Module-3

Context-Free Grammar (CFG)

A context-free grammar (or CFG) to be a grammar in which each rule must:

* have a left-hand side that is a single nonterminal &
* have a right-hand that is ε or terminal, nonterminal one or more.

A context grammar G is 4 tuple (or) Quad tuple

$G = (V, T, P, S)$

where

V is set of variables / non terminals

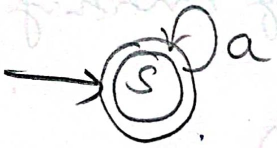
T is set of terminals

P is set of production

Each production is of the form $\alpha \to \beta$ where α is a string from variable & α cannot be ε, but β is string from $(V \cup T)^*$. Hence it can include ε also.

Eg Obtain grammar to generate string consisting of any no of a's

Soln:



$\delta(S, a) = S$ $\Rightarrow$ $S \to aS$

$S \to aS \mid \varepsilon$

$S \to \varepsilon$

$G = (V, T, P, S)$

$V = \{S\}$

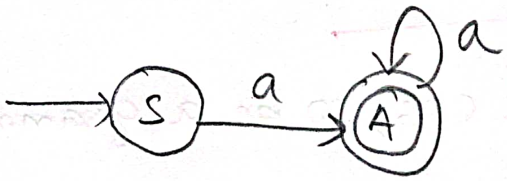
$T = \{a\}$

$P = \{ S \to aS \mid \varepsilon \}$

S is start symbol.

Eg Obtain a grammar to generate string consisting of atleast 1a.

Sol:



$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a\}$

$P = \{ S \rightarrow aA$

$A \rightarrow aA \mid \varepsilon \}$

S is start symbol.

$\delta(S, a) = A$

$S \rightarrow aA$

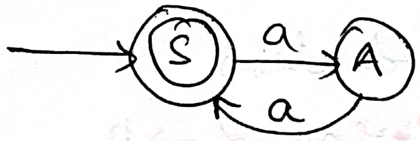
$\delta(A, a) = A$

$A \rightarrow aA$

$A \rightarrow \varepsilon$

Eg Obtain a grammar to generate string consisting of even no of a's

Sol:



$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a\}$

$P = \{ S \rightarrow aA \mid \varepsilon$

$A \rightarrow aS \}$

S is start symbol.

$\delta(S, a) = A$

$S \rightarrow aA$

$S \rightarrow \varepsilon$

$\delta(A, a) = S$

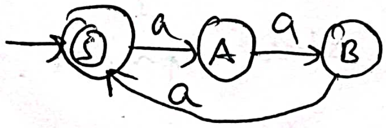
$A \rightarrow aS$

Eg Obtain a grammar to generate string consisting of multiples of 3 a's.

Sol:

$S \rightarrow aaaS \mid \varepsilon$

or



$G = (V, T, P, S)$

$V = \{S\}$

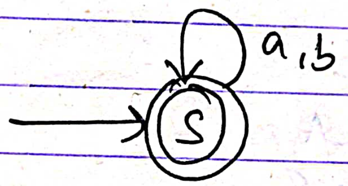
$T = \{a\}$

$P = \{ S \rightarrow aaaS \mid \varepsilon \}$

S is start symbol

③. obtain grammar to generate string consisting of any number of a's & b's

Soln



$S \rightarrow \varepsilon$

$S \rightarrow aS$

$S \rightarrow bS$

so $S \rightarrow \varepsilon / aS / bS$

$G = (V, T, P, S)$

$V = \{ S \}$

$T = \{ a, b \}$

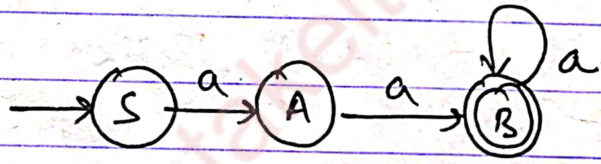
$P = \{$

$S \rightarrow \varepsilon / aS / bS \}$

S is start symbol

④ obtain grammar to generate string consisting of at least two a's

Soln



$S \rightarrow aA$

$A \rightarrow aB$

$B \rightarrow \varepsilon$

$B \rightarrow aB$

$G = (V, T, P, S)$

$V = \{ S, A, B \}$

$T = \{ a \}$

$P = \{$

$S \rightarrow aA$

$A \rightarrow aB$

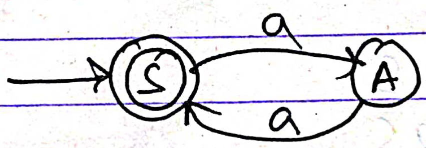
$B \rightarrow \varepsilon / aB \}$.

~~B→aB~~ }

S. is the start symbol

eg. obtain grammar to generate string consisting of even number of a's.

Sol^n



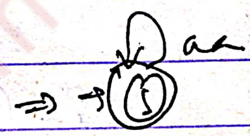
$S \to \varepsilon$

$S \to aA$

$A \to aS$

or

$S \to \varepsilon \mid aaS \Rightarrow$



$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a\}$

$P = \{ \; S \to \varepsilon \mid aA$

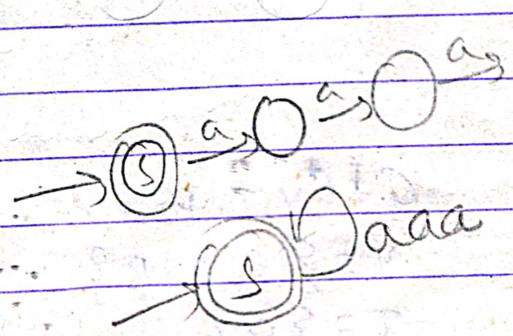
$\quad A \to aS \}$

where S is start symbol.

eg obtain grammar to generate string consisting of multiples of three a's.

Sol^n

$S \to \varepsilon \mid aaaS$



$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a\}$

$P = \{ \; S \to \varepsilon \mid aaaS \}$

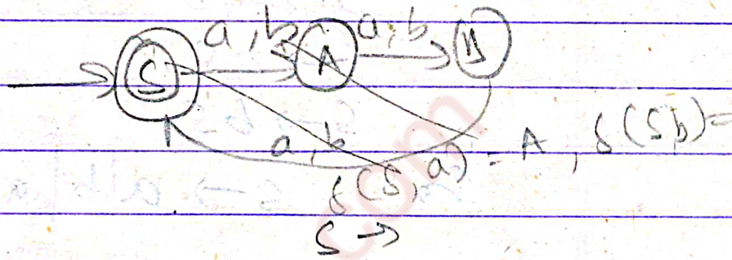
where S is start symbol.

Eg obtain grammar to generate strings of a's & b's such that string length is multiple of 3

Soln

$S \rightarrow \varepsilon / AAAS$

$A \rightarrow a/b.$



$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a, b\}$

$P = \{$

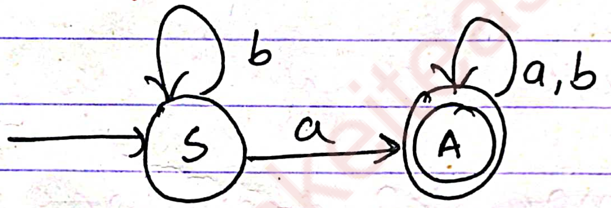
$S \rightarrow \varepsilon / AAAS$

$A \rightarrow a / b \}$

S is the start symbol.

Eg obtain grammar to generate string consisting of any number of a's & b's with atleast one a.

Soln



$S \rightarrow bS$

$S \rightarrow aA$

$A \rightarrow \varepsilon / aA / bA$

$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a, b\}$

$P = \{ S \rightarrow aA / bS$

$A \rightarrow aA / bA / \varepsilon \}$

S is the start symbol.

Eg Obtain grammar to generate string consisting of any no of a's & b's with atleast one a or atleast one b.

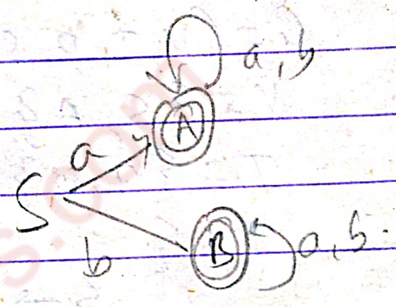
Sol

$S \to a|b$

$S \to aS$

$S \to bS$

So $S \to a|b|aS|bS$



$G = (V, T, P, S)$

$S \to aA|bB$

$V = \{S\}$

$T = \{a, b\}$

$S \to aA$

$A \to aA|bA|\varepsilon$

$P = \{ S \to a|b|aS|bS \}$.

$S \to bB$

$B \to aB|bB|\varepsilon$

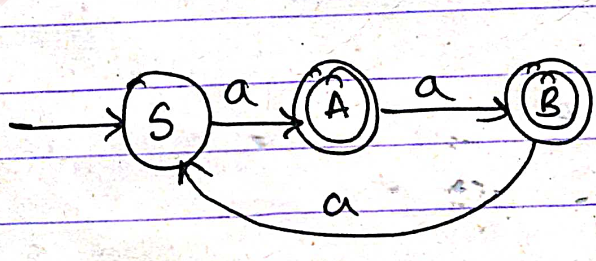
S is the start symbol.

Eg Obtain grammar to accept the following language

$L = \{ w : |w| \bmod 3 > 0 \text{ where } w \in \{a\}^* \}$

Sol.

$S \to a|aa|aaaS$



$S \to aA$

$1 \bmod 3 = 1$

$A \to aB|\varepsilon$

$B \to aS|\varepsilon$

$G = (V, T, P, S)$

$V = \{S, A, B\}$

$T = \{a\}$

$P = \{ S \rightarrow aA$

$A \rightarrow aB \mid \varepsilon$

$B \rightarrow aS \mid \varepsilon \}$

S is the start symbol.

Grammar from Regular expressions

Eg. obtain grammar to generate strings of a's & b's having a substring ab.

Soln. The RE representing strings of a's & b's having a substring is given by

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

$$\Downarrow \quad \Downarrow \quad \Downarrow$$

$$A \quad ab \quad A$$

$S \rightarrow AabA$

$A \rightarrow \varepsilon | aA | bA$

$G = (V, T, P, S)$

$V = \{S, A\}$

$T = \{a b\}$

$P = \{ S \rightarrow AabA$

$A \rightarrow \varepsilon | aA | bA \}$

S is start symbol.

Eg. obtain grammar to generate string of a's ending with string ab.

Soln

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

$$\Downarrow \quad \Downarrow$$

$$A \quad ab$$

$S \rightarrow Aab$

$A \rightarrow \varepsilon | aA | bA$

$G = (V, T, P, S)$
$V = \{S, A\}$
$T = \{a, b\}$
$P = \{$
$S \Rightarrow Aab$
$A \rightarrow aA \mid bA \mid \varepsilon \}$
S is the start symbol.

eg. obtain grammar to generate strings of a's & b's starting with ab.

Sol:

$$ab(a+b)^*$$
$$\Downarrow \quad \Downarrow$$
$$ab \quad A$$

$S \rightarrow abA$
$A \rightarrow \varepsilon \mid aA \mid bA$.
$G = (V, T, P, S)$
$V = \{S, A\}$
$T = \{a, b\}$
$P = \{ S \rightarrow abA$
$A \rightarrow \varepsilon \mid aA \mid bA \}$
S is the start symbol.

eg. obtain the grammar to generate the following language

$$L = \{w : n_a(w) \bmod 2 = 0 \text{ where } w \in \{a, b\}^*\}$$

Sol:

$$b^* a b^* a b^*$$
$$\Downarrow \; \Downarrow \; \Downarrow \; \Downarrow \; \Downarrow$$
$$S \; a \; S \; a \; S$$

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

$S \Rightarrow AaAaA \mid bA \mid \epsilon$

$A \rightarrow \epsilon \mid bA$

$S \rightarrow SaSaS$

$S \rightarrow \epsilon \mid bS$

$S \rightarrow \epsilon \mid bS \mid SaSaS$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{ S \rightarrow \epsilon \mid bS \mid SaSaS \}$

S is start symbol.

Grammars for other languages

Eg. Obtain a grammar to generate the following language

$$L = \{a^n b^n \mid n \geq 0\}.$$

Sol:

$$S \rightarrow \epsilon \mid aSb$$

Eg: $S \Rightarrow \epsilon$

$S \Rightarrow aSb$

$\Rightarrow ab$

$S \Rightarrow aSb$

$\Rightarrow aaSbb$

$\Rightarrow aaabbb$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{ S \rightarrow \epsilon \mid aSb \}$

S is the start symbol

(Eg). obtain a grammar to generate the following language

$$L = \{a^n b^n : n \geq 1\}$$

soln.

$$L = \{ ab, aabb, aaabbb \ldots \}$$

$$S \to ab \mid aSb$$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \to ab \mid aSb\}$

S is the start symbol.

(Eg). obtain a grammar to generate the following lang

$$L = \{a^{n+1} b^n : n \geq 0\}$$

soln.

$$L = \{ a\boxed{\epsilon}, a\boxed{ab}, a\boxed{aabb}, a\boxed{aaabbb} \ldots \}$$

$$S \to a \mid aSb$$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \to a \mid aSb\}$

S is start symbol.

Eg) obtain a grammar to generate the following lang

$L = \{a^n b^{n+1} : n \geq 0\}$.

Soln.

$S \rightarrow b \mid aSb$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \rightarrow b \mid aSb\}$

S is start symbol.

Eg) obtain a grammar to generate the following lan

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

Soln Two extra b's should be generated. So, the final grammar to generate given lang is

$S \rightarrow bb \mid aSb$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \rightarrow bb \mid aSb\}$

S is start symbol.

Eg obtain a grammar to generate the following language

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

Soln

$S \to \varepsilon \mid aSbb$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \to \varepsilon \mid aSbb\}$

S is start symbol.

Eg Let $\Sigma = \{a, b\}$ obtain a grammar G generating set of all palindromes over Σ.

Soln.

$S \to \varepsilon \mid a \mid b \mid aSa \mid bSb$

Madam

$S \to mSm \mid aSa \mid$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{ S \to \varepsilon$

$S \to a \mid b$

$S \to aSa \mid bSb \}$

S is the start symbol.

(Eg) Obtain a grammar to generate a lang consisting of all non-palindromes over {a, b}

Sol)

$G = (V, T, P, S)$

$V = \{S, A, B\}$

$T = \{a, b\}$

$P = \{$

$S \rightarrow aSa \mid bSb$

$S \rightarrow A$

$A \rightarrow aBb \mid bBa$

$B \rightarrow aB \mid bB \mid \varepsilon \}$

S is start symbol.

(Eg) Obtain a grammar to generate the following language

$L = \{WW^R \text{ where } W \in \{a, b\}^*\}$.

$L = \{aa, bb, abba, baab, \ldots\}$

Sol) $S \rightarrow \varepsilon \mid aSa \mid bSb$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a, b\}$

$P = \{S \rightarrow \varepsilon \mid aSa \mid bSb\}$

S is start symbol.

(Eg) Obtain the grammar to generate the language

$$L = \{0^m 1^m 2^n \mid m \geq 1 \text{ \& } n \geq 0\}$$

Sol^n

$$L = \{\underbrace{0^m 1^m}\, 2^n \mid m \geq 1 \text{ \& } n \geq 0\}$$

$$S \rightarrow A \; B$$

The variable A should produce m no of 0's & followed by m no. of 1's.

$$A \rightarrow 01 \mid 0A1$$

B should produce any no of 2's

$$B \rightarrow \varepsilon \mid 2B$$

$$S \rightarrow A B$$
$$A \rightarrow 01 \mid 0A1$$
$$B \rightarrow \varepsilon \mid 2B$$

$$G = (V, T, P, S)$$

$$V = \{S, A, B\}$$

$$T = \{0, 1, 2\}$$

$$P = \{ S \rightarrow A B,$$
$$A \rightarrow 01 \mid 0A1$$
$$B \rightarrow \varepsilon \mid 2B \}$$

S is start symbol.

Eg Obtain the grammar to generate the lang

$$L = \{ w \mid n_a(w) = n_b(w) \}$$

Sol^n $S \to \varepsilon \mid aSb \mid bSa \mid SS$ → string starting & ending with the same str

Eg: abba, baab

$G = (V, T, P, S)$

$V = \{ S \}$

$T = \{ a, b \}$

$P = \{$

$S \to \varepsilon \mid aSb \mid bSa \mid SS$

$\}$

S is start symbol.

Eg Obtain a CF grammar to generate a string balanced parentheses.

Sol^n: $S \to (S) \mid \varepsilon \mid SS$

$G = (V, T, P, S)$

$V = \{ S \}$

$T = \{ (,) \}$

$P = \{ S \to (S) \mid SS \mid \varepsilon \}$

S is start symbol.

(eg) obtain a grammar to generate the lang:

$$L = \{a^i b^j \mid i \neq j,\ i \geq 0 \ \&\ j \geq 0\}$$

Soln

$L = \{a^i b^j \mid i \neq j, i \geq 0, j \geq 0\}$

$S \to aSb \mid A \mid B$

$A \to a \mid aA$

$B \to b \mid bB$

$G = (V, T, P, S)$

$V = \{S, A, B\}$

$T = \{a, b\}$

$P = \{ S \to aSb \mid A \mid B$

$A \to a \mid aA$

$B \to b \mid bB \}$

S is start symbol.

$L = \{0^i 1^j \mid i \neq j, i \geq 0, j \geq$

$S \to 0S1 \mid A \mid B$

$A \to 0 \mid 0A$

$B \to 1 \mid 1B$

$G = (V, T, P, S)$

$V = \{S, A, B\}$

$T = \{0, 1\}$

$P = \{ S \to 0S1 \mid A \mid B$

$A \to 0 \mid 0A$

$B \to 1 \mid 1B$

(eg) obtain a grammar to generate the language

$$L = \{a^{n+2} b^m \mid n \geq 0 \ \&\ m > n\}$$

Soln

$S \to aAB$

$A \to aAb \mid ab.$

$B \to bB \mid \epsilon$

$G = (V, T, P, S)$

$V = \{S, A, B\}$, $T = \{a, b\}$ $P = \{ S \to aAB,$

$A \to aAb \mid ab$

$B \to bB \mid \epsilon \}$

S is start symbol

eg) obtain a grammar to generate the language

$$L = \{a^n b^m \mid n \geq 0, m > n\}$$

Sol^n:

$n=0$	$n=1$	$n=2$
$m \geq 1$	$m \geq 2$	$m \geq 3$

$$L = \{\boxed{\varepsilon}\, bb^*,\ \boxed{ab}\, bb^*,\ \boxed{aabb}\, bb^*, \ldots\}$$

$S \rightarrow aSb \mid B$ [Generates $a^n b^n \mid n \geq 0$ followed by atleast one b]

$B \rightarrow bB \mid b$ [Generates one or more b's]

$G = (V, T, P, S)$

$V = \{S, B\}$

$T = \{a, b\}$

$P = \{S \rightarrow aSb \mid B,$
$\quad B \rightarrow bB \mid b\}$

S is start symbol.

eg) obtain a grammar to generate the language

$$L = \{a^n b^{n-3} \mid n \geq 3\}$$

Sol^n:

$n=3$	$n=4$	$n=5$

$L = \{aaa, aaaab, aaaaabb, \ldots\}$

$P = \{ S \rightarrow aaaA$
$\quad A \rightarrow aAb \mid \varepsilon \}$

Derivation.

Defn: Let $A \rightarrow \alpha B \gamma$ & $B \rightarrow \beta$ are production in grammar G, where α, β and γ are strings of termi- and/or non-terminals. A and B are non-termina- The non-terminal A produces the string $\alpha \beta \gamma$ by replacing the non-terminal B in $\alpha B \gamma$ by the string β by applying the production $B \rightarrow \beta$ & can be written as

$$A \Rightarrow \alpha \beta \gamma$$

This process of obtaining string of terminals &/or non terminals from the start symbol by applying some or all productions is called derivation.

If a string is obtained by applying only one prodn, then it is called on-step derivation & is denoted by the symbol '$\Rightarrow$'. If one or more productions are applied to get the string $\alpha\beta\gamma$ from A then we write

$$A \Rightarrow^{+} \alpha \beta \gamma$$

If zero or more prodns are applied to get the string $\alpha\beta\gamma$ from A, then we write

$$A \overset{*}{\Rightarrow} \alpha \beta \gamma .$$

Eg IA consider the grammar shown below from which any arithmetic expⁿ can be obtained.

$$E \to E+E$$
$$E \to E-E$$
$$E \to E*E$$
$$E \to E/E$$
$$E \to id.$$

E is considered to be the start symbol. obtain the string id+id*id & show the derivation for the same

Solⁿ:
$$E \Rightarrow E+E$$
$$\Rightarrow id+E$$
$$\Rightarrow id+E*E$$
$$\Rightarrow id+id*E$$
$$E \Rightarrow id+id*id$$

Sentence or Sentential form

Defⁿ: Let $G=(V,T,P,S)$ be a grammar. The string w obtained from the grammar G such that $S \Rightarrow^{+} w$ is called Sentence of grammar G. Here w is string of terminals.

In the above eg id+id*id is the sentence of the grammar.

If there is a derivation $S \stackrel{*}{\Rightarrow} \alpha$, where α conta

strings of terminals and/or non-terminals, then α is called sentential form of G.
In the derivation shown in above eg.

$$E+E,\ id+E,\ id+E*E,\ id+id*E,\ id+id*id$$

are all sentential forms of ur grammar.

Language

Def^n: Let $G=(V,T,P,S)$ be a grammar. The language $L(G)$ generated by the grammar G is

$$L(G)=\{w \mid S \overset{*}{\Rightarrow} w \text{ \& } w \in T^*\}$$

i.e, w is a string of terminals (may be ϵ) obtained from the start symbol S by applying an arbitrary number of productions.

Leftmost Derivation

Def^n: In the derivation process if a left most variable is replaced at every step, then the derivation is said to be leftmost.

Eg obtain the leftmost derivation for the string id+id*id using the following grammar

$E \rightarrow E+E$

$E \rightarrow E-E$

$E \rightarrow E*E$

$E \rightarrow E|E$

$E \rightarrow E \wedge E$ $\quad$ ^: denotes exponential

$E \rightarrow id$.

Sol^n

$$E \underset{lm}{\overset{+}{\Rightarrow}} E+E$$

$$\Rightarrow id+E$$

$$\Rightarrow id+E*E$$

$$\Rightarrow id+id*E$$

$$\Rightarrow id+id*id$$

The string id+id*id is obtained from the start symbol E by applying leftmost derivation Eq can be written as

$$E \underset{lm}{\overset{+}{\Rightarrow}} id+id*id.$$

Eg) Obtain the leftmost derivation for the string aaabbabbba using the following grammar

$S \rightarrow aB \mid bA$

$A \rightarrow aS \mid bAA \mid a$

$B \rightarrow bS \mid aBB \mid b$.

Sol)

$S \underset{lm}{\Rightarrow} aB$ (Applying $S \rightarrow aB$

$\Rightarrow aaBB$ ($B \rightarrow aBB$)

$\Rightarrow aaaBBB$ ($B \rightarrow aBB$)

$\Rightarrow aaabBB$ ($B \rightarrow b$)

$\Rightarrow aaabbB$ ($B \rightarrow b$)

$\Rightarrow aaabbaBB$ ($B \rightarrow aBB$)

$\Rightarrow aaabbabB$ ($B \rightarrow b$)

$\Rightarrow aaabbabbS$ ($B \rightarrow bS$)

$\Rightarrow aaabbabbbA$ ($S \rightarrow bA$)

$\Rightarrow aaabbabbba$ ($A \rightarrow a$)

Rightmost Derivation

Def^n: In the derivation process if a right most variable is replaced at every step, then the derivation is said to be rightmost.

Eg obtain the rightmost derivation for the string id+id*id using following grammar.

$E \rightarrow E+E$
$E \rightarrow E*E$
$E \rightarrow E-E$
$E \rightarrow E/E$
$E \rightarrow id$

Sol^n

$$E \underset{rm}{\Rightarrow} E+E$$
$$\Rightarrow E+E*E$$
$$\Rightarrow E+E*id$$
$$\Rightarrow E+id*id$$
$$\Rightarrow id+id*id.$$

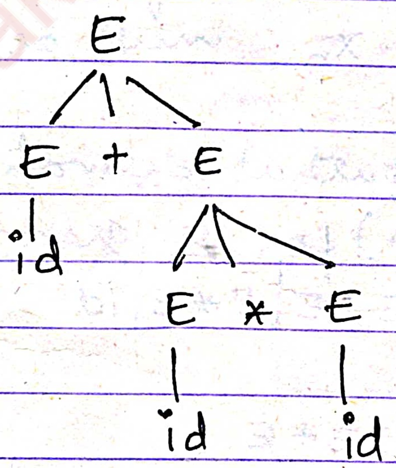
Derivation Tree (parse tree)

The derivation can be shown in the form of a tree. Such trees are called derivation or parse trees.

Def^n (Parse tree or Derivation tree) : Let $G = (V, T, P, S)$ be a CFG. The tree is derivation tree (parse tree) with the following properties.

1. The root has the label S
2. Every vertex has a label which is in $(V \cup T \cup \epsilon)$
3. Every leaf node has label from T & an interior Vertex has a label from V.
4. If a vertex is labeled A & if $X_1, X_2, X_3 \ldots X_n$ are all children of A from left, then
   $A \rightarrow X_1 X_2 X_3 \ldots X_n$ must be a production in P.

parse tree for above eg.



The string id+id*id is called yield of the tree.

Yield of the tree

The yield of a tree is the string of symbols obtained by only reading the leaves of the tree from left to right with considering the ε-symbols. The yield of a tree is always a terminal string.

Def^n (partial parse tree or partial Derivation tree): Let $G = (V, T, P, S)$ be a CFG. The tree is partial derivation tree (parse tree) with the following properties:

1. The root has the label S
2. Every vertex has a label which is in $(V \cup T \cup \epsilon)$
3. Every leaf node has a label from $(V \cup T \cup \epsilon)$.
4. If a vertex is labeled A & if $X_1, X_2, X_3 \cdots X_n$ are all children of A from left. then $A \rightarrow X_1 X_2 X_3 \cdots X_n$ must be a prod^n in P.

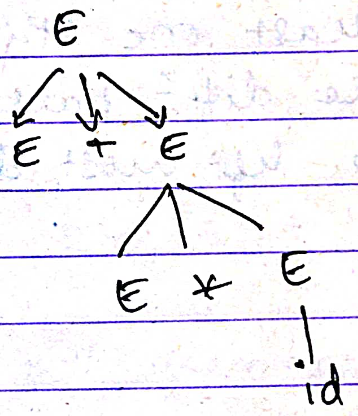
Eg consider the partial derivations (by applying right most der^n) for the grammar.

$$E \rightarrow E+E \mid E*E \mid id.$$

is shown below

$$E \rightarrow E+E$$
$$\Rightarrow E+E*E$$
$$\Rightarrow E+E*id$$

For this partial right most derivation, the partial derivation tree is shown below



It is clear from the parse tree & partial parse tree that all the leaves in parse tree are the symbols from (TUE) whereas in partial parse tree the leaves will be from (VUTUE).

Ambiguous Grammar

Defn: Let $G = (V, T, P, S)$ be a context free grammar. A grammar G is ambiguous if & if there exists at least one string $w \in T^*$ for which two or more diffn parse trees exists by applying either the left most derivation or right most derivation.

Note:

1) obtain the leftmost derivation & get a string w. obtain the rightmost derivation & get a string w. For both the derivations construct the parse tree. If there are two diffn parse trees, then the grammar is ambiguous.

2) obtain the string w by applying leftmost derivation twice & construct the parse tree. If the two parse trees are different, the grammar is ambiguous.

3) obtain the string w by applying rightmost derivation twice & construct the parse tree. If the two parse trees are diffn, the grammar is ambiguous.

eg) Consider the grammar shown below

$E \to E+E$

$E \to E-E$

$E \to E*E$

$E \to E/E$

$E \to (E) \mid I$

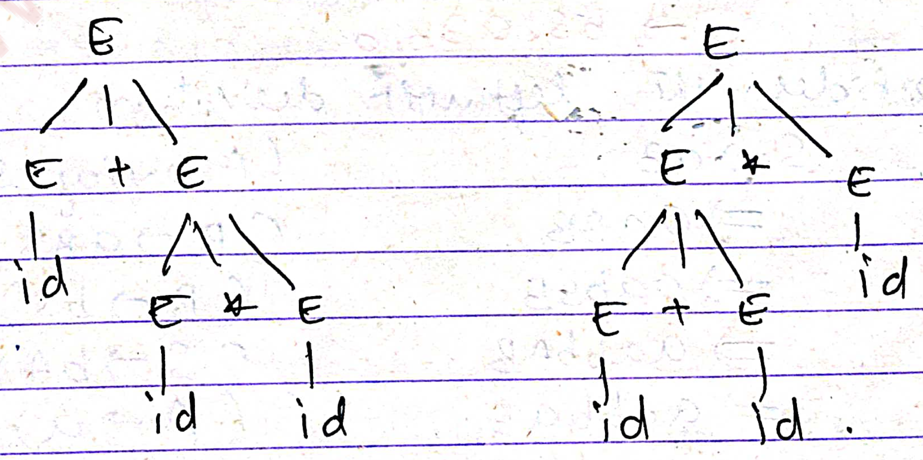
$I \to id$

Show that the grammar is ambiguous.

Sol? The sentence $id+id*id$ can be obtained from leftmost derivation in two ways as shown below

$E \Rightarrow E+E$	$E \Rightarrow E*E$
$\Rightarrow id+E$	$\Rightarrow E+E*E$
$\Rightarrow id+E*E$	$\Rightarrow id+E*E$
$\Rightarrow id+id*E$	$\Rightarrow id+id*E$
$\Rightarrow id+id*id$	$\Rightarrow id+id*id$

The corresponding derivation trees for the two leftmost ~~derivat~~ derivation are shown below



Since the two parse trees are diff^n for the same sentence $id+id*id$ by applying leftmost derivation, the grammar is ambiguous.

Eg Is the following grammar ambiguous?

$$S \rightarrow aS \mid X$$
$$X \rightarrow aX \mid a$$

Sol^n Consider the two leftmost derivations for the string aaaa.

$S \Rightarrow aS$	$S \Rightarrow X$
$\Rightarrow aaS$	$\Rightarrow aX$
$\Rightarrow aaaS$	$\Rightarrow aaX$
$\Rightarrow aaaX$	$\Rightarrow aaaX$
$\Rightarrow aaaa$	$\Rightarrow aaaa$

$S \Rightarrow aS \Rightarrow aX \Rightarrow aa$

Since there are two leftmost derivations for the same sentence aaaa, the given grammar is ambiguous.

Show that the grammar is ambiguous for the string aabbab

Eg Is the following grammar ambiguous?

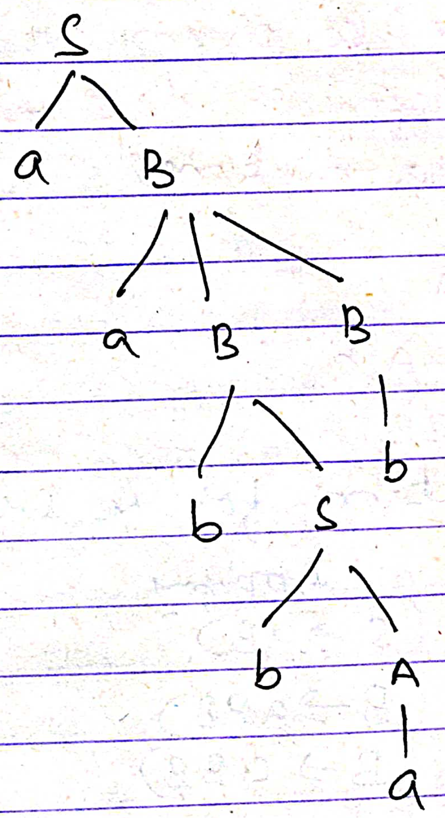
$$S \rightarrow aB \mid bA$$
$$A \rightarrow aS \mid bAA \mid a$$
$$B \rightarrow bS \mid aBB \mid b$$

$S \Rightarrow aB \Rightarrow abS \Rightarrow abaB \Rightarrow abab$

Sol? Consider the leftmost derivation

$S \Rightarrow aB$	(Applying $S \rightarrow aB$)
$\Rightarrow aaBB$	($B \rightarrow aBB$)
$\Rightarrow aabSB$	($B \rightarrow bS$)
$\Rightarrow aabbAB$	($S \rightarrow bA$)
$\Rightarrow aabbaB$	($A \rightarrow a$)
$\Rightarrow aabbab$	($B \rightarrow b$)

Parse tree



The same string aabbab can be obtained again by applying leftmost derivation as shown below.

Applying

$S \Rightarrow aB \quad (S \rightarrow aB)$

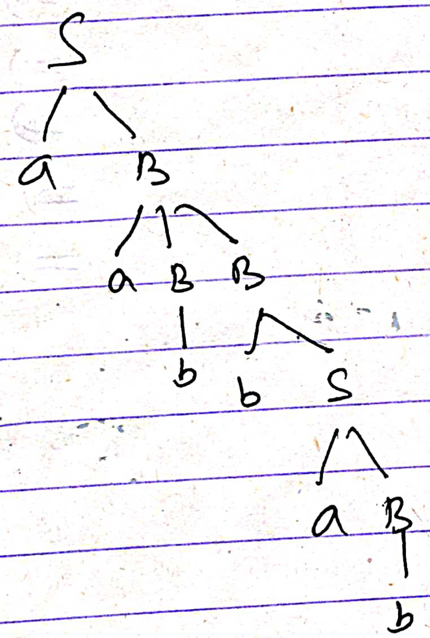
$\Rightarrow aaBB \quad (B \rightarrow aBB)$

$\Rightarrow aabB \quad (B \rightarrow b)$

$\Rightarrow aabbS \quad (B \rightarrow bS)$

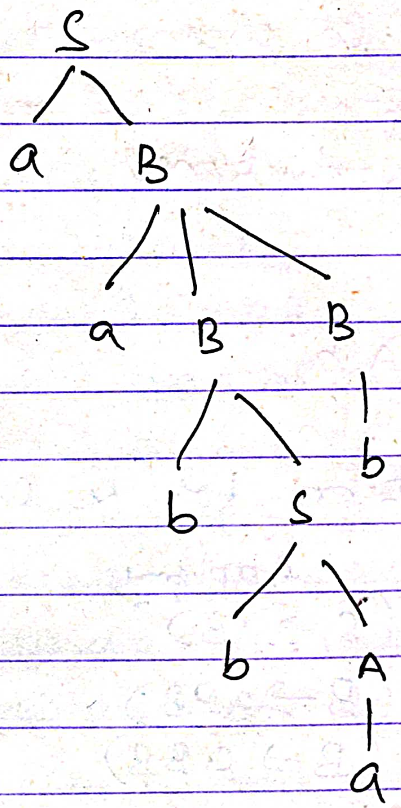
$\Rightarrow aabbaB \quad (S \rightarrow aB)$

$\Rightarrow aabbab \quad (B \rightarrow b)$



There are two parse trees for the string aabbab by applying leftmost derivation & so the given grammar is ambiguous.

Parse tree



The same string aabbab can be obtained again by applying leftmost derivation as shown below.

Applying

$S \Rightarrow aB \quad (S \rightarrow aB)$

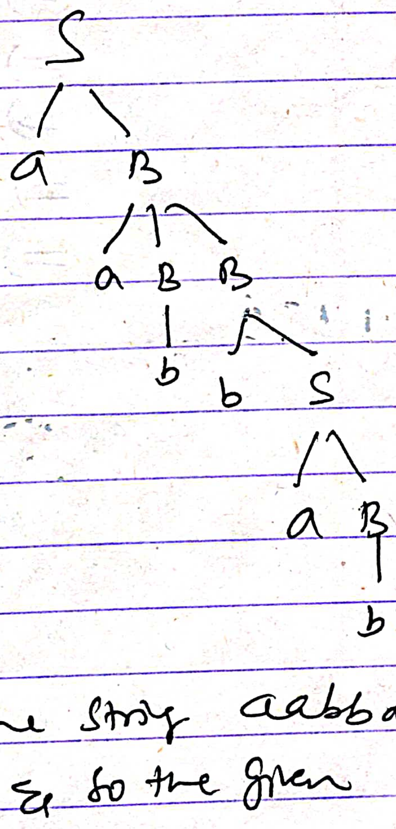
$\Rightarrow aaBB \; (B \rightarrow aBB)$

$\Rightarrow aabB \; (B \rightarrow b)$

$\Rightarrow aabbS \; (B \rightarrow bS)$

$\Rightarrow aabbaB \; (S \rightarrow aB)$

$\Rightarrow aabbab \; (B \rightarrow b)$



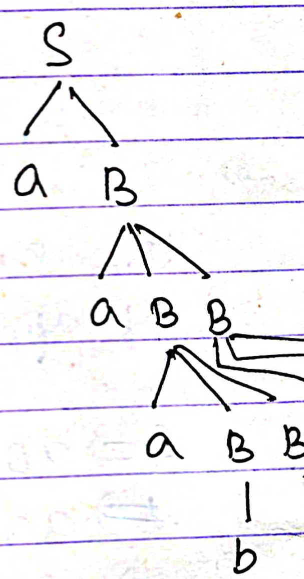
There are two parse trees for the string aabbab by applying leftmost derivations & so the given grammar is ambiguous.

Eg Obtain the string aaabbabbba by applying leftmost derivation & the parse tree for the grammar shown below. Is it possible to obtain the same string again by applying leftmost derivation but by selecting different productions?

$$S \to aB \mid bA$$
$$A \to aS \mid bAA \mid a$$
$$B \to bS \mid aBB \mid b$$

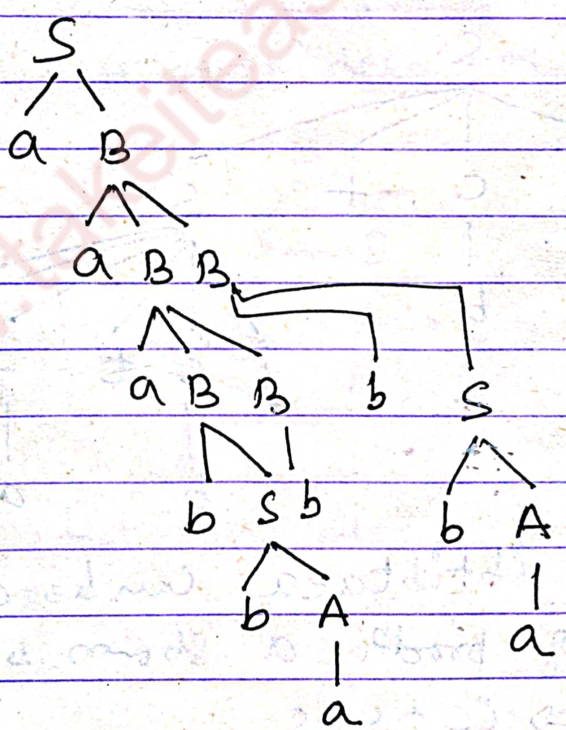
Sol: The leftmost derivation for the string aaabbabbb

	Applying
$S \Rightarrow aB$	$(S \to aB)$
$\Rightarrow aaBB$	$(B \to aBB)$
$\Rightarrow aaaBBB$	$(B \to aBB)$
$\Rightarrow aaabBB$	$(B \to b)$
$\Rightarrow aaabbB$	$(B \to b)$
$\Rightarrow aaabbaBB$	$(B \to aBB)$
$\Rightarrow aaabbabB$	$(B \to b)$
$\Rightarrow aaabbabbS$	$(B \to bS)$
$\Rightarrow aaabbabbbA$	$(S \to bA)$
$\Rightarrow aaabbabbba$	$(A \to a)$



The leftmost derⁿ for the same string aaabbabbba but by applying diffⁿ set of prodⁿs is shown below:

$$
\begin{aligned}
S &\Rightarrow aB & (S \rightarrow aB)\\
&\Rightarrow aaBB & (B \rightarrow aBB)\\
&\Rightarrow aaaBBB & (B \rightarrow aBB)\\
&\Rightarrow aaabSBB & (B \rightarrow bS)\\
&\Rightarrow aaabbABB & (S \rightarrow bA)\\
&\Rightarrow aaabbaBB & (A \rightarrow a)\\
&\Rightarrow aaabbabB & (B \rightarrow b)\\
&\Rightarrow aaabbabbS & (B \rightarrow bS)\\
&\Rightarrow aaabbabbbA & (S \rightarrow bA)\\
&\Rightarrow aaabbabbba & (A \rightarrow a)
\end{aligned}
$$



(Eg) Is the following grammar ambiguous?

$S \rightarrow ictS \mid ictSeS \mid a$

$C \rightarrow b$

Soln. The String ibtibtaea can be obtained by applying the leftmost derivn as shown below

$S \Rightarrow ictS$

$\Rightarrow ibtS$

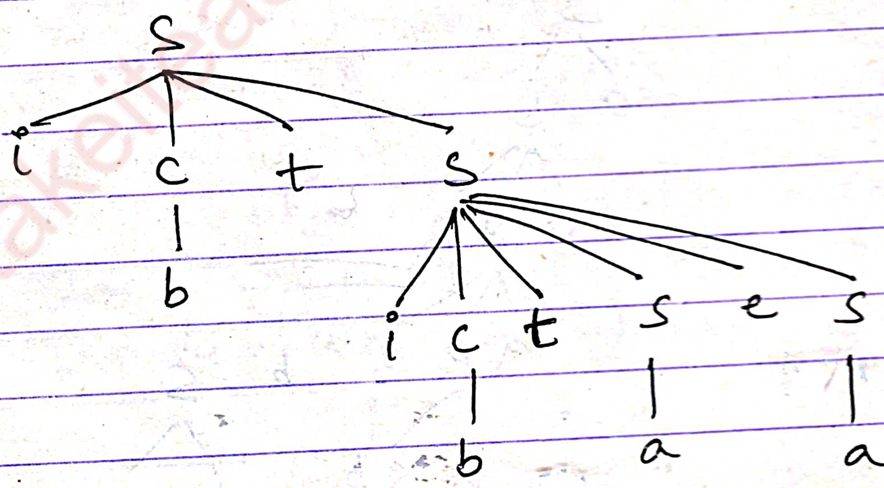
$\Rightarrow ibtictSeS$

$\Rightarrow ibtibtSeS$

$\Rightarrow ibtibtaeS$

$\Rightarrow ibtibtaea$

parse tree



The String ibtibtaea canbe obtained by using diffn set of prodn a shown below -

$S \Rightarrow ictSeS$

$\Rightarrow ibtSeS$

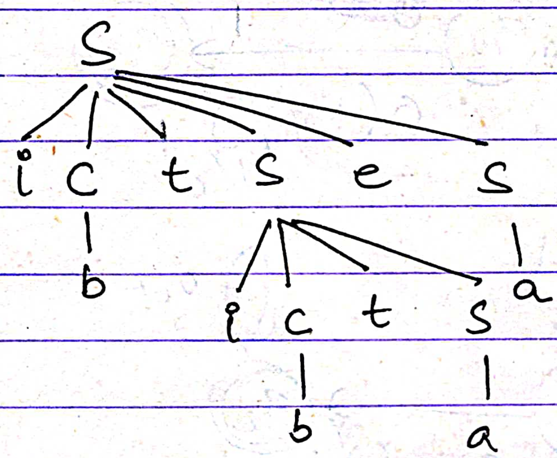
$\Rightarrow ibtictSeS$

$\Rightarrow ibtibtSeS$

$\Rightarrow ibtibtaeS$

$\Rightarrow ibtibtaea$

The parse tree for this is shown below



Q) Is the grammar ambiguous?

$S \rightarrow AB \mid aaB$

$A \rightarrow a \mid Aa$

$B \rightarrow b.$

$S \Rightarrow AB$ $\Rightarrow AaB$ $\Rightarrow aab$ $S \Rightarrow aaB$ $\Rightarrow aab$

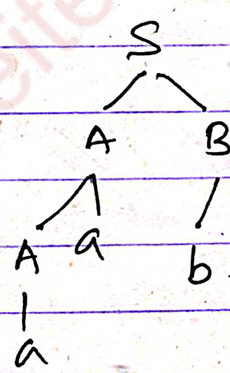
eg1) consider the left most derⁿ for the string aab

$S \Rightarrow AB$

$\Rightarrow AaB$

$\Rightarrow aaB$

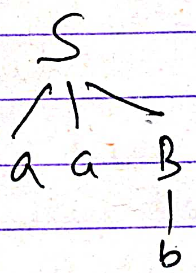
$\Rightarrow aab$



consider the leftmost derⁿ again for the string aab

$S \Rightarrow aaB$

$\Rightarrow aab$



Since there are two parse trees for the string aab, the given grammar is ambiguous.

Properties of context-free languages

The goal of this section is to show that every CFL (without ϵ) is generated by a CFG in which all prodns are of the form $A \to BC$ or $A \to a$, where A, B & C are variables & a is a terminal. This form is called Chomsky ~~Normal~~ Normal form (CNF).

To get there, we need to make a number of preliminary simplifications, which are themselves useful in various ways:

1) We must eliminate useless symbols, those variables or terminals that do not appear in any derivation of a terminal string from the start symbol.

2) We must eliminate ϵ-prodns, those of the form $A \to \epsilon$ for some variable A.

3) We must eliminate unit prodn, those of the form $A \to B$ for variable A & B.

Eliminating useless symbols:

We say a symbol x is useful for a grammar $G = (V, T, P, S)$ if there is a some derivation of the form $S \overset{*}{\Rightarrow} \alpha x \beta \overset{*}{\Rightarrow} w$, where w is in T^*. Note that x may be in either V or T, & the sentential form $\alpha x \beta$ might be the first or last in the derivation. If x is not useful, we say it is useless. Evidently, omitting useless symbols from a grammar will not change the language generated. So we may as well detect & eliminate all useless symbols

Our approach to eliminating useless symbols begins by identifying the two things a symbol has to be able to do to be useful:

1) We say x is generating if $x \overset{*}{\Rightarrow} w$ for some terminal string w. Note that every terminal is generating, since w can be that terminal itself, which is derived by zero steps.

2) We say x is reachable if there is a derivation $S \overset{*}{\Rightarrow} \alpha x \beta$ for some α & β.

Surely a symbol that is useful will be both generating & reachable. If we eliminate the symbols that are not generating first & then eliminate from the remaining grammar those symbols that are not reachable we have only the useful symbols left.

Eg) Eliminate useless symbols from the following grammar

$S \to AB \mid a$

$A \to a.$

1st step: find the set of var & the prodns from which we get only string of terminals.

OV	NV	prodn.
ϕ	S, A	$S \to a$ $A \to a$
S, A	S, A	$S \to a$ $A \to a$

The resulting grammar, from which we get only string of terminals is given by.

$$G_1 = (V_1, T_1, P_1, S)$$

Where

$V_1 = \{S, A\}$

$T_1 = \{a\}$

$P_1 = \{S \to a, A \to a\}$

Step 2: Eliminate the symbol & production which are not reachable from the start symbol.

P'	T'	V'
–	–	S
$S \to a$	a	S

So the final grammar obtained after eliminating useless symbols & productions is given by

$$G' = (V', T', P', S)$$

Where
$V' = \{S\}$

$T' = \{a\}$

$P' = \{S \to a\}$

S is the start symbol.

Eliminate the useless symbols in the grammar

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

OV	NV	Prod^ns
ϕ	A, E, D	$A \to a$ $E \to d$ $D \to ab$
A, E, D	A, E, D, S	$S \to aA$ $A \to aA \mid a$ $D \to ab \mid Ea$
A, D, E, S	A, D, E, S	–

The resulting grammar $G_1 = (V_1, T_1, P_1, S)$ where

$V_1 = \{A, D, E, S\}$
$T_1 = \{a, b, d\}$
$P_1 = \{ S \to aA,\ A \to aA \mid a,\ D \to ab \mid Ea,\ E \to d \}$
S is the start symbol.

P'	T'	V'
–	–	S
$S \to aA$	a	S, A
$A \to aA \mid a$	a	S, A

The resulting grammar $G' = (V', T', P', S)$ where

$V' = \{S, A\}$

$T' = \{a\}$

$P' = \{S \to aA,\ A \to aA \mid a\}$

}

S is the start symbol.

Eg) Eliminate useless symbols from the following grammar

$S \to aAB$

$A \to aB \mid B$

$B \to aB \mid b \mid bC$

$D \to Ea$

$E \to a \mid aE \mid bC$

Sol)

OV	NV	production.
$\emptyset$	B, E	$B \to b$ $E \to a$
B, E	A, B, D, E	$A \to aB \mid B$ $B \to aB$ $D \to Ea$ $E \to aE$

OV	NV	Prod^n.
A, B, D, E	S, A, B, D, E	$S \to aAB$
S, A, B, D, E	–	–

$G_1 = (V_1, T, P_1, S)$

$V_1 = \{S, A, B, D, E\}$

$T_1 = \{a, b\}$

$P_1 = \{S \to aAB,\ A \to aB \mid B,\ B \to aB \mid b,\ D \to Ea,$
$E \to aE \mid a\}$

12

Prod^n (P')	T'	V'
–	–	S
$S \to aAB$	a	S, A, B
$A \to aB \mid B$	a	S, A, B
$B \to aB \mid b$	a, b	S, A, B.

$G' = (V', T', P', S)$

$V' = \{S, A, B\}$

$T' = \{a, b\}$

$P' = \{S \to aAB,\ A \to aB \mid B,\ B \to aB \mid b\}$

S is the start symbol.

Eg Eliminate useless symbols from the following grammar

$S \to aA \mid a \mid Bb \mid cC$

$A \to aB$

$B \to a \mid Aa$

$C \to cCD$

$D \to ddd$

Sol: Step 1:

OV	NV	prod^n.
ϕ	S, B, D	$S \to a$ $B \to a$ $D \to ddd$
S, B, D	S, A, B, D	$S \to Bb$ $A \to aB$
S, A, B, D	S, A, B, D	$S \to aA$ ~~$A \to aB$~~ $B \to Aa$

$G_1 = (V_1, T_1, P_1, S)$

$V_1 = \{S, A, B, D\}$

$T_1 = \{a, b, d\}$

$P_1 = \{ S \to a \mid Bb \mid aA,\ A \to aB,\ B \to a \mid Aa,\ D \to ddd \}$

Production (P')	T'	V'
–	–	S
$S \to a \mid Bb \mid aA$	a, b	S, A, B
$A \to aB$	a, b	S, A, B
$B \to a \mid Aa$	a, b	S, A, B

The resulting grammar is

$V' = \{S, A, B\}$
$T' = \{a, b\}$
$P' = \{S \to a \mid Bb \mid aA,\ A \to aB,\ B \to a \mid Aa\}$

S is the start symbol.

Eliminating ε-productions.

A production of the form $A \to \varepsilon$ is undesirable in a CFG, unless an empty string is derived from the start symbol. Suppose, the language generated from a grammar G does not derive any empty string & the grammar consist of ε-productions. Such ε-prod^{ns} can be removed.

Def^n: Let $G = (V, T, P, S)$ be a CFG. A production in P of the form

$$A \to \varepsilon$$

is called an ε-prod^n or null prod^n. After applying the production the variable A is erased. For each A in V, if there is a derivation of the form

$$A \overset{*}{\Rightarrow} \varepsilon$$

then A is a nullable variable.

Eg: In grammar if there is are prods of the form

$$B \to \varepsilon$$

$$C \to \varepsilon$$

are ε-prod^{ns} and the variable B, C are nullable variables. If there is a production

$$A \to BC$$

as both B, C are nullable variables, then A is also a nullable variable.

Eliminate all ε-productions from the grammar

$S \to ABCa \mid bD$

$A \to BC \mid b$

$B \to b \mid \varepsilon$

$C \to c \mid \varepsilon$

$D \to d$

$B \to \varepsilon$

$C \to \varepsilon$

$A \to BC$

$V = \{B, C, A\}$ are all nullable variables.

productions	Resulting prodns
$S \to ABCa$	$S \to ABCa \mid BCa \mid ACa \mid ABa \mid Ca \mid Aa \mid Ba \mid a$
$S \to bD$	$S \to bD$
$A \to BC \mid b$	$A \to BC \mid B \mid C \mid b$
$B \to b \mid \varepsilon$	$B \to b$
$C \to c \mid \varepsilon$	$C \to c$
$D \to d$	$D \to d$

The grammar $G' = (V', T', P', S)$ where

$V' = \{S, A, B, C, D\}$

$T' = \{a, b, c, d\}$

$P' = \{ S \to ABCa \mid BCa \mid ACa \mid ABa \mid Ca \mid Aa \mid Ba \mid a \mid bD$

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

$B \to b$

$C \to c$

$D \to d \}$

S is the start symbol.

Eg Eliminate all ε-productions from the grammar

$S \to BAAB$

$A \to 0A2 \mid 2A0 \mid \varepsilon$

$B \to AB \mid 1B \mid \varepsilon$

Sol) $A \to \varepsilon$

$B \to \varepsilon$

$S \to BAAB$

So S, A, B are all nullable variables.

productions	Resulting prodns (P')
$S \to BAAB$	$S \to BAAB \mid AAB \mid BAB \mid BAA \mid AB \mid BB \mid BA \mid AA \mid A \mid B$
$A \to 0A2$	$A \to 0A2 \mid 02$
$A \to 2A0$	$A \to 2A0 \mid 20$
$B \to AB$	$B \to AB \mid B \mid A$
$B \to 1B$	$B \to 1B \mid 1$

The grammar $G' = (V', T', P', S)$ where

$V' = \{S, A, B\}$

$T' = \{0, 1, 2\}$

$P' = \{ S \to BAAB \mid AAB \mid BAB \mid BAA \mid AB \mid BB \mid BA \mid AA \mid A \mid B$

$A \to 0A2 \mid 02 \mid 2A0 \mid 20$

$B \to AB \mid B \mid A \mid 1B \mid 1 \}$

S is the start symbol.

Eliminating unit productions

A unit production is a prodⁿ of the form $\boxed{A \to B}$, where both A & B are variables.

Eliminate all unit prodⁿˢ from the grammar

$S \to AB$

$A \to a$

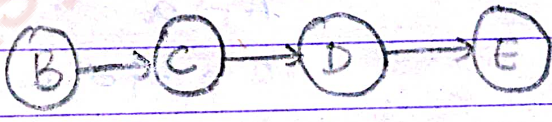
$B \to C \mid b$

$C \to D$

$D \to E \mid bC$

$E \to d \mid Ab$

Non unit prodⁿˢ	unit prodⁿˢ
$S \to AB$	$B \to C$
$A \to a$	$C \to D$
$B \to b$	$D \to E$
$D \to bC$	
$E \to d \mid Ab$	



It is clear from the dependency graph that $D \to E$, so the non-unit prodⁿˢ generated from 'E' can also be generated from 'D'.

$E \to d \mid Ab$

Since $D \overset{*}{\Rightarrow} E$

$D \to d \mid Ab$

The resulting D prodⁿˢ are

$D \to d \mid Ab \mid bC$

Since $C \overset{*}{\Rightarrow} D \overset{*}{\Rightarrow} E$

$C \rightarrow d \mid Ab \mid bC$

Since $B \overset{*}{\Rightarrow} C$, $C \overset{*}{\Rightarrow} D$, $D \overset{*}{\Rightarrow} E$

$B \rightarrow b \mid d \mid Ab \mid bC$

The final grammar obtained after eliminating unit productions is shown below:

$V = \{S, A, B, C, D, E\}$
$T = \{a, b, d\}$

$P = \{ \; S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b \mid d \mid Ab \mid bC$
$C \rightarrow bC \mid d \mid Ab$
$D \rightarrow bC \mid d \mid Ab$
$E \rightarrow d \mid Ab$
$\}$

S is the start symbol.

Eg Eliminate unit productions from the grammar

$S \rightarrow A0 \mid B$
$B \rightarrow A \mid 11$
$A \rightarrow 0 \mid 12 \mid B$

Non-Unit prod^n

$S \to A0$

$B \to 11$

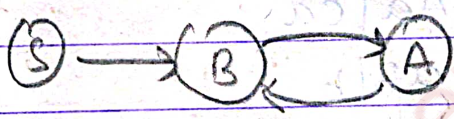
$A \to 0 \mid 12$

Unit prod^n

$S \to B$

$B \to A$

$A \to B$



It is clear from the dependence graph that $S \overset{*}{\Rightarrow} B$, $S \overset{*}{\Rightarrow} A$, $B \overset{*}{\Rightarrow} A$ & $A \overset{*}{\Rightarrow} B$ so the new prod^n for S, A & B are

$S \to A0 \mid 11 \mid 0 \mid 12$

$B \to 0 \mid 12 \mid 11$

$A \to 0 \mid 12 \mid 11$

The resulting grammar without unit prod^{n}

$G' = (V', T', P', S)$

$V' = \{S, A, B\}$

$T' = \{0, 1, 2\}$

$P' = \{ S \to A0 \mid 11 \mid 0 \mid 12,$

$B \to 0 \mid 12 \mid 11$

$A \to 0 \mid 12 \mid 11 \}$

S is the start symbol.

Eg Eliminate unit prod^ns from the grammar

$S \to Aa \mid B \mid Ca$

$B \to aB \mid b$

$C \to Db \mid D$

$D \to E \mid d$

$E \to ab$

Sol^n. Non-unit prod^ns.

$S \to Aa \mid Ca$

$B \to aB \mid b$

$C \to Db$

$D \to d$

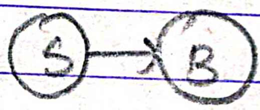
$E \to ab$

Unit prod^ns

$S \to B$

$C \to D$

$D \to E$



$S \overset{*}{\Rightarrow} B$

$S \to Aa \mid aB \mid b \mid Ca$

~~$C \overset{*}{\Rightarrow} D$~~ $D \overset{*}{\Rightarrow} E$

$D \to ab \mid d$

$C \stackrel{1}{\Rightarrow} D \quad D \stackrel{*}{\Rightarrow} E$

$C \rightarrow Db \mid ab \mid d$

The resulting grammar

$G' = (V', T', P', S)$

$V' = \{S, A, B, C, D, E\}$

$T' = \{a, b, d\}$

$P' = \{ \ S \rightarrow Aa \mid aB \mid b \mid ca$

$B \rightarrow aB \mid b$

$C \rightarrow Db \mid ab \mid d$

$D \rightarrow d \mid ab$

$E \rightarrow ab \ \}$

S is the start symbol.

Chomsky Normal form.

Every nonempty CFL without ε has a grammar G in which all productions are in one of two forms, either

1) $A \to BC$, where A, B & C are variable
2) $A \to a$, where A is a variable & a is a terminal

Further, G has no useless symbols. Such a grammar is said to be in Chomsky Normal Form, or CNF.

Eg) Convert the following grammar to CNF

① remove ε-prod

② remove unit-p

③ remove useless

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

Soln

Std form

$A \to 1$

$B \to 0$

$S \to 0A \mid 1B$

$S \to B_0A \mid B_1B$, $A \to B_0AA \mid B_1S$, $B \to B_1BB \mid B_0S$

$B_0 \to 0$ $\quad$ $B_0 \to 0$ $\quad$ $B_0 \to 0$

$B_1 \to 1$ $\quad$ $B_1 \to 1$ $\quad$ $B_1 \to 1$

Grammar obtained

$V_1 = \{S, A, B, B_0, B_1\}$

$T_1 = \{0, 1\}$

$P_1 = \{ S \to B_0A \mid B_1B,\ A \to B_0AA \mid B_1S \mid 1,\ B \to B_1BB \mid B_0S$

$B_0 \to 0$

$B_1 \to 1 \}$

S is the start symbol.

$S \rightarrow B_0 A \mid B_1 B$

$A \rightarrow B_0 AA \mid B_1 S \mid 1$

$B \rightarrow B_1 BB \mid B_0 S \mid 0$

$B_0 \rightarrow 0$

$B_1 \rightarrow 1$

CNF prodⁿ are

$S \rightarrow B_0 A \mid B_1 S$

$A \rightarrow B_1 S \mid 1$

$B \rightarrow B_0 S \mid 0$

$B_0 \rightarrow 0$

$B_1 \rightarrow 1$

Productions which are not in CNF form.

$A \rightarrow B_0 AA$

$B \rightarrow B_1 BB$

$A \rightarrow B_0 AA$, $A \rightarrow B_0 D_1$

$D_1 = AA$

$B \rightarrow B_1 BB$

$B \rightarrow B_1 D_2$

$D_2 \Rightarrow BB$

$G_1 = \{S, A, B, B_0, B_1, D_1, D_2\}$

$T_1 = \{0, 1\}$

$P_1 = \{ S \rightarrow B_0 A \mid B_1 S,\ A \rightarrow B_0 D_1 \mid B_1 S \mid 1,\ B \rightarrow B_1 D_2 \mid B_0 S \mid 0$

$B_0 \rightarrow 0$, $D_1 \Rightarrow \cancel{A} A$

$B_1 \rightarrow 0$ $D_2 \rightarrow BB \}$

S is the start symbol

(Eg) Convert the following grammar to CNF

$I \rightarrow c \mid d \mid Ia \mid Ib \mid I0 \mid I1$
$F \rightarrow I \mid (E)$
$T \rightarrow F \mid T*F$
$E \rightarrow T \mid E+T$

(Eg) Find a grammar equivalent to

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

with no useless symbols

(Eg) Begin with the grammar

$S \rightarrow ASB \mid \epsilon$
$A \rightarrow aAS \mid a$
$B \rightarrow SbS \mid A \mid bb$

(a) Are there any useless symbols? Eliminate them if so.
(b) Eliminate ϵ-productions
(c) Eliminate unit-prodn
(d) put the grammar into CNF.

Greibach Normal Form (GNF).

In GNF there is no restriction on the number of symbols on the right hand side, but there is restriction on the terminals & variables appear on the right hand side of the prodn.

Defn: Let $G=(V,T,P,S)$ be a CFG. The CFG G is said to be in GNF if all the productions are of the form

$$\boxed{A \rightarrow a\alpha}$$

where $a \in T$ and $\alpha \in V^*$, i.e the first symbol on the right hand side of the prodn must be a terminal & it can be followed by zero or more variables.

Convert the following grammar to GNF.

$S \rightarrow AB1 \mid 0$

$A \rightarrow 00A \mid 1A1$

$B \rightarrow 1A1$

$S \rightarrow ABA_1 \mid 0$

$A \rightarrow A_0A_0A \mid A_1AA_1$

$B \rightarrow A_1AA_1$

$A_1 \rightarrow 1$

$A_0 \rightarrow 0$

Now restrict the no of variables on the right hand side of the prodⁿ to two & the resulting grammar in CNF notation is

$S \rightarrow AD_1 \mid 0$

$A \rightarrow A_0 D_2 \mid A_1 D_3$

$B \rightarrow A_1 D_3$

$A_1 \rightarrow 1$

$A_0 \rightarrow 0$

$D_1 \rightarrow BA_1$

$D_2 \rightarrow A_0 A$.

$D_3 \rightarrow AA_1$

Now let us rename all the variables as shown below:

Let $S = A_1$, $A = A_2$, $B = A_3$, $A_0 = A_4$, $A_1 = A_5$,

$D_1 = A_6$, $D_2 = A_7$, $D_3 = A_8$

Now the grammar can be rewritten as:

$A_1 \rightarrow A_2 A_6 \mid 0$

$A_2 \rightarrow A_4 A_7 \mid A_5 A_8$

$A_3 \rightarrow A_5 A_8$

$A_5 \rightarrow 1$

$A_4 \rightarrow 0$

$A_6 \rightarrow A_3 A_5$

$A_7 \rightarrow A_4 A_2$

$A_8 \rightarrow A_2 A_5$

In the above prod^{n}s, note that A_4 & A_5-prod^{n}s are in GNF

Consider A_3 prod^n : Substituting for A_5 in A_3-prod^n we get

$$A_3 \rightarrow A_5 A_8 = 1 A_8 \longrightarrow \text{GNF}.$$

Consider A_2 prod^n

$$A_2 \rightarrow A_4 A_7 \mid A_5 A_8 = 0 A_7 \mid 1 A_8 \longrightarrow \text{GNF}.$$

Consider A_1 prod^n.

$$A_1 \rightarrow A_2 A_6 \mid 0 = (0A_7 \mid 1A_8) A_6 \mid 0 = 0A_7A_6 \mid 1A_8A_6 \mid 0 \rightarrow \text{GNF}$$

Consider A_6 prod^n.

$$A_6 \rightarrow A_3 A_5 = \cancel{1A_8}\ (A_5 A_8) A_5$$

$$A_6 \rightarrow A_5 A_8 A_5 = 1 A_8 A_5 \longrightarrow \text{GNF}$$

Consider A_7 prod^n

$$A_7 \rightarrow A_4 A_2 = 0 A_2 \rightarrow \text{GNF}.$$

Consider A_8 prod^n

$$A_8 \rightarrow A_2 A_5 = (0A_7 \mid 1A_8) A_5$$
$$= 0A_7A_5 \mid 1A_8A_5 \longrightarrow \text{GNF}$$

$G = (V, T, P, S)$

$V = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\}$

$T = \{0, 1\}$

$P = \{$

$A_1 \rightarrow 0A_7A_6 \mid 1A_8A_6 \mid 0$

$A_2 \rightarrow 0A_7 \mid 1A_8$

$A_3 \rightarrow 1A_8$

$A_4 \rightarrow 0$

$A_5 \rightarrow 1$

$A_6 \rightarrow 1A_8A_5$

$A_7 \rightarrow 0A_2$

$A_8 \rightarrow 0A_7A_5 \mid 1A_8A_5$

$\}$

S is start symbol.