{S} && (C \rightarrow b) \\ &\rightarrow ibtibti\boldsymbol{C}tSeS && (S \rightarrow iCtSeS) \\ &\rightarrow ibtibtibt\boldsymbol{S}eS && (C \rightarrow b) \\ &\rightarrow ibtibtibtae\boldsymbol{S} && (S \rightarrow a) \\ &\rightarrow ibtibtibtaea && (S \rightarrow a) \end{aligned}$$

The derivation tree

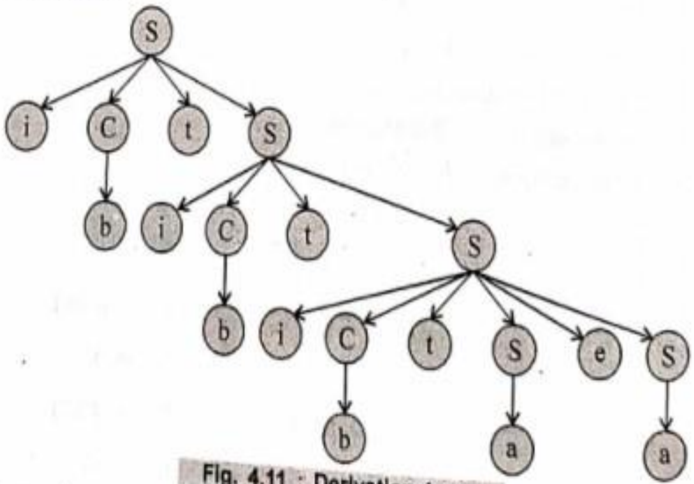


Fig. 4.11 : Derivation tree.

Since, there exists more than one derivation tree for the string w = ***ibtibtibtaea*** hence, the **grammar** is ambiguous.

Example 4.6 : Check whether the grammar $G = (V_N, \Sigma, P, S)$ is ambiguous or not.

where,
$$V_N = \{S, A\}$$
$$\Sigma = \{a, b\}$$
$$P = \{S \rightarrow AA$$
$$A \rightarrow AAA$$
$$A \rightarrow a$$
$$A \rightarrow bA$$
$$A \rightarrow Ab\}$$

Solution : Let us consider the string $w = babbab$

Let us derive the string $w = babbab$ using above production rules:

(i)
$$\begin{aligned} S &\rightarrow AA \\ &\rightarrow AbA && (A \rightarrow Ab) \\ &\rightarrow bAbA && (A \rightarrow bA) \\ &\rightarrow babA && (A \rightarrow a) \\ &\rightarrow babbA && (A \rightarrow bA) \\ &\rightarrow babbAb && (A \rightarrow Ab) \\ &\rightarrow babbab && (A \rightarrow a) \end{aligned}$$

Derivation tree for the above production is :

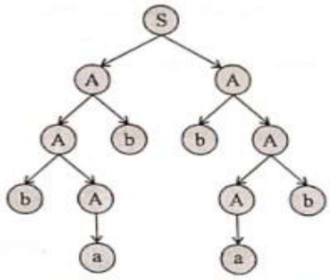


Fig. 4.12 : Derivation tree.

(ii)
$$\begin{aligned} S &\rightarrow AA \\ &\rightarrow bAA && (A \rightarrow bA) \\ &\rightarrow baA && (A \rightarrow a) \\ &\rightarrow baAb && (A \rightarrow Ab) \\ &\rightarrow babAb && (A \rightarrow bA) \\ &\rightarrow babbAb && (A \rightarrow bA) \\ &\rightarrow babbab && (A \rightarrow b) \end{aligned}$$

Derivation tree for the above productions is :

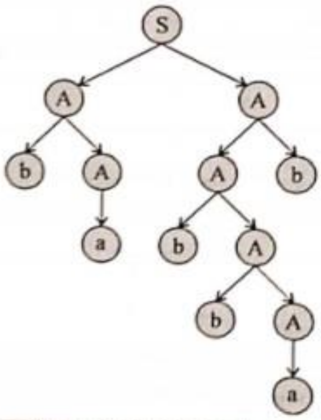


Fig. 4.13 : Derivation tree.

Since, there exits more than one derivation tree for the string hence the grammar is **ambiguous**.

Example 4.7 : Check whether the

... the grammar is ambiguous.

Example 4.7 : Check whether the grammar

$$S \rightarrow a \mid abSb \mid aAb$$

$$A \rightarrow bS \mid aAAb$$ **is ambiguous or not.**

Solution : Given :

$$S \rightarrow a \mid abSb \mid aAb$$

$$A \rightarrow bS \mid aAAb$$

Let us consider the string $w = ababab b$ and derive the string from above production rules.

(i)

$$S \rightarrow abSb$$

$$\rightarrow abaAbb \qquad (S \rightarrow aAb)$$

$$\rightarrow ababSbb \qquad (A \rightarrow bS)$$

$$\rightarrow abababb \qquad (S \rightarrow a)$$

Derivation tree for the production rules :

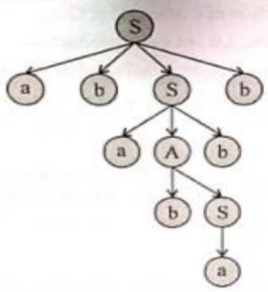


Fig. 4.14 : Derivation tree.

(ii) The string $w = abababb$ can also be derived as :

$$S \rightarrow aAb$$

$$\rightarrow abSb \qquad (A \rightarrow bS)$$

$$\rightarrow ababSbb \qquad (S \rightarrow abSb)$$

$$\rightarrow abababb \qquad (S \rightarrow a)$$

... the grammar is ambiguous.

Example 4.7 : Check whether the grammar

$$S \rightarrow a \,/\, abSb \,/\, aAb$$

$$A \rightarrow bS \,/\, aAAb$$ **is ambiguous or not.**

Solution : Given :

$$S \rightarrow a \,/\, abSb \,/\, aAb$$

$$A \rightarrow bS \,/\, aAAb$$

Let us consider the string $w = ababab b$ and derive the string from above production rules.

(i) $S \rightarrow abSb$

$\rightarrow abaAbb \qquad (S \rightarrow aAb)$

Scanned by CamScanner

$\rightarrow ababSbb \qquad (A \rightarrow bS)$

$\rightarrow ababab b \qquad (S \rightarrow a)$

Derivation tree for the production rules :

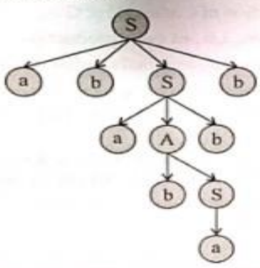


Fig. 4.14 : Derivation tree.

(ii) The string $w = abababb$ can also be derived as :

$S \rightarrow aAb$

$\rightarrow abSb \qquad (A \rightarrow bS)$

$\rightarrow ababSbb \qquad (S \rightarrow abSb)$

$\rightarrow abababb \qquad (S \rightarrow a)$

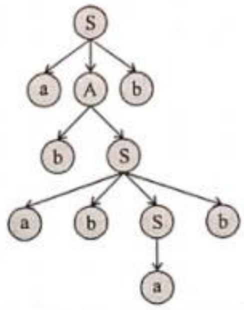


Fig. 4.15 : Derivation tree.

Since, there exists more than one derivation tree for $w = abababb$, hence, the grammar is **ambiguous.**

Example 4.8 : Consider the Context Free Grammar (CFG) G

$$S \rightarrow AB \mid a$$
$$A \rightarrow b$$

Eliminate the useless symbols from the above Context Free Grammar (CFG).

Solution :

Given :

$$S \rightarrow AB \mid a$$
$$A \rightarrow b$$

1. Identify non-generating symbols

Observing each production in the CFG, it becomes very clear that B does not derive terminal string while A and S both derive the terminal string, i.e., $A \rightarrow b$ and $S \rightarrow a$ respectively. Hence, B is non-generating.

$\therefore$ We remove the production $S \rightarrow AB$ from the grammar, now the CFG becomes

$$S \rightarrow a$$
$$A \rightarrow b$$

2. Identify non-reachable symbols

Here, A is non-reachable symbol, as it can not be reached by starting symbols S. Hence, we remove the production.

$$A \rightarrow b$$

Now, the CFG becomes

$$S \rightarrow a$$

Which is the required Reduced Grammar.

Example 4.9 : Consider the following Context Free Grammar (CFG) G whose productions are :

$$S \rightarrow AB \mid CA$$
$$A \rightarrow a$$
$$B \rightarrow BC \mid AB$$
$$C \rightarrow aB \mid b$$

Find the reduced grammar equivalent to the above grammar G.

Solution : Given

$$S \rightarrow AB \mid CA$$

$$A \rightarrow a$$
$$B \rightarrow BC \mid AB$$
$$C \rightarrow aB/b$$

1. Identify non-generating symbols

(i) Non-terminal A **is generating** as there is a production $A \rightarrow a$.

(ii) Non-terminal **C is generating** as there is a production $C \rightarrow b$.

(iii) Non-terminal **S is generating** as there is a production. $S \rightarrow CA$ and non-terminals C and A both derive the terminal string, i.e., $C \rightarrow$ b and $A \rightarrow$ a.

(iv) Non-terminal **B is non-generating** because B does not derive any terminal string.

The set of non-generating symbol = $\{B\}$.

Removing the productions involving the non-generating symbol B, we get the following grammar.

$$S \rightarrow CA$$
$$A \rightarrow a$$
$$C \rightarrow b$$

2. Identify non-reachable symbols

(i) Non-terminal S is reachable as it is the **start symbol.**

(ii) Non-terminal A and C are also reachable as there is a production

$$S \rightarrow CA.$$

Therefore, no production is removed.

$\therefore$ The required reduced grammar is

$$S \rightarrow CA$$
$$A \rightarrow a$$
$$C \rightarrow b$$

Example 4.10 : Consider the context free grammar G whose productions are given as:

$$S \to XY / 0$$
$$X \to 1$$

Find the reduced grammar equivalent to the above grammar G.

Solution :

Given :

$$S \to XY / 0$$
$$X \to 1$$

1. Identify non-generating symbols :

(i) Non-terminal X **is generating** as there is a production $X \to 1$.

(ii) Non-terminal S **is also generating** as there is a production $S \to 0$.

(iii) Non-terminal **Y is non-generating** as Y does not derive any terminal string

$\therefore$ Set of non-generating symbols = $\{Y\}$.

Removing the productions involving the non-generating symbol Y, we get,

$$S \to 0$$
$$X \to 1$$

2. Identify non-reachable symbols :

(i) S is included in the reachable symbols as S is the start symbol.

(ii) X is non-reachable as there is no way to reach X from S.

Set of non-reachable symbols = $\{X\}$.

So, the production corresponding to X, i.e., $X \to 1$ is removed from the final grammar.

$\therefore$ The reduced grammar is :

$$S \to 0$$

Example 4.11 : Remove useless symbols from the following context free grammar G whose productions are:

ग

$$S \to aA \mid bB$$
$$A \to aA \mid a$$
$$B \to bB$$
$$D \to ab \mid Ea$$
$$E \to aC \mid d$$

Solution : Given :

$$S \to aA \mid bB$$
$$A \to aA \mid a$$
$$B \to bB$$
$$D \to ab \mid Ea$$
$$E \to aC \mid d$$

1. Identify non-generating symbols.

(i) Non-terminal A **is generating** as there is a production $A \to a$.

(ii) Non-terminal D **is generating** as there is a production $D \to ab$

(iii) Non-terminal E **is generating** as there is a production $E \to d$.

(iv) Non-terminal S **is also generating** as there is a production $S \to aA$ and non terminal A on RHS is a generating symbol.

(v) Non-terminal B **is non-generating** as the production $B \to bB$ will be continuously in loop and will not derive any non terminal.

(vi) Non-terminal C **is non-generating** as it does not derive any terminal string.

$\therefore$ Set of non-generating symbols = $\{B, C\}$.

Removing the productions involving the non-generating symbol B and C, we get,

$$S \to aA$$
$$A \to aA \mid a$$
$$D \to ab \mid Ea$$
$$E \to d$$

2. Identify non-reachable symbols :

(i) S is included in reachable symbol **as S is the start symbol.**

(ii) $S \to aA$ is a production means **S derives non-terminal** so A is also reachable.

(iii) A does not derive any non-terminal symbol

$\therefore$ Set of **reachable symbol** = $\{S, A\}$ and

Set of **non-reachable symbol** = $\{D, E\}$.

On removing the productions corrosponding to the **non-reachable symbols D and E**, we get

$$S \to aA$$
$$A \to aA \mid a$$

Which is the reduced grammar.

Example 4.12 : Consider the context-free grammar G whose productions are given as:

$$S \to aS \mid A \mid C$$
$$A \to a$$
$$B \to aa$$
$$C \to aCb$$

Find the reduced grammar equivalent to the grammar G.

Solution : Given :

$$S \to aS \mid A \mid C$$
$$A \to a$$
$$B \to aa$$
$$C \to aCb$$

1. Identify the non-generating symbol :

(i) Non-terminal A **is generating** as there is a production $A \to a$.

(ii) Non-terminal B **is generating** as there is a production $B \to aa$.

(iii) Non-terminal S **is generating** as there is a production $S \to A$ and non-terminal A on RHS is generating.

(iv) Non-terminal C **is non-generating** as the production $C \to aCb$ will be continuously in the loop and does not derive any string.

$\therefore$ **Set of non-generating symbol = $\{C\}$.**

Removing the production involving the non-generating symbol C, we get

$$S \to aS \mid A$$
$$A \to a$$
$$B \to aa$$

2. Identify non reachable symbols :

(i) S is included in reachable symbol as it is the **start symbol.**

(ii) $S \to A$ is a production, i.e., S derives non-terminal A so A **is also reachable.**

(iii) A does not derive any non-terminal symbol.

$\therefore$ **Set of reachable symbol** = $\{S, A\}$, and

Set of non-reachable symbol = $\{B\}$.

On removing the productions corrosponding to the **non-reachable symbols B**, we get

$$S \to aS \mid A$$
$$A \to a$$

Which is the required reduced grammar.

Example 4.13 : Eliminate the useless symbols from the following grammar.

$$S \rightarrow aA \,/\, a \,/\, Bb \,/\, cC$$
$$A \rightarrow aB$$
$$B \rightarrow a \,/\, Aa$$
$$C \rightarrow cCD$$
$$D \rightarrow ddd$$

Solution : Given :

$$S \rightarrow aA \,/\, a \,/\, Bb \,/\, cC$$
$$A \rightarrow aB$$
$$B \rightarrow a \,/\, Aa$$
$$C \rightarrow cCD$$
$$D \rightarrow ddd$$

1. Identify non-generating symbols

(i) Non-terminal D **is a generating** symbol as there is a production $D \rightarrow ddd$.

(ii) Non-terminal B **is a generating** symbol as there is a production $B \rightarrow a$.

(iii) Non-terminal A **is generating** as there is a production $A \rightarrow aB$ and non-terminal B on RHS is generating symbol.

(iv) Non-terminal S **is generating** as there is a production $S \rightarrow a$.

(v) Non-terminal C **is non-generating** as the production $C \rightarrow cCD$ will be in loop and does not derive string.

(C on RHS can not be replaced by any terminal).

$\therefore$ Set of non-generating symbol = $\{C\}$.

Removing the production involving the non-generating symbol C, we get,

$$S \rightarrow aA \,/\, a \,/\, Bb$$
$$A \rightarrow aB$$
$$B \rightarrow a \,/\, Aa$$
$$D \rightarrow ddd$$

2. Identify non-reachable symbol :

(i) S is included in the reachable symbol as **S is the start symbol.**

(ii) $S \rightarrow aA$ is a production, i.e., S derives non-terminal A, hence A is also **reachable.**

(iii) $S \rightarrow Bb$ is a production so, B is also **reachable.**

(iv) Non-terminal S, A and B does not derive any new non-terminal.

$\therefore$ **Set of reachable symbol** = $\{S, A, B\}$ and

Set of non-reachable symbol = $\{D\}$.

On reamoving the productions corrosponding to the non-reachable symbol D, we get,

$$S \rightarrow aA \,/\, a \,/\, Bb$$
$$A \rightarrow aB$$
$$B \rightarrow a \,/\, Aa$$

Which is the required reduced grammar.

any nullable variable is also included in the grammar.

Example 4.14 : Consider the following Context Free Grammar (CFG) whose productions are:

$$S \to aA$$
$$A \to b / \wedge$$

Eliminate the null productions.

Solution : Given :

$$S \to aA$$
$$A \to b / \wedge$$

1. Nullable variables

Since, $A \to \wedge$ is a production, therefore, A is the nullable variable.

$\therefore$ The set of nullable variable = $\{A\}$.

2. The productions whose RHS is a nullable variable (Here, it is A) is $S \to aA$

Replace A by $\wedge$, we get,

$S \to a$ ....(1)

$S \to aA$ (original production) ....(2)

3. The productions whose RHS **does not** contain any **nullable variable :**

$A \to b$ ....(3)

$\therefore$ The resultant grammar which has no $\wedge$ productions :

$$S \to a\,A$$
$$S \to a$$
$$A \to b$$

or

$$S \to aA / a$$
$$A \to b$$

Example 4.15 : Consider the Context Free Grammar G whose productions are:

$$S \to aS / AB$$
$$A \to \wedge$$
$$B \to \wedge$$
$$D \to b$$

Construct the equivalent grammar without null productions.

Solution : Given

$$S \to aS / AB$$
$$A \to \wedge$$
$$B \to \wedge$$
$$D \to b.$$

1. Nullable Variables

(i) $A \to \wedge$ is a production, therefore, A **is nullable** variable

(ii) $B \to \wedge$ is a production, therefore, B **is a nullable** variable

(iii) $S \to AB$ is a production and A and B are both nullable variable, therefore, S **is a** also **nullable variable**

$\therefore$ Set of nullable variable = $\{S, A, B\}$.

2. The productions whose RHS is a nullable variable :

$$S \to aS$$
$$S \to AB$$

(i) For $S \to a\,S$ replace S by $\wedge$ in RHS, We get,

$S \to a$ ...(1)

$S \to aS$ (original production) ...(2)

(ii) For $S \to AB$ in this case, we **can not** erase both nullable variables A and B in $S \to AB$ as we will get $S \to \wedge$. Hence, we get two productions.

$S \to A$ ...(3)

$S \to B$ ...(4)

3. The productions whose RHS does not contain nullable variables :

$D \to b$ ...(5)

$\therefore$ The resultant grammar which has no-null productions is the grammar having productions from equation (1) to (5), i.e.,

$$S \to aS$$
$$S \to a$$
$$S \to A$$
$$S \to B$$
$$D \to b$$

or

$$S \to aS / a / A / B$$
$$D \to b$$

$D \to b$

Example 4.16 : Consider the CFG whose productions are given as :

$S \to XYX$

Scanned by CamScanner

$X \to 0X/\wedge$

$Y \to 1Y/\wedge$

Eliminate the null productions from above grammar.

Solution : Given :

$S \to XYX$

$X \to 0X/\wedge$

$Y \to 1Y/\wedge$

1. Find nullable variables

$\because$ $X \to \wedge$ and

$Y \to \wedge$

$\therefore$ set of nullable variable = $\{X, Y\}$

2. The productions whose RHS is nullable variable :

(i) $S \to XYX$

In this case, we can not erase both nullable variable as we will get $S \to \wedge$. We can erase at most two variables, hence, we get following set of productions.

$S \to XY$ ....(1)

$S \to YX$ ....(2)

$S \to XX$ ....(3)

$S \to X$ ....(4)

$S \to Y$ ....(5)

(ii) $X \to 0X$

Replacing X on RHS by $\wedge$, we get

$X \to 0$ ....(6)

$X \to 0X$ (Original Production) ....(7)

(iii) $Y \to 1Y$

Replacing Y on RHS by $\wedge$, we get,

$Y \to 1$ ....(8)

$Y \to 1Y$ (Original Production) ....(9)

3. The production whose RHS does not contain any nullable variable :

No production exist under this category.

The **resultant grammar** which is free from null production is the set of production from (i) to (ix), i.e.,

$S \to XY/YX/XX/X/Y$

$X \to 0X/0$

$Y \to 1Y/1$

Example 4.17 : Eliminate the $\wedge$-productions from the CFG given below.

$$A \to 0B1/1B1$$
$$B \to 0B/1B/\wedge$$

Solution : Given

$$A \to 0B1/1B1$$
$$B \to 0B/1B/\wedge$$

1. Find nullable variable :

Since, $B \to \wedge$ is a production, therefore, B is a nullable variable

The set of nullable variable = $\{B\}$

2. The productions whose RHS is a nullable variable :

(i) $A \to 0B1$

Replacing b on RHS by $\wedge$, we get

$A \to 01$ ...(1)

$A \to 0B1$ (Original Production) ...(2)

(ii) $A \to 1B1$

Replacing B on RHS by $\wedge$, we get,

$A \to 11$ ...(3)

$A \to 1B1$ (Original Production) ...(4)

(iv) $B \to 0B$

Replacing B on RHS by $\wedge$, we get

$B \to 0$ ...(5)

$B \to 0B$ (Original Production) ...(6)

(iv) $B \to 1B$

Replacing B on RHS by $\wedge$, we get

$B \to 1$ ...(7)

$B \to 1B$ (Original Production) ...(8)

(3) Productions whose RHS does not contain any nullable variable :

No production exist under this category.

The resultant grammar which is free from null move contains the set of productions from equation (1) to (8), i.e.,

$$A \to 01/0B1/11/1B1$$
$$B \to 0/0B/1/1B$$

Example 4.18 : Eliminate the $\wedge$-productions from the grammar whose production rules are given below :

$$S \to aSa$$
$$S \to bSb$$
$$S \to \wedge$$

Solution : Given :

$$S \to aSa$$
$$S \to bSb$$
$$S \to \wedge$$

1. Find nullable variable :

$S \to \wedge$ is a production so S is a nullable variable.

$\therefore$ set of nullable variable = $\{S\}$

(2) Productions whose RHS contain the nullable variable :

(i) $S \to aSa$

Replacing S on RHS by $\wedge$, we get

$S \to aa$ ...(1)

$S \to aSa$ (Original Production) ...(2)

(ii) $S \to bSb$

Replacing S on RHS by $\wedge$, we get,

$S \to bb$ ...(3)

$S \to bSb$ (Original Production) ...(4)

3. There is **no production** in the given grammar whose **RHS does not contain any nullable** variable.

$\therefore$ The resultant grammar which is free from null move contains the set of productions from equation (1) to (4) i.e.,

$$S \to aa/aSa/bb/bSb$$

Example 4.19 : For the CFG given below, remove the null productions.

$$S \rightarrow a \mid Ab \mid aBa$$
$$A \rightarrow b \mid \wedge$$
$$B \rightarrow b \mid A$$

Solution : Given :

$$S \rightarrow a \mid Ab \mid aBa$$
$$A \rightarrow b \mid \wedge$$
$$B \rightarrow b \mid A$$

1. Find nullable variables :

$A \rightarrow \wedge$ is a production. So A is a nullable variable.

As A is nullable variable and there is a production $B \rightarrow A$ so **B is also a nullable variable**

$\therefore$ set of nullable variable = $\{A, B\}$

(2) Production whose RHS contains the nullable variable

(i) $S \rightarrow Ab$

Replacing A by $\wedge$ on RHS, we get,

$$S \rightarrow b \quad \text{...(1)}$$
$$S \rightarrow Ab \quad \text{(Original Production)} \quad \text{...(2)}$$

(ii) $S \rightarrow aBa$

Replacing B by $\wedge$ on RHS, we get,

$$S \rightarrow aa \quad \text{...(3)}$$
$$S \rightarrow aBa \quad \text{(Original Production)} \quad \text{...(4)}$$

(iii) $B \rightarrow A$

This production is deleted from the final set of production rule as it can not produce any useful language of the grammar.

(3) Production whose RHS does not contain any nullable variable :

$$S \rightarrow a \quad \text{...(5)}$$
$$A \rightarrow b \quad \text{...(6)}$$
$$B \rightarrow b \quad \text{...(7)}$$

The resultant grammar which is free from null move contains the set of production from equation (1) to (7) i.e.,

$$S \rightarrow b \mid Ab \mid aa \mid aBa \mid a$$
$$A \rightarrow b$$
$$B \rightarrow b$$

Example 4.20 : ...

$B \rightarrow b$

Example 4.20 : Consider the CFG given below :

$$S \rightarrow XY$$
$$S \rightarrow Zb$$
$$Y \rightarrow BW$$
$$Z \rightarrow AB$$
$$W \rightarrow Z$$
$$A \rightarrow aA/bA/\wedge$$
$$B \rightarrow Ba/Bb/\wedge$$

Elimiate the null productions from above grammar.

Solution : Given :

$$S \rightarrow XY$$
$$S \rightarrow Zb$$
$$Y \rightarrow BW$$
$$Z \rightarrow AB$$
$$W \rightarrow Z$$
$$A \rightarrow aA/bA/\wedge$$
$$B \rightarrow Ba/Bb/\wedge$$

1. Find nullable variable :

(i) $A \rightarrow \wedge$ and $B \rightarrow \wedge$ are the productions so **A and B are nullable variables.**

(ii) As A and B are nullable variable and there is a production $Z \rightarrow AB$, so Z is also a nullable variable.

(iii) There is a production $W \rightarrow Z$ and Z is *a* nullable variable so W is also a nullable variable.

$\therefore$ Set of nullable variable = $\{A, B, Z, W\}$

(2) Productions whose RHS contain the nullable variable :

(i) $X \rightarrow Zb$, Z is replaced by $\wedge$ on RHS, we get

$X \rightarrow b$ ....(1)

$X \rightarrow Zb$ (Original Production) ....(2)

(ii) $Y \rightarrow BW$ as both B and W on RHS are nullable varible, we can not erase both the variables, Hence, we get **two** productions,

$Y \rightarrow B$ ....(3)

$Y \rightarrow W$ ....(4)

(iii) $Z \rightarrow AB$ again A and B both are nullable variables, we can not erase both the variables, therefore, it gives

$Z \rightarrow A$ ....(5)

$Z \rightarrow B$ ....(6)

(iv) $W \rightarrow Z$ this production is deleted from the set of productions as it can not produce any useful language of the grammar.

(v) $A \rightarrow aA$ replacing A on RHS by $\wedge$, gives

$A \rightarrow a$ ....(7)

$A \rightarrow aA$ (Original Production) ....(8)

(vi) $A \to bA$ replacing A on RHS by $\wedge$, gives

$A \to b$ ....(9)

$A \to bA$ (Original Production) ....(10)

(vii) $B \to Ba$ replacing b by $\wedge$ on RHS, we get

$B \to a$ ....(11)

$B \to Ba$ (Original Production) ....(12)

(viii) $B \to Bb$ replacing B on RHS by $\wedge$, gives,

$B \to b$ ....(13)

$B \to Bb$ (Original Production) ....(14)

(3) Productions whose RHS does not contain any nullable variable :

$S \to XY$ ....(15)

The resultant grammar which is free from null move contains the set of productions from equation (1) to (15). i.e.,

$$X \to b/Zb$$
$$Y \to B/W$$
$$Z \to A/B$$
$$A \to a/aA/b/bA$$
$$B \to a/Ba/b/Bb$$
$$S \to XY$$

Example 4.21 : Eliminate the $\wedge$-productions from the CFG given below :

$$S \to P0Q/QQ/0R/RQP$$
$$P \to R0/1R/RR/RQP$$
$$Q \to Q0/PQ/\wedge$$
$$R \to 0P/QQQ$$

Solution : Given

$$S \to P0Q/QQ/0R/RQP$$
$$P \to R0/1R/RR/RQP$$
$$Q \to Q0/PQ/\wedge$$
$$R \to 0P/QQQ$$

1. Find nullable variables :

(i) $Q \to \wedge$ is a production so Q is nullable variable.

(ii) $R \to QQQ$ is a production and Q is a nullable variable so R is also a nullable variable.

(iii) As R is nullable variable and there is a production $P \to RR$, therefore, P is also nullable variable

(iv) As Q is a nullable variable and there is a production $S \to QQ$, therefore, S is also nullable variable.

The set of nullable variable = $\{S, P, Q, R\}$

(2) Productions whose RHS contains the nullable variable :

(i) $S \to P0Q$ gives

$S \to 0Q$ ....(1)

$S \to P0$ ....(2)

$S \to 0$ ....(3)

$S \to P0Q$ (Original production) ....(4)

(ii) $S \to QQ$ gives

$S \to Q$ ....(5)

(Since, it can not erase both variable or RHS).

(iii) $S \to 0R$ gives

$S \to 0$ ....(6)

$S \to 0R$ (Original Production) ....(7)

(iv) $S \to RQP$ gives

$S \to RQ$ ....(8)

$S \to QP$ ....(9)

$S \to RP$ ....(10)

$S \to R$ ....(11)

$S \to Q$ ....(12)

$S \to P$ ....(13)

(v) $P \to R0$ gives

$P \to 0$ ....(14)

$P \to R0$ (Original Production) ....(15)

(vi) $P \to 1R$ gives

(vi) $P \rightarrow 1R$ gives

$P \rightarrow 1$ ....(16)

$P \rightarrow 1R$ (Original Production) ....(17)

(vii) $P \rightarrow RR$ gives

$P \rightarrow R$ ....(18)

(Since, it can not erase both variable on RHS).

(viii) $P \rightarrow RQP$ gives

$P \rightarrow RQ$ ....(19)

$P \rightarrow QP$ ...(20)

$P \rightarrow RP$ ....(21)

$P \rightarrow R$ ....(22)

$P \rightarrow Q$ ....(23)

(ix) $Q \rightarrow Q0$ gives

$Q \rightarrow 0$ ....(24)

$Q \rightarrow Q0$ (Original Production) ....(25)

(x) $Q \rightarrow PQ$ gives

$Q \rightarrow P$ ....(26)

(xi) $R \rightarrow 0P$ gives

$R \rightarrow 0$ ....(27)

$R \rightarrow 0P$ (Original Production) ....(28)

(xii) $R \rightarrow QQQ$ gives

$R \rightarrow QQ$ ....(29)

$R \rightarrow Q$ ....(30)

Production from (1) to (30) form the required grammar which is free from null productions.

4.5.4 Elimination of Unit Productions

Definition

A production in this context free grammar G of the form $A \to B$ where A and B are non-terminals/variables in grammer G.

Procedure to eliminate Unit Production :

1. **Construction of Unit Pairs :** If there is a production $A \to B$ and $B \to C$ in the given Context Free Grammar (CFG) than make the unit pair $A \to C$ and apply this **chain rule** until we get non-unit production $X \to \alpha$.
2. Replace these unit productions by a single production $A \to \alpha$. Repeat above steps for each non-terminal in the given CFG.
3. Add all non-unit productions in the resultant grammar.

Example 4.22 : Consider the following Context Free Grammar (CFG) G whose productions are given below :

$$S \to A\ B$$
$$A \to a$$
$$B \to C\ /\ b$$
$$C \to D$$
$$D \to E$$
$$E \to a$$

Eliminate unit productions and find the resultant grammar.

Solution : Given

$$S \to A\ B$$
$$A \to a$$

$$B \to C\ /\ b$$
$$C \to D$$
$$D \to E$$
$$E \to a$$

1. Construction of Unit pairs

(i) For non-terminals S and A there is no unit productions.

(ii) For unit production $B \to C$, we have

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

Replace these unit production by single production, we get,

$$B \to a \qquad \text{....(1)}$$

(iii) For unit production $C \to D$, we have

$$C \to D$$
$$D \to E$$
$$E \to a$$

Replace these unit productions by single production, we get,

$$C \to a \qquad \text{....(2)}$$

(iv) For unit production $D \to E$, we have

$$D \to E$$
$$E \to a$$

Replace these unit production by the single production, we get,

$$D \to a \qquad \text{....(3)}$$

(v) For E there is no unit production.

2. Non-unit productions in the given Context Free Grammar (CFG) :

$$S \to A\ B \qquad \text{....(4)}$$
$$A \to a \qquad \text{....(5)}$$
$$B \to b \qquad \text{....(6)}$$
$$E \to a \qquad \text{....(7)}$$

Therefore, the set of production rules in the resultant grammar which is free from unit oductions consists of set of rules as in equation from (1) to (7), i.e.,

$$S \to AB$$
$$A \to a$$
$$B \to b$$

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

or

$$S \to A\ B$$
$$A \to a$$
$$B \to a\ /\ b$$
$$C \to a$$
$$D \to a$$
$$E \to a$$

$E \rightarrow a$

Example 4.23 : Eliminate the unit productions from the Context-free grammar *G* given below :

$$S \rightarrow 0A \,/\, 1B \,/\, C$$
$$A \rightarrow 0S \,/\, 00$$
$$B \rightarrow 1 \,/\, A$$
$$C \rightarrow 01$$

Solution : Given :

$$S \rightarrow 0A \,/\, 1B \,/\, C$$
$$A \rightarrow 0S \,/\, 00$$
$$B \rightarrow 1 \,/\, A$$
$$C \rightarrow 01$$

1. Construction of unit pairs :

(i) For the **unit production $S \rightarrow C$,** we have

$$S \rightarrow C$$
$$C \rightarrow 01$$

Replacing this pair of unit production by single production, we get,

$$S \rightarrow 01 \qquad \text{....(1)}$$

(ii) For the **unit production $B \rightarrow A$,** we have

$$B \rightarrow A$$
$$A \rightarrow 0S \,/\, 00$$

Replacing above pairs by single production, we get,

$$B \rightarrow 0S \,/\, 00 \qquad \text{....(2)}$$

There is **no other unit production** in the grammar.

2. Non-unit productions in the CFG :

$$S \rightarrow 0A \,/\, 1B \qquad \text{....(3)}$$
$$A \rightarrow 0S \,/\, 00 \qquad \text{....(4)}$$
$$B \rightarrow 1 \qquad \text{....(5)}$$
$$C \rightarrow 01 \qquad \text{....(6)}$$

Combining the productions from (1) to (6), we get the required set of productions which is free from unit production.

$$S \rightarrow 0A \,/\, 1B \,/\, 01$$
$$A \rightarrow 0S \,/\, 00$$
$$B \rightarrow 1 \,/\, 0S \,/\, 00$$
$$C \rightarrow 01$$

Example 4.24 : Eliminate the unit productions from the context-free grammar given below :

$$S \rightarrow AB \mid A$$
$$A \rightarrow C \mid d$$
$$C \rightarrow b$$

Solution : Given :

$$S \rightarrow AB \mid A$$
$$A \rightarrow C \mid d$$
$$C \rightarrow b$$

1. Construction of unit pairs :

(i) For the **unit production** $S \rightarrow A$, we have

$$S \rightarrow A$$
$$A \rightarrow C$$
$$C \rightarrow b$$

Replacing these pairs of production by single production, we get,

$$S \rightarrow b \qquad \text{....(1)}$$

(ii) For the **unit production** $A \rightarrow C$, we have,

$$A \rightarrow C$$
$$C \rightarrow b$$

Replacing the above pair by single production, we get,

$$A \rightarrow b \qquad \text{....(2)}$$

There is no other unit production in the given *CFG*.

2. Non-unit production in the *CFG* :

$$S \rightarrow AB \qquad \text{....(3)}$$
$$A \rightarrow d \qquad \text{....(4)}$$
$$C \rightarrow b \qquad \text{....(5)}$$

Productions from (1) to (5) form the required set of productions free from unit productions. Which is shown below :

$$S \rightarrow b \mid AB$$
$$A \rightarrow b \mid d$$
$$C \rightarrow b$$

Example 4.25 : Convert the following context-free grammar into Chomsky normal form.

$$S \rightarrow aaaaS$$
$$S \rightarrow aaaa$$

Solution :

Given :

$$S \rightarrow aaaaS$$
$$S \rightarrow aaaa$$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : **Productions of the form $A \rightarrow a$ and $A \rightarrow BC$:** No production under this category exist.

Step 4 : Elimination of terminal symbol on RHS :

(a) Production $S \rightarrow aaaaS$ yields

$$S \rightarrow AAAAS \quad \text{....(1)}$$

(A new non-terminal A is added due to terminal a)

Scanned by CamScanner

and $$A \rightarrow a \quad \text{....(2)}$$

(b) Production $S \rightarrow aaaa$ yields

$$S \rightarrow AAAA \quad \text{....(3)}$$

Step 5 : Restriction on number of variables on RHS.

(a) Production (2) is in required form of CNF.

(b) For production (1), $S \rightarrow AAAAS$, we get,

$$S \rightarrow AR_1 \text{ (where, } R_1 = AAAS) \quad \text{....(4)}$$

$$R_1 \rightarrow AR_2 \text{ (where, } R_2 = AAS) \quad \text{....(5)}$$

$$R_2 \rightarrow AR_3 \text{ (where, } R_3 = AS) \quad \text{....(6)}$$

$$R_3 \rightarrow AS \quad \text{....(7)}$$

(c) For production (2), $S \rightarrow AAAA$, we can replace by

$$S \rightarrow R_4R_5 \text{ (where } R_4 = AA) \quad \text{....(8)}$$

$$R_4 \rightarrow AA \quad \text{....(9)}$$

$$R_5 \rightarrow AA \quad \text{....(10)}$$

Note : Production (9) and (10) can be replaced by single production, and one of the production can be removed.

The productions of equation (2), (4), (5), (6), (7), (8) and (10) constitute the required productions of chomsky normal form.

Example 4.26 : Convert the following context-free grammar into equivalent Chomsky normal form: 5

$$S \rightarrow aAbB$$
$$A \rightarrow aA/a$$
$$B \rightarrow bB/b$$

Solution :

Given :

$$S \rightarrow aAbB$$
$$A \rightarrow aA/a$$
$$B \rightarrow bB/b$$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : Productions of the form $A \rightarrow a$ and $A \rightarrow BC$:

$$A \rightarrow a \qquad \text{....(1)}$$
$$B \rightarrow b \qquad \text{....(2)}$$

Step 4 : Elimination of terminal symbol on RHS :

(a) Production $S \rightarrow aAbB$ yields

$$S \rightarrow R_1AR_2B \qquad \text{....(3)}$$

(b) Production $A \rightarrow aA$ yields

$$A \rightarrow R_1A \qquad \text{....(4)}$$

(c) Production $B \rightarrow bB$ yields

$$B \rightarrow R_2B \qquad \text{....(5)}$$

(d) Productions corresponding to the **new non-terminals R_1 and R_2 :**

$$R_1 \rightarrow a \qquad \text{....(6)}$$
$$R_2 \rightarrow b \qquad \text{....(7)}$$

Step 5 : (a) Productions obtained in equation (1), (2), (4), (5), (6) and (7) are in required form of CNF.

(b) For production obtained in equation (3), $\dot{S} \rightarrow R_1AR_2B$, we can add new production rules as follows :

$$S \rightarrow R_1R_3 \text{ (where, } R_3 = AR_2B) \qquad \text{....(8)}$$
$$R_3 \rightarrow AR_4 \text{ (where, } R_4 = R_2B) \qquad \text{....(9)}$$
$$R_4 \rightarrow R_2B \qquad \text{....(10)}$$

The productions obtained in equation (1), (2), (4), (5), (6), (7), (8), (9) and (10) are the required productions of chomsky normal form.

Example 4.27 : Convert the following context free grammar in Chomsky normal form

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow a$$
$$S \rightarrow b$$

Solution :

Given :

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow a$$
$$S \rightarrow b$$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : Productions of the form $A \rightarrow a$ and $A \rightarrow BC$:

$$S \rightarrow a \quad \text{...(1)}$$
$$S \rightarrow b \quad \text{...(2)}$$

Step 4 : Elimination of terminal symbols on RHS :

(a) Production $S \rightarrow aSa$ yield

$$S \rightarrow R_1SR_1 \quad \text{...(3)}$$

(b) Production $S \rightarrow bSb$ yield

$$S \rightarrow R_2SR_2 \quad \text{...(4)}$$

(c) Productions corrosponding to the **new non-terminals R_1 and R_2 :**

$$R_1 \rightarrow a \quad \text{...(5)}$$
$$R_2 \rightarrow b \quad \text{...(6)}$$

Step 5 :

(a) Productions in equation (1), (2), (5) and (6) are in required from of CNF.

(b) For production $S \rightarrow R_1SR_1$, new productions are added as follows :

$$S \rightarrow R_1R_4 \text{ (where } R_4 = SR_1\text{)} \quad \text{...(7)}$$
$$R_4 \rightarrow SR_1 \quad \text{...(8)}$$

(c) For production $S \rightarrow R_2SR_4$, new productions are added as follows :

$$S \rightarrow R_2R_5 \text{ (where } R_5 = SR_4\text{)} \quad \text{...(9)}$$
$$R_5 \rightarrow SR_4 \quad \text{...(10)}$$

The productions of equation (1), (2), (5), (6), (7), (8), (9) and (10) form the required productions of CNF.

productions of CNF.

Example 4.28 : Consider the grammar $G = (\{S, A\}, \{a, b\}, P, S)$ where P consists of

$$S \rightarrow aAS \mid a$$

$$A \rightarrow SbA \mid SS \mid ba$$

Convert it into equivalent Chomsky normal form.

Solution : Given :

$$S \rightarrow aAS \mid a$$

$$A \rightarrow SbA \mid SS \mid ba$$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : Productions of the form $A \rightarrow a$ and $A \rightarrow BC$:

Scanned by CamScanner

$$S \rightarrow a \quad \text{....(1)}$$

$$A \rightarrow SS \quad \text{....(2)}$$

Step 4 : Elimination of terminal symbol on RHS :

(a) Production $S \rightarrow aAS$ yields

$$S \rightarrow R_1AS \quad \text{....(3)}$$

(b) Production $A \rightarrow SbA$ yields

$$A \rightarrow SR_2A \quad \text{....(4)}$$

(c) Production $A \rightarrow ba$ yields

$$A \rightarrow R_2R_1 \quad \text{....(5)}$$

Productions corresponding to the new non-terminals R_1 and R_2 :

$$R_1 \rightarrow a \quad \text{....(6)}$$

$$R_2 \rightarrow b \quad \text{....(7)}$$

Step 5 :

(a) Production in equation (1), (2), (5), (6) and (7) are in required form of CNF.

(b) Production of equation (3) $S \rightarrow R_1AS$ in changed as follows :

$$S \rightarrow R_1R_3 \text{ (where } R_3 = AS) \quad \text{....(8)}$$

$$R_3 \rightarrow AS \quad \text{....(9)}$$

(c) Production of equation (4), $A \rightarrow SR_2A$ is changed as follows :

$$A \rightarrow SR_4 \text{ (where } R_4 = R_2A) \quad \text{....(10)}$$

$$R_4 \rightarrow R_2A \quad \text{....(11)}$$

(1), (2), (5), (6), (7), (8), (9), (10) and (11) are the required productions of the form CNF.

Example 4.29 : Convert the following CFG into equivalent CNF.

$$S \rightarrow aB$$
$$S \rightarrow bA$$
$$A \rightarrow a$$
$$A \rightarrow aS$$
$$A \rightarrow bAA$$
$$B \rightarrow b$$
$$B \rightarrow aS$$
$$B \rightarrow aBB$$

Solution : Given :

$$S \rightarrow aB$$

$$S \rightarrow bA$$
$$A \rightarrow a$$
$$A \rightarrow aS$$
$$A \rightarrow bAA$$
$$B \rightarrow b$$
$$B \rightarrow aS$$
$$B \rightarrow aBB$$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : Productions of the from $A \rightarrow a$ and $A \rightarrow BC$:

$$A \rightarrow a \quad \text{....(1)}$$
$$B \rightarrow b \quad \text{....(2)}$$

Step 4 : Elimination of terminal symbol on RHS :

(a) Production $S \rightarrow aB$ yields

$$S \rightarrow R_1B \quad \text{....(3)}$$

(b) Production $S \rightarrow bA$ yields

$$S \rightarrow R_2A \quad \text{....(4)}$$

(c) Production $A \rightarrow aS$ yields

$$A \rightarrow R_1S \quad \text{....(5)}$$

(d) Production $A \rightarrow bAA$ yields

$$A \rightarrow R_2AA \quad \text{....(6)}$$

(e) Production $B \rightarrow aS$ yields

$$B \rightarrow R_1S \quad \text{....(7)}$$

(f) Production $B \rightarrow aBB$ yields

$$B \rightarrow R_1BB \quad \text{....(8)}$$

Productions corrosponding to the new non-terminals R_1 and R_2 :

$$R_1 \rightarrow a \quad \text{....(9)}$$
$$R_2 \rightarrow b \quad \text{....(10)}$$

Step 5 :

(a) Production in equation (1), (2), (3), (4), (5), (7), (9) and (10) are in required form of CNF.

(b) For production in equation (6), $A \rightarrow A_2AA$, new productions are added ab follows

$$A \rightarrow R_2R_3 \text{ (where, } R_3 = AA) \quad \text{....(11)}$$
$$R_3 \rightarrow AA \quad \text{....(12)}$$

(c) For productions in equation (8), $B \rightarrow R_1BB$, new productions are added as follows :

$$B \rightarrow R_1R_4 \text{ (where } R_4 = BB) \quad \text{....(13)}$$
$$R_4 \rightarrow BB \quad \text{....(14)}$$

(1), (2), (3), (4), (5), (7), (9), (10), (11), (12), (13) and (14) are the required productions of the form CNF.

Example 4.30 : Convert the following grammar into equivalent chomsky normal form:

$$S \rightarrow \neg S / [S \supset S] / p / q$$

Solution : **Given :** $S \rightarrow \neg S / [S \supset S] / p / q$

Step 1 : The context-free grammar does not contain any null production, so, we can skip this step.

Step 2 : **Elimination of unit production :** The given context-free grammar also does not contain any unit production so, we can skip this step.

Step 3 : Productions of the form $A \rightarrow a$ and $A \rightarrow BC$:

$$S \rightarrow p \qquad ....(1)$$

$$S \rightarrow q \qquad ....(2)$$

Step 4 : Elimination of terminal symbols on RHS :

(a) Production $S \rightarrow \neg S$ yields

$$S \rightarrow R_1 S \qquad ....(3)$$

(b) Production $S \rightarrow [S \supset S]$ yields

$$S \rightarrow R_2 S R_3 S R_4 \qquad ....(4)$$

Productions corrosponding to the new non-terminals R_1, R_2, R_3 and R_4 :

$$R_1 \rightarrow \neg \qquad ....(5)$$

$$R_2 \rightarrow [\qquad ....(6)$$

$$R_3 \rightarrow \supset \qquad ....(7)$$

$$R_4 \rightarrow] \qquad ....(8)$$

Step 5 : (a) Productions in equation (1), (2), (3), (5), (6), (7) and (8) are in required form of CNF.

(b) For production obtained in equation (4), $S \rightarrow R_2 S R_3 S R_4$, new productions are added as follows :

$$S \rightarrow R_2 R_5 \text{ (where } R_5 = S R_3 S R_4) \qquad ....(9)$$

$$R_5 \rightarrow S R_6 \text{ (where } R_6 = R_3 S R_4) \qquad ....(10)$$

$$R_6 \rightarrow R_3 R_7 \text{ (where } R_7 = S R_4) \qquad ....(11)$$

$$R_7 \rightarrow S R_4 \qquad ....(12)$$

(1), (2), (3), (5), (6), (7), (8), (9), (10), (11) and (12) are the required productions of CNF.

Example 4.31 : Consider the Context Free Grammer (CFG) G whose productions are given below :

$$A \rightarrow aBD / bDB / c$$

$$A \rightarrow AB / AD$$

Remove left recursion from the grammar.

Solution : Given

$$A \rightarrow aBD / bDB / c$$

$$A \rightarrow AB \mid AD$$

Here, $\alpha_1 = B,\ \alpha_2 = D$

$\beta_1 = a\ B\ D,\ \beta_2 = b\ D\ B,\ \beta_3 = c$

From above definition

$\therefore$ New A–productions are :

$$A \rightarrow a B D / bDB / c$$

$$A \rightarrow aBDZ / bDBZ / cZ \text{ and}$$

New Z–productions are :

$$Z \rightarrow B / D$$

$$Z \rightarrow BZ / DZ$$

Example 4.32 : Consider the following CFG G whose productions are given below:

$$S \to AB$$
$$A \to BS/b$$
$$B \to SA/a$$

Convert it into equivalent GNF.

Solution : Given

$$S \to AB$$
$$A \to BS/b$$
$$B \to SA/a$$

1. The given CFG is in the CNF.
2. Rename Variables S, A and B as A_1, A_2 and A_3 respectively, we get

$$A_1 \to A_2A_3$$
$$A_2 \to A_3A_1/b$$
$$A_3 \to A_1A_2/a$$

(Here, number of variables, $n = 3$)

3. Seperation of Productions :

(i) Productions of the form $A_i \to a\gamma$ or $A_i \to A_j\gamma$ where $j > i$ are in required form

$$A_1 \to A_2A_3$$
$$A_2 \to A_3A_1 \mid b$$
$$A_3 \to a$$

are in required form.

(ii) Productions of the form $A_i \to A_j\gamma$ where $j \le i$, $A_3 \to A_1A_2$

4. Apply left factoring on this production.

We get,

$$A_3 \to A_2A_3A_2 \qquad (\because A_1 \to A_2A_3)$$

Apply once again left factoring on this production since $j \le i$, we get,

$$A_3 \to A_3A_1A_2 / bA_3A_2 \qquad (\because A_2 \to A_3A_1 | b)$$

A_3-productions are :

$$A_3 \to a$$
$$A_3 \to bA_3A_2$$
$$A_3 \to A_3A_1A_3A_2$$

5. Apply left recursion to A_3-productions by introducing new variable Z_3, we get :

(i) A-productions :

$$A_3 \to a$$
$$A_3 \to bA_3A_2$$
$$A_3 \to aZ_3$$
$$A_3 \to bA_3A_2Z_3$$

(ii) Z-productions :

$$Z_3 \to A_1A_3A_2$$
$$Z_3 \to A_1A_3A_2Z_3$$

6. A_n-production :

(i) A_3-productions are :

$$A_3 \to bA_3A_2 / bA_3A_2Z_3 / a / aZ_3 \qquad ...(1)$$

(ii) A_2-productions are :

$$A_2 \to A_3A_1 / b$$

Among these productions, we retain $A_2 \to b$ and eliminate $A_2 \to A_3A_1$ by using left factoring, we get,

$$A_2 \to bA_3A_2A_1 / bA_3A_2Z_3A_1 / aA_1 / aZ_3A_1 \qquad ...(2)$$

(iii) A_1-productions are :

$$A_1 \to A_2A_3$$

Apply left factoring on above production, we get,

$$A_1 \to bA_3A_2A_1A_3 / bA_3A_2Z_3A_1A_3 / aA_1A_3 / aZ_3A_1A_3 \quad ...(3)$$

7. Z-productions :

Modify the Z-production using left factoring, we get,

$$Z_3 \to bA_3A_2A_1A_3A_3A_2/bA_3A_2Z_3A_1A_3A_3A_2/aA_1A_3A_3A_2/ aZ_3A_1A_3A_3A_2$$

and $$Z_3 \to bA_3A_2A_1A_3A_3A_2Z_3/bA_3A_2Z_3A_1A_3A_3A_2Z_3/aA_1A_3A_3A_2Z_3/aZ_3A_1A_3A_3A_2Z_3 \quad ...(4)$$

The prodcutions obtained in equation (1) to (4) constitute the productions of resultant grammar.

Example. 4.33 : Construct the grammar in Greibach normal form equivalent to the context free grammer given below

$$S \to AA \;/\; a,$$
$$A \to SS \;/\; b$$

Solution : Given :

$$S \to AA \;/\; a,$$
$$A \to SS \;/\; b$$

Step 1. The given grammar is in CNF so, skip this step.

Step 2. Rename variables S and A as A_1, and A_2, respectively, we get

$$A_1 \to A_2A_2 \;/\; a,$$
$$A_2 \to A_1A_1 \;/\; b$$

Here, the number of varaibles $(n) = 2$

Step 3. Now, seperate the productions :

(i) Productions of the form $A_i \to a\gamma$ or $A_i \to A_j\gamma$ where $j > i$

$$A_1 \to a$$
$$A_1 \to A_2A_2$$
$$A_2 \to b$$

(ii) Productions of the form $A_i \to A_j\gamma$ where $j \leq i$

$$A_2 \to A_1A_1$$

Step 4. Apply left factoring on A_n-production.

(i)

$A_2 \to A_1A_1$ is replaced by

$$A_2 \to A_2A_2A_1 \qquad (\because\; A_1 \to A_2A_2)$$
$$A_2 \to aA_1 \qquad (\because\; A_1 \to a)$$

$\therefore$ A_n-productions are

$$A_2 \to A_2A_2A_1$$
$$A_2 \to aA_1$$

and

$$A_2 \to b$$

Step 5. Apply left recursion on A_n-productions : We introduce new variable Z_2 corrosponding to the variable A_2, we get,

(i) A_2-productions :

$$A_2 \to aA_1$$
$$A_2 \to b$$
$$A_2 \to aA_1Z_2$$

and $A_2 \to bZ_2$

(ii) Z_2-productions :

$$Z_2 \to A_2A_1$$
$$Z_2 \to A_2A_1Z_2$$

Step 6.(a) Modified A_n-productions :

$$A_2 \to aA_1$$
$$A_2 \to b$$
$$A_2 \to aA_1Z_2$$

and $A_2 \to bZ_2$

(b) A_{n-1}-productions :

$$A_1 \to a$$

and $A_1 \to A_2A_2$

Among A_1-productions, we retain $A_1 \to a$ which is in required form and eliminate $A_1 \to A_2A_2$ by using left factoring. The resultant A_1-productions are

$$A_1 \to aA_1A_2$$
$$A_1 \to bA_2$$
$$A_1 \to aA_1Z_2A_2$$
$$A_1 \to bZ_2A_2$$

Step 7. Modify Z-productions :

$$Z_2 \to aA_1A_1$$
$$Z_2 \to bA_1$$
$$Z_2 \to aA_1Z_2A_1$$
$$Z_2 \to bZ_2A_1$$
$$Z_2 \to aA_1A_1Z_2$$
$$Z_2 \to bA_1Z_2$$
$$Z_2 \to aA_1A_1Z_2$$
$$Z_2 \to bZ_2A_1Z_2$$

Hence, the equivalent grammar GNF is :

$$A_1 \to a \;/\; aA_1A_2 \;/\; bA_2 \;/\; aA_1Z_2A_2 \;/\; bZ_2A_2$$
$$A_2 \to aA_1 \;/\; b \;/\; aA_1Z_2 \;/\; bZ_2$$
$$Z_2 \to aA_1A_1 \;/\; bA_1 \;/\; aA_1Z_2A_1 \;/\; bZ_2A_1$$
$$Z_2 \to aA_1A_1Z_2 \;/\; bA_1Z_2 \;/\; aA_1Z_2A_1Z_2 \;/\; bZ_2A_1Z_2$$

Example 4.34 : Convert the following context-free grammar into equivalent GNF.

$$E \rightarrow E + T \,/\, T$$
$$T \rightarrow T * F \,/\, F$$
$$F \rightarrow (E) \,/\, a$$

Solution : Given :

$$E \rightarrow E + T \,/\, T$$
$$T \rightarrow T * F \,/\, F$$
$$F \rightarrow (E) \,/\, a$$

(1) Convert the grammar in CNF :

(A) First eliminate the unit productions :

(i) Construction of unit pairs

⇒ For production $E \rightarrow T$, we have

$$E \rightarrow T$$
$$T \rightarrow F$$
$$F \rightarrow a$$

Replacing this pair of unit productions by single production, we get,

$$E \rightarrow a \quad \text{...(1)}$$

⇒ For production $T \rightarrow F$, we have

$$T \rightarrow F$$
$$F \rightarrow a \quad \text{...(2)}$$

Replacing this pair of unit production by single production, we get, ...(3)

$$T \rightarrow a$$

There is no other unit pair in the grammar. ...(4)

(ii)Non-unit productions in CFG.

$$E \rightarrow E + T \quad \text{...(5)}$$
$$T \rightarrow T * F \quad \text{...(6)}$$
$$F \rightarrow (E) \,/\, a \quad \text{...(7)}$$

The equivalent grammar without unit production is :

$$E \rightarrow E + T \,/\, T * F \,/\, (E) \,/\, a$$
$$T \rightarrow T * F \,/\, (E) \,/\, a$$
$$F \rightarrow (E) \,/\, a$$

Note : $T \rightarrow F$ is replaced by $T \rightarrow (E)$ and $T \rightarrow a$. Similarly, $E \rightarrow T$ is replaced $E \rightarrow T * F$, $E \rightarrow (E)$ and $E \rightarrow a$.

(B) The grammar does not contain any null production.

(C) Productions of the form $A \rightarrow a$ and $A \rightarrow BC$

$$E \rightarrow a$$
$$T \rightarrow a$$
$$F \rightarrow a$$

(D) Elimination of terminal symbols in RHS

(i) Production $E \rightarrow E + T$ yields

$$E \rightarrow EAT$$

(ii) Production $E \rightarrow T * F$ yields

$$E \rightarrow TBF$$

(iii) Production $E \rightarrow (E)$ yields

$$E \rightarrow (EC$$

(iv) Productions corresponding to the new non-terminals A, B and C, we get,

$$A \rightarrow +$$
$$B \rightarrow *$$
$$C \rightarrow)$$

The modified productions are

$$E \rightarrow EAT \,/\, TBF \,/\, (EC \,/\, a$$
$$T \rightarrow TBF \,/\, (EC \,/\, a$$
$$A \rightarrow (EC \,/\, a$$
$$A \rightarrow +,\ B \rightarrow *,\ C \rightarrow)$$

(2) Rename variable A, B, C, F, T, E as A_1, A_2, A_3, A_4, A_5 and A_6 respectively, we get,

$$A_1 \rightarrow +$$
$$A_2 \rightarrow *$$
$$A_3 \rightarrow)$$
$$A_4 \rightarrow (A_6A_3 \,/\, a$$
$$A_5 \rightarrow A_5A_2A_4 \,/\, A_6A_3 \,/\, a$$
$$A_6 \rightarrow A_6A_1A_5 \,/\, A_5A_2A_4 \,/\, (A_6A_3 \,/\, a$$

(3) Seperate the productions :

(i) Productions of the form $A_i \rightarrow a\gamma$ or $A_i \rightarrow A_j\gamma$ where, $j > i$,

$$A_1 \rightarrow +$$
$$A_2 \rightarrow *$$
$$A_3 \rightarrow)$$
$$A_4 \rightarrow (A_6A_3 \,/\, a$$
$$A_5 \rightarrow (A_6A_3 \,/\, a$$

$$A_5 \rightarrow (A_6A_3 / a \qquad \ldots(1)$$

(4) Apply left factoring on A_n-Production

(**Note** : Here, number of variables, $n = 6$)

$$A_6 \rightarrow A_6A_1A_5$$

(a)Apply left-recursion on A_5-Productions, $A_5 \rightarrow A_5A_2A_4$, we get,

$$A_5 \rightarrow (A_6A_3 / a$$

$$A_5 \rightarrow (A_6A_3Z_5 / aZ_5$$

and

$$Z_5 \rightarrow A_2A_4 / A_2A_4Z_5 \qquad \ldots(2)$$

(b) Apply left-factoring on A_6-productions, $A_6 \rightarrow A_5A_2A_4$, we get

$$A_6 \rightarrow (A_6A_3A_2A_4 / aA_2A_4 / (A_6A_3Z_5A_2A_4 / aZ_5A_2A_4$$

Now, A_n-productions are

$$A_6 \rightarrow (A_6A_3A_2A_4 / aA_2A_4 / (A_6A_3Z_5A_2A_4 / aZ_5A_2A_4$$

$$A_6 \rightarrow (A_6A_3 / a \qquad \ldots(3)$$

(5) Apply left recursion on A_n-Productions, $A_6 \rightarrow A_6A_1A_5$, we get,

$$A_6 \rightarrow (A_6A_3A_2A_4 / aA_2A_4 / A_6A_3Z_5A_2A_4$$

$$A_6 \rightarrow aZ_5A_2A_4 / (A_6A_3 / a$$

$$A_6 \rightarrow (A_6A_3A_2A_4A_6 / aA_2A_4Z_6 / (A_6A_3Z_5A_2A_4Z_6$$

$$A_6 \rightarrow aZ_6A_2A_4A_6 / (A_6A_3Z_6 / aZ_6 \qquad \ldots(4)$$

Z-productions are :'

$$Z_6 \rightarrow A_1A_5 / A_1A_5Z_6 \qquad \ldots(5)$$

(6) The productions A_{n-1}, A_{n-2}, ... A_1, i.e., A_5, A_4, A_3, A_2 and A_1 are in required form.

(7) Modified Z-Productions :

(i) Z_5-Productions :

$$Z_5 \rightarrow {}^*A_4 / {}^*A_4Z_5 \qquad \ldots(6)$$

(ii) Z_6-Productions :

$$Z_6 \rightarrow +A_5 / +A_5Z_6 \qquad \ldots(7)$$

Productions of equation (1) to (7) constitute the resultant grammar.

Example. 4.35 : Find the gra...
...the resultant grammar.

Example. 4.35 : Find the grammar in GNF equivalent to the context-free grammar G whose productions are given below :

$$A_1 \rightarrow A_2A_3$$

$$A_2 \rightarrow A_3A_1 / b$$

$$A_3 \rightarrow A_1A_2 / a$$

Solution : Given :

$$A_1 \rightarrow A_2A_3$$

$$A_2 \rightarrow A_3A_1 / b$$

$$A_3 \rightarrow A_1A_2 / a$$

Step 1. The given is already in CNF, we can skip this step.

Step 2. **Rename variables** : There is no need to rename the variables as these are already in sequence.

Step 3. Seperate the productions

(i) Productions of the form $A_i \rightarrow a\gamma$ or $A_i \rightarrow A_j\gamma$ where $j > i$

$$A_1 \rightarrow A_2A_3$$

$$A_2 \rightarrow A_3A_1 / b$$

$$A_3 \rightarrow a$$

(ii) Productions of the form $A_i \rightarrow A_j\gamma$ where $j \leq i$

$$A_3 \rightarrow A_1A_2$$

Step 4. Apply left factoring an production

$$A_3 \rightarrow A_1A_2 \text{, we get}$$

$$A_3 \rightarrow A_2A_3A_2 \qquad (A_1 \rightarrow A_2A_3)$$

Again apply left factoring an above production

$$A_3 \rightarrow A_3A_1A_3A_2 \qquad (A_2 \rightarrow A_3A_1)$$

$$A_3 \rightarrow bA_3A_2 \qquad (A_2 \rightarrow b)$$

Now, A_3-productions are

$$A_3 \rightarrow A_3A_1A_3A_2$$

$$A_3 \rightarrow bA_3A_2$$

$$A_3 \rightarrow a$$

Step 5. Apply left recursion on A_n-productions :

We introduce new variable Z_3 corresponding to the variable A_3, we get

(i) A_3-productions are :

$$A_3 \rightarrow bA_3A_2$$

$$A_3 \rightarrow a$$

$$A_3 \rightarrow bA_3A_2Z_3$$

$$A_3 \rightarrow aZ_3$$

(ii) Z_3-productions are :

$$Z_3 \rightarrow A_1A_3A_2$$

$$Z_3 \rightarrow A_1A_3A_2Z_3$$

Step 6. A_n-production :

(i) A_3-productions are :

$$A_3 \rightarrow bA_3A_2 / bA_3A_2Z_3 / a / aZ_3 \qquad \ldots(1)$$

(ii) A_2-productions are :

$$A_2 \rightarrow A_3A_2 / b$$

Among these productions, we retain $A_2 \to b$ and eliminate $A_2 \to A_3A_1$ by using left factoring, we get,

$$A_2 \to bA_3A_2A_1 \,/\, bA_3A_2Z_3A_1 \,/\, aA_1 \,/\, aZ_3A_1 \qquad ...(2)$$

(iii) A_1-productions are :

$$A_1 \to A_2A_3$$

Apply left factoring on above production, we get,

$$A_1 \to bA_3A_2A_1A_3 \,/\, bA_3A_2Z_3A_1A_3 \,/\, aA_1A_3 \,/\, aZ_3A_1A_3 \qquad ...(3)$$

Step 7. Z-productions :

Modify the Z-production using left factoring, we get,

$$Z_3 \to bA_3A_2A_1A_3A_3A_2 / bA_3A_2Z_3A_1A_3A_3A_2 / aA_1A_3A_3A_2 / aZ_3A_1A_3A_3A_2$$

and $$Z_3 \to bA_3A_2A_1A_3A_3A_2Z_3 / bA_3A_2Z_3A_1A_3A_3A_2Z_3 / aA_1A_3A_3A_2Z_3 / aZ_3A_1A_3A_3A_2Z_3 \quad ...(4)$$

The productions obtained in equation (1) to (4) constitute the productions of resultant grammar.

Example 4.36 : Show that the language $L = \{a^nb^nc^n : n \geq 0\}$ is not context free.

Solution

Given : $$L = \{a^nb^nc^n : n \geq 0\}$$

We use pumping lemma to show that language L is not context free language.

Assume for contraction that the $L = \{a^nb^nc^n : n \geq 0\}$ is context free.

Let us consider a number m such that $W \in L$ and $|W| \geq m$.

We pick,

$$W = a^mb^mc^m$$

Now, we write

$$W = uvxyz \text{ with lengths } |vxy| \leq m \text{ and } |vy| \geq 1$$

From, the pumping lemma,

$$uv^ixy^iz \in L \qquad \forall i \geq 0$$

Now, we examine all possible cases for string vxy in W.

Case 1 : vxy is within a^m.

$$\overset{u}{aa}\,\overset{vxy}{a \,....\, a}\,\overset{z}{abbb \,....\, bbb\, ccc \,....\, ccc}$$
$$\underbrace{aa\,a \,....\, a\,a}_{m}\underbrace{bbb \,....\, bbb}_{m}\underbrace{ccc \,....\, ccc}_{m}$$

v and y consists only from a.

Repeating v and y, $k \geq 1$.

$$\overset{u}{aa}\,\overset{v^2xy^2}{aaaa \,....\, aaaa}\,\overset{z}{abbb \,....\, bbb\, ccc \,....\, ccc}$$
$$\underbrace{aa\,aaaa \,....\, aaaa\,a}_{m+k}\underbrace{bbb \,....\, bbb}_{m}\underbrace{ccc \,....\, ccc}_{m}$$

From pumping lemma,

$$uv^2xy^2z \in L, \qquad k \geq 1$$

But, $$uv^2xy^2z = a^{m+k}b^mc^m \notin L$$

$$\Downarrow$$

Contradiction

Case 2 : vxy is within b^m.

$$\overset{u}{aaa \,...\, aaabb}\,\overset{vxy}{b \,....\, bb}\,\overset{z}{bccc \,....\, ccc}$$
$$\underbrace{aaa \,...\, aaa}_{m}\underbrace{bbb \,....\, bbb}_{m}\underbrace{ccc \,....\, ccc}_{m}$$

Similarly, analysing as in with case 1.

Scanned by CamScar

Case 3 : vxy is within c^m.

$$\overset{u}{aaa \,....\, aaabbb \,....\, bbbcc}\,\overset{vxy}{c \,....\, c}\,\overset{z}{cc}$$
$$\underbrace{aaa \,....\, aaa}_{m}\underbrace{bbb \,....\, bbb}_{m}\underbrace{ccc \,....\, ccc}_{m}$$

Similar analysis with case 1

Case 4 : vxy overlaps among a^m and b^m.

$$\overset{u}{aaa \,....\, aa}\,\overset{vxy}{abbb}\,\overset{z}{\,....\, bbbccc \,....\, ccc}$$
$$\underbrace{aaa \,....\, aaa}_{m}\underbrace{bbb \,....\, bbb}_{m}\underbrace{ccc \,....\, ccc}_{m}$$

Possibility 1 : v contains only a

y contains only b

$$k_1 + k_2 \geq 1$$

$$\overbrace{aaa}^{u}\overbrace{\underbrace{\ldots\ aaaaaaa}_{m+k_1}\underbrace{\ldots\ bbbbbbb\ \ldots\ bbb}_{m+k_2}}^{vx^2y^2}\overbrace{\underbrace{ccc\ \ldots\ ccc}_{m}}^{z}$$

From pumping lemma :

$$uv^2xy^2z \in L, \qquad k_1 + k_2 \geq 1$$

But,

$$uv^2xy^2z = a^{m+k_1}b^{m+k_2}c^m \not\subset L$$

$$\Downarrow$$

CONTRAD'CTION

Possibility 2 : v contains a and b

y contains only b.

$$\overbrace{\underbrace{aaa\ \ldots}_{m}}^{u}\overbrace{aaabbb}^{vxy}\overbrace{\underbrace{\ldots\ bbb}_{m}\underbrace{ccc\ \ldots\ ccc}_{m}}^{z}$$

$$k_1 + k_2 + k \geq 1$$

$$\overbrace{\underbrace{aaa\ \ldots}_{m}}^{u}\overbrace{\underbrace{aaaabbaabbbbbbbbb}_{k_1 k_1}}^{v^2xy^2}\overbrace{\underbrace{\ldots\ bbb}_{m}\underbrace{ccc\ \ldots\ ccc}_{m}}^{z}$$

From pumping lemma

$$uv^2xy^2z \in L; \qquad k_1 + k_2 + k \geq 1$$

But,

$$uv^2xy^2z = a^m b^{k_1} a^{k_2} b^{m+k} c^m \notin L$$

$$\Downarrow$$

CONTRADICTION

Scanned by

Possibility 3 : v contains only a

y contains a and b.

$$\overbrace{\underbrace{aaa\ \ldots}_{m}}^{u}\overbrace{aaabbb}^{vxy}\overbrace{\underbrace{\ldots\ bbb}_{m}\underbrace{ccc\ \ldots\ ccc}_{m}}^{z}$$

Similar analysis with possibility 2.

Case 5 : vxy overlaps b^m and c^m.

$$\overbrace{\underbrace{aaa\ \ldots\ aaa}_{m}\underbrace{bbb\ \ldots}_{m}}^{u}\overbrace{bbbccc\ \ldots}^{vxy}\overbrace{ccc}^{z}$$

Similar analysis with case 4.

All the cases have been considered and there are no other cases to consider.

$\because$ $|vxy| \leq m$, string vxy can not overlap a^m, b^m and c^m at the same time.

In all case, we get the CONTRADICTION.

$\because$ The assumption that $L = \{a^n b^n c^n : n \geq 0\}$ is context free is wrong.

Conclusion : L is not context free.

Example 4.37 : Prove that the language $L = \{vv : v \in \{a, b\}^*\}$ is not context free.

Solution : Given : $L = \{vv : v \in \{a, b\}^*\}$

Assume for contradiction that language

$$L = \{vv : v \in \{a, b\}^*\} \text{ is context free.}$$

$\because$ L is context free and of infinite length, we can apply punping lemma.

Let us consider a number m such that

$$W \in L \text{ and } |W| \geq m$$

We pick, $\quad a^m b^m a^m b^m \in L$

We can write,

$$a^m b^m a^m b^m = uvxyz \text{ with lengths } |vxy| \leq m \text{ and } |vy| \geq 1.$$

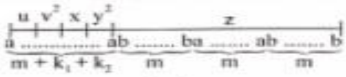
From punping lemma,

$$uv^i xy^i z \in L \quad \forall i \geq 0$$

Now, we consider all possible cases : for string vxy in $a^m b^m a^m b^m$.

Case 1 : vxy is within the first a^m :

$$v = a^{k_1}, \quad y = a^{k_2}, \quad k_1 + k_2 \geq 1$$



$$a^{m+k_1+k_2} b^m a^m b^m = uv^2 xy^2 z \in L; \quad k_1 + k_2 \geq 1$$

But, from pumping lemma

$$uv^2 xy^2 \in L$$

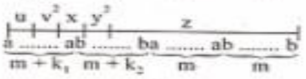
$$\Downarrow$$

CONTRADICTION

Case 2 : v is in the first a^m

y is in the first b^m

$$v = a^{k_1}, \quad y = b^{k_2}, \quad k_1 + k_2 \geq 1$$



$$a^{m+k_1} b^{m+k_2} a^m b^m = uv^2 xy^2 z \in L; \quad k_1 + k_2 \geq 1$$

But from Pumping lemma.

$$uv^2 xy^2 z \in L$$

$$\Downarrow$$

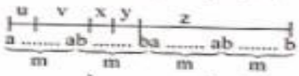
CONTRADICTION

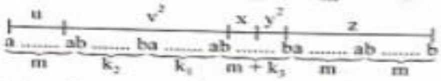
$\Downarrow$

CONTRADICTION

Case 3 :

$$v = a^{k_1}b^{k_2}, \qquad y = b^{k_3}, \qquad k_1, k_2 \geq 1$$





$$a^m b^{k_2} a^{k_1} b^{m+k_3} a^m b^m = uv^2xy^2z \in L; \qquad k_1, k_2 \geq 1$$

Scanned by CamScann

$$a^m b^{k_2} a^{k_1} b^{k_3} a^m b^m = uv^2xy^2z \notin L$$

But, from pumping lemma.

$$uv^2xy^2z \in L$$

$\Downarrow$

CONTRADICTION

Case 4 : v is in first a^m

y overlaps the first $a^m b^m$.

Analysis is similar to case 3.

Other cases : (i) vxy is within $a^m b^m a^m b^m$ means vxy is in first b^m, or second a^m or in second b^m.

Analysis is similar to case 1

i.e. $\underline{a}^m b^m a^m b^m$

(ii) vxy overlaps $a^m \underline{b^m a^m} b^m$ or $a^m b^m \underline{a^m b^m}$

Analysis is similar to case 2, 3 and 4.

There are no other case to consider

$\because$ $|vxy| \leq m$, it is not possible from vxy to overlap : $\underline{a^m b^m} a^m b^m$ nor $a^m \underline{b^m a^m b^m}$ nor $\underline{a^m b^m a^m b^m}$.

In all cases, we get contradiction,

$\therefore$ Our assumption that the language $L = \{vv : v \in \{a, b\}^*\}$ is context free is **not** true.

Conclusion : L is not context free language

Example 4.38 : Prove that the language $L = \{a^{n!} : n \geq 0\}$ is not context free.

Solution : Given : $L = \{a^{n!} : n \geq 0\}$

Assume for contradiction that the language $L = \{a^{n!} : n \geq 0\}$ is context free.

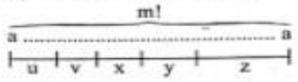
$\because$ The language L is context free and infinite, we can apply the pumping lemma.
Let us consider a number **m** such that $W \in L$ and $|W| \geq m$.

We pick, $\quad a^{m!} \in L$

We can write, $\quad a^{m!} = uvxyz$ with lengths $|vxy| \leq m$ and $|vy| \geq 1$

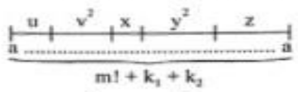
From pumping lemma, $\quad uv^ixy^iz \in L \quad \forall i \geq 0$

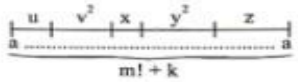
Now, we examine all possible cases of string vxy in $a^{m!}$



$\therefore$

$$v = a^{k_1}, \qquad y = a^{k_2} \ ; \qquad 1 \leq k_1 + k_2 \leq m$$





where,
$$k = k_1 + k_2; \qquad 1 \leq k \leq m$$
$$a^{m!+k} = uv^2xy^2z \qquad (1 \leq k \leq m)$$

$\because$
$$1 \leq k \leq m, \qquad \forall m \geq 2$$

We have,

$$\begin{aligned} m! + k &\leq m! + m \\ &< m! + m!m \\ &= m!(1 + m) \\ &= (m+1)! \end{aligned}$$
$$\Downarrow$$
$$m! < m! + k < (m+1)!$$
$$\Downarrow$$
$$a^{m!+k} = uv^2xy^2z \in L$$

However, from pumping lemma

$$vu^2xy^2z \in L$$
$$a^{m!+k} = uv^2xy^2z \in L$$
$$\Downarrow$$
$$\Downarrow$$

CONTRADICTION

$\therefore$ Our assumption that the language $L = \{a^{n!} : n \geq 0\}$ is context free is not true.

Conclusion : L is not context free.

Example 4.39 : Prove that the language $L = \{a^{n^2} b^n : n \geq 0\}$ is not context free.

Solution : Given : $L = \{a^{n^2} b^n : n \geq 0\}$

Assume for contradiction that the the language $L = \{a^{n^2} b^n : n \geq 0\}$ is context free.

$\because$ L is context free and infinite, we can apply the pumping lemma.

Let us consider the number **m** such that $W \in L$ and $|W| \geq m$.

We pick,

$$a^{m^2} b^m \in L$$

We can write, $a^{m^2} b^m$ = uvxyz with lengths $|vxy| \leq m$ and $|vy| \geq 1$

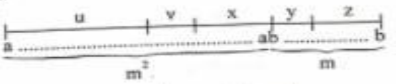
From pumping lemma

$$uv^i xy^i z \in L, \quad \forall i \geq 0$$

Now, we can examine all possible cases of string vxy in $a^{m^2} b^m$.

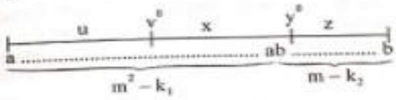
Case 1 : v is in a^m

y is in b^m



where
$$v = a^{k_1}, \; y = b^{k_2} \text{ and}$$
$$1 \leq k_1 + k_2 \leq m$$

(i) $k \neq 0$ and $k_2 \neq 0$



$$a^{m^2-k_1} b^{m-k_2} = uv^0 xy^0 z$$

$$k_1 \neq 0 \text{ and } k_2 \neq 0 \; ; \quad 1 \leq k_1 + k_2 \leq m$$

$$\Downarrow$$

$$(m - k_2)^2 \leq (m - 1)^2$$

$$= m^2 - 2m + 1$$
$$< m^2 - k_1$$
$$\Downarrow$$
$$m^2 - k_1 \neq (m - k_2)^2$$
$$\Downarrow$$
$$a^{m^2-k_1} b^{m-k_1} = uv^0 xy^0 z \notin L$$

But, from pumping lemma

But, from pumping lemma

$$uv^0 xy^0 z \in L$$
$$a^{m^2-k_1} b^{m-k_2} = uv^0 xy^0 z \notin L$$
$$\Downarrow$$

CONTRADICTION

$\therefore$ Our assumption that the language $L = \left\{a^{n^2} b^n : n \geq 0\right\}$ is context free is not true.

Conclusion : L is not context free.

Example 4.40 : Show that the set $L = \{a^i b^j c^k \mid k = \max(i, j)\}$ is not context free.

Solution : Given $L = \{a^i b^j c^k \mid k = \max(i, j)\}$

We prove it by contradiction

Assume L is context free with grammar G.

Let $W \in L$ and $|W| \geq m$;

Let $W = a^m b^m c^m$

By pumping lemma

$W = uvxyz$ satisfying the three conditions.

By the length condition, if vxy contains character of a single type, we are done, by 'pumping down or pumping up'. Otherwise, it can not contain both a and c.

1. vy contains c, then the number of c's in uxz is less than m (there are m of them altogether in w), while the number of a's in uxz is still m.

$\Downarrow$

CONTRADICTION

2. vy does not contain c. In this case, "pumping up" implies that either the number of a's or that of b's can be increased without altering the number of c's.

$\Downarrow$

CONTRADICTION

CFL closed → union, concatenation, star
CFL Not closed → Intersection, complement

$\therefore$ Our assumption that the language $L = \{a^i b^j c^k \mid k = \max(i, j)\}$ is not true.

Conclusion : L is not context free.