Lucky Page No.:
Date: / /

Unit 3 : Context free Grammers

* Context free Grammer :

Context free Grammar is a formal grammer which is used to generate all possible strings in a given formal language.

Grammar is set of rules. It check correctness of sentence or syntax.

FA and Grammar are classified by chomsky.

Context free Grammar G is 4 tuple.

$G = (V, \Sigma, S, P)$

Where,

$V \Rightarrow$ variable, Non Terminal symbol

$\Sigma \Rightarrow$ Terminal symbol

$S \Rightarrow$ start symbol

$P \Rightarrow$ Production in grammar.

Draw the regular language of the following

$L = \{0,1\}^* \{10\}$

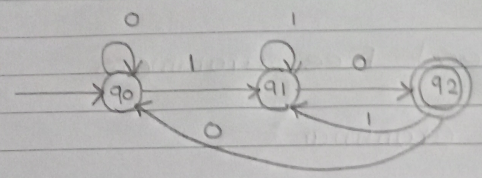


fig: DFA for $\{1.0\}^* \{10\}$

consider the language which is accepted by given DFA i.e.

W = 1100010

Initial string	state
$\wedge$	q_0
1	q_1
11	q_1
110	q_2
1100	q_0
11000	q_0
110001	q_1
1100010	q_2

Now convert given table into CFG form

$q_0 \rightarrow 1q_1$ / $q_0 \xrightarrow{1} q_1$

$q_1 \rightarrow 1q_1$ / $q_1 \xrightarrow{1} q_1$

$q_1 \rightarrow 0q_2$ / $q_1 \xrightarrow{0} q_2$

so it having shorthand notation

$q_0 \longrightarrow a.q_1$

where $a \in \Sigma$ and q_0 and q_1 are non-terminal symbol

$q_0, q_1 \in V$

When last production is their then we can take $q_0 \rightarrow b$ where $q_0 \in V$ and $b \in \Sigma$.

Where V = non terminal symbol

Σ = Terminal symbol

Lucky Page No.:
Date: / /

* Union, Concatenation and kleene on CFG

① Union :

If L_1 and L_2 are two context free language which having,

$G_1 = (V_1, \Sigma, S_1, P_1)$

$G_2 = (V_2, \Sigma, S_2, P_2)$

union of G_1 and G_2 will be noted as,

$\boxed{G_u = G_1 \cup G_2}$

$\boxed{L(G_u) = L_1 \cup L_2}$

$G_u = (V_u, \Sigma, S_u, P_u)$

where, $\boxed{V_u = V_1 \cup V_2 \cup \{S_u\}}$

$S_u = S_0$ starting symbol in new union CFL

$\boxed{P_0 = P_1 \cup P_2 \cup \{S_0 \rightarrow S_1 \mid S_2\}}$

Lucky Page No.:
Date : / /

② Concatenation :

We have G_1 and G_2 as two CFG

$G_c = (V_c, \Sigma, S_c, P_c)$

$$\boxed{V_c = V_1 \cup V_2 \cup \{S_c\}}$$

S_c = start symbol in G_c

$$\boxed{P_c = P_1 \cup P_2 \cup \{S_c \rightarrow S_1 S_2\}}$$

③ kleene :

$G_1 = (V_1, \Sigma, S_1, P_1)$

then

$G_K = (V_K, \Sigma, S_K, P_K)$

where,

$$\boxed{V_K = V_1 \cup \{S_K\}}$$

S_K = new starting symbol

$$\boxed{P_K = P_1 \cup \{S_K \rightarrow S_1 S_K \mid \Lambda\}}$$

Lucky Page No.:
Date: / /

* Union, concatenation and kleen's of CFL's

If L_1 and L_2 are CFL then $L_1 \cup L_2$, $L_1 \cdot L_2$, L_1^* also CFL

By contractive method,

consider $G_1 = \{V_1, \Sigma, S_1, P_1\}$ and

$G_2 = \{V_2, \Sigma, S_2, P_2\}$

G_1 and G_2 are generating L_1 and L_2 respectively.

① Union :

$G_u = \{V_u, \Sigma, S_u, P_u\}$ generating $L_1 \cup L_2$

If necessary we rename the elements of V_2 then

$V_1 \cap V_2 = \phi$

$V_u = V_1 \cup V_2 \cup \{S_u\}$

where, S_u = new start symbol not in V_1 & V_2

$P_u = P_1 \cup P_2 \cup \{S_u \rightarrow S_1 | S_2\}$

if x is in L_1 or L_2 then $S_u \overset{*}{\Longrightarrow} x$ is in grammar G_u

because we start the derivation in

$S_u \to S_1$ or $S_u \to S_2$ and continue the derivation of x in G_1 or G_2

$\therefore L_1 \cup L_2 \subseteq L(G_u)$

If x is derivable from S_u in G_u the first step in any derivation must be

$S_u \Rightarrow S_1$ or $S_u \Rightarrow S_2$

In the first case all subsequent production are must be production in because no variable in V_2 are involved and thus $x \in L_1$ and second case $x \in L_2$

$$\boxed{L(G_u) \subseteq L_1 \cup L_2}$$

② Concatenation :

A Grammar $G_c = (V_c, \Sigma, S_c, P_c)$ generating $L_1 \cdot L_2$

where $V_1 \cap V_2 = \phi$

$V_c = V_1 \cup V_2 \cup \{S_c\}$

Lucky Page No.:
Date: / /

$P_c = P_1 \cup P_2 \cup \{ S_c \rightarrow S_1 . S_2 \}$

if $x \in L_1 . L_2$ then $x = x_1 x_2$ where $x_i \in L_i$ for i
we may then derive x in G_c as follows.

$S_c \Rightarrow S_1 S_2 \overset{*}{\Rightarrow} x_1 S_2 \overset{*}{\Rightarrow} x_1 x_2 = x$

x_1 in G_1 and
x_2 in G_2

if x is derived from S_c then $S_c \Rightarrow S_1 . S_2$ x
must be derived from $x = x_1 x_2$

where for each i, x_i can be derived from S_i in G_c

$\because V_1 \cap L_2 = \phi$, being derivable from S_i in G_c
means being derivable from S_i in G_i and so

$$\boxed{x \in L_1 L_2}$$

③ kleene :

A Grammar $G^* = \{ V, \Sigma, S, P \}$ generating L_1^*

$V = \{ V_1 \cup \{ S_{*1} \} \}$

where, $S_1 \notin V_1$

Lucky Page No.:
Date: / /

L_1^* contains string of the form

$x = x_1, x_2, \ldots x_k$

where $x_1 \in L_1$

Each x_i is derived from S_1, then to derive x from S. it is enough to be able to derive a string of k S_1's

$S \rightarrow S_1 S \mid \Lambda$ in P

$P = P_1 \cup \{S \rightarrow S_1 S \mid \Lambda\}$

if $x \in L(G^*)$ then on other hand $x = \Lambda$ or x can be derived from some string of the form S_1^k in G^*, second case only production in G^* beginning with S_1 are those G_1 we conclude

$$\boxed{x \in L(G_1)^k \subseteq L(G_1)^*}$$

Lucky Page No.:
Date: / /

* Derivation tree and ambiguity :

Like any language english or other there is some fixed strcture like that in CFG it contain also fixed pattern called Derivation tree or Parse tree.

Rules :

① Root node in Parse/Derivation tree is always tabled with start symbol

② All the terminal symbols are labeled with some terminal symbols of the grammar. So nodes are called terminal or leaf node

③ If string contain n symbols then it arranged from left to right

④ Non terminal symbols labbled with nonterminal string.

⑤ If the string contain $\{\wedge\}$, ϵ, λ empty symbol then it will be ignored.

Ex : ① If the Grammar $G = \{V, \Sigma, S, P\}$ in such a way that

① $V = \{S, A, B\}$

② $T / \Sigma = \{a, b\}$

③ $P \Rightarrow \{ S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b \}$

④ $S \Rightarrow$ starting symbol

Derivation tree

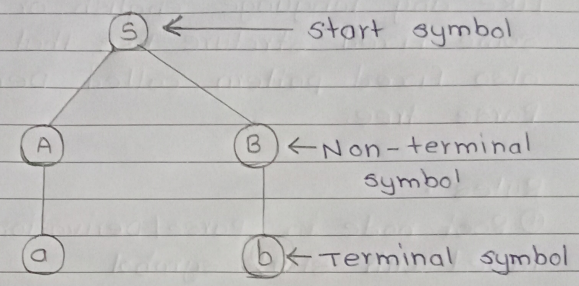


fig: Derivation tree for string ab

Ex ②: Draw the derivation tree for string aabbaa using following grammar:

① $G = (V_n, \Sigma, P, S)$

② $V_n = \{S, A\}$, $\Sigma = \{a, b\}$

③ $P = \{$

$S \rightarrow aAS$

$A \rightarrow SbA$

$S \rightarrow a$

$A \rightarrow ba$

$\}$

④ $S \Rightarrow$ starting symbol

Lucky Page No.:
Date : / /

Now, S is start symbol so any string genera-
-ted will stored at S

$S \rightarrow aAS$

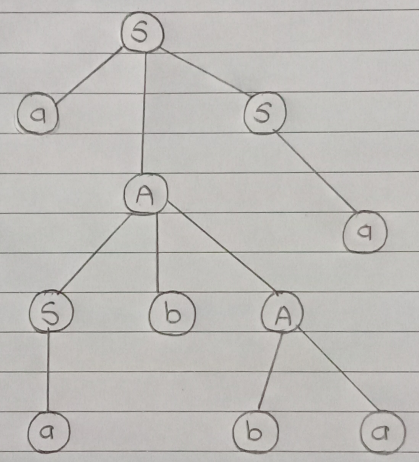


fig: Derivation tree

Reading from left the string is aabbaa hence
Ans.

Lucky Page No.:
Date: / /

* Ambiguity / Ambigous Grammar :

In case of CFG they called as Ambigous if,

① It contain terminal string with more than one parse tree.

② More than one leftmost derivation and rightmost derivation.

Ex : ① If $G = (V, \Sigma, P, S)$ be a grammar with

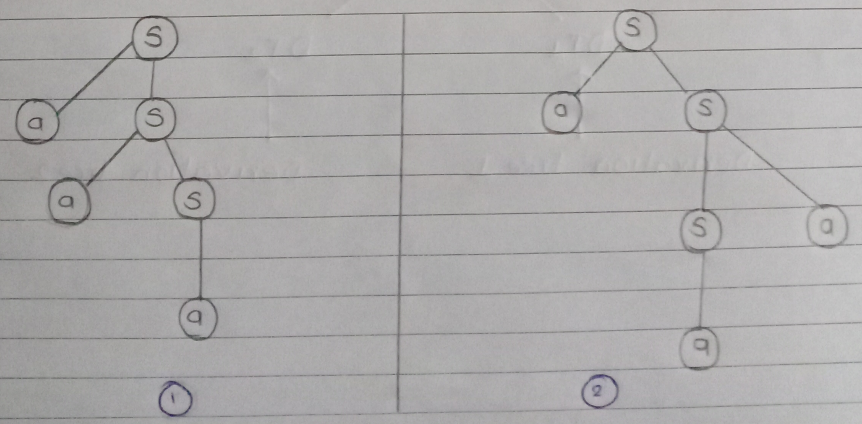
$G = (\{S\}, \{a\}, P, S)$

$V = \{S\}$

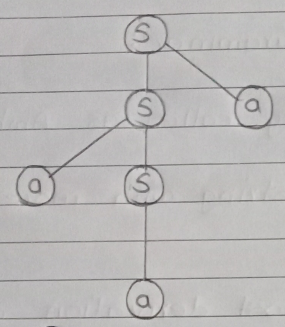
$\Sigma / T = \{a\}$

$P = \{S \rightarrow aS \mid SaS\}$ find out

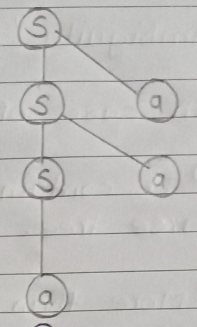
Parse tree for aaa



① ②



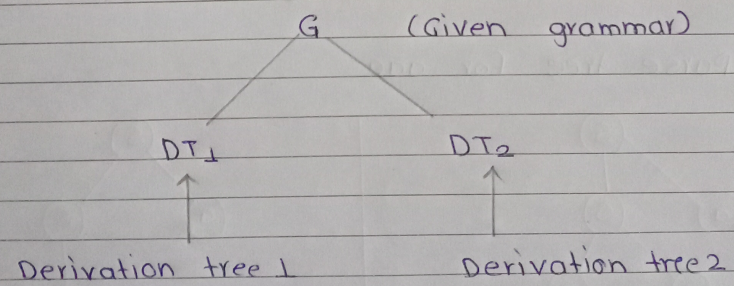
③



④

by above 4 way we can represent the aaa so it is ambigous grammar.

If given grammar having more than one derivation tree then it is called as an Ambigous grammar.



Lucky Page No.:
Date: / /

Ex ②: The following grammar:

$S \rightarrow A$
$S \rightarrow bSa$
$A \rightarrow \epsilon$
$A \rightarrow AcA$

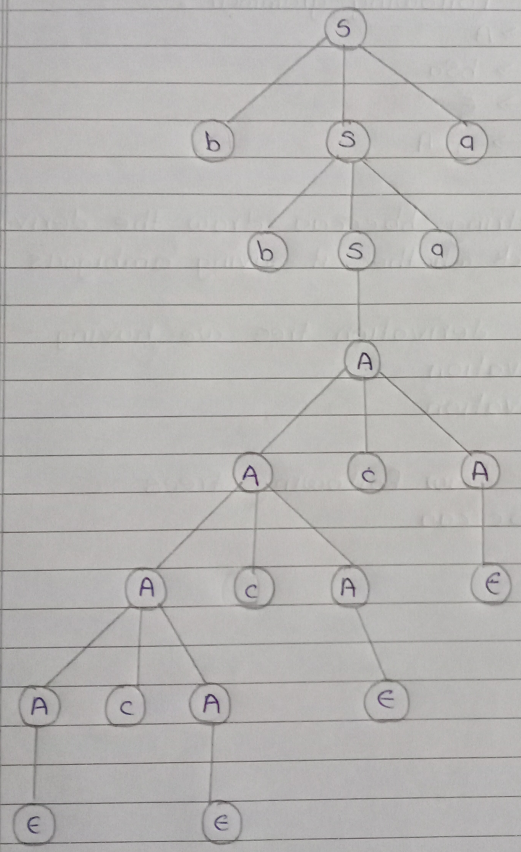
It accepts string bbcccaa draw the derivation tree and check whether it having ambigous or not.

→ Now to find derivation tree we having-
① left derivation
② Right derivation

Now we can draw following trees
String ⇒ bbcccaa

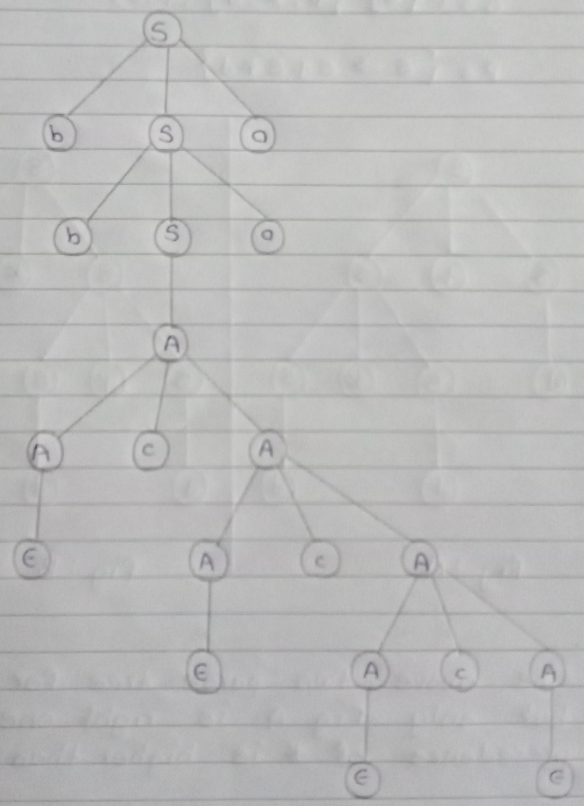
$S \rightarrow A$
$S \rightarrow bSa$
$A \rightarrow e$
$A \rightarrow AcA$

Tree no 1:



bbcccaa string accepting.

Tree No 2 :



bbccc aa string accepting

Lucky Page No.: Date: / /

Ex ③: If $G = (V, \Sigma, S, P\}$ be with grammar

$G = (\{S\}\ \{+, *, d\}, P, S)$

$P = \{S \rightarrow S+S*d\}$

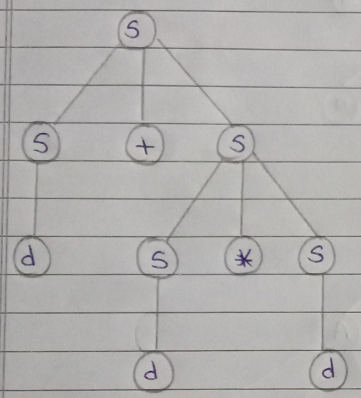


fig : A

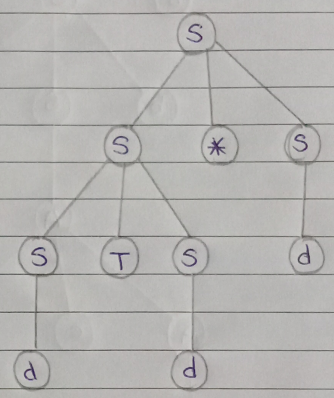


fig : B

The above two parse tree for d+d*d but only fig A is right one because presedence of * is higher than +

Lucky | Page No.:
Date: / /

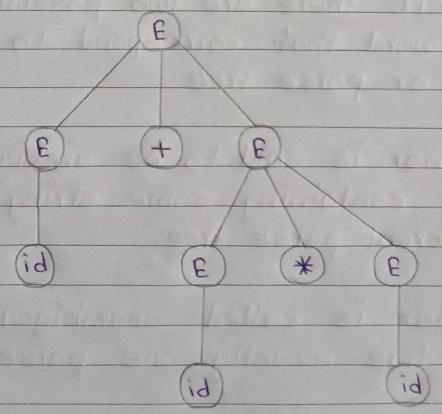
$V = \{E\}$
$\Sigma = \{id\}$
$S = E$
$P \Rightarrow E \rightarrow E+E$
$E \rightarrow E*E$
$E \rightarrow id$

By using above 3 productions we can construct

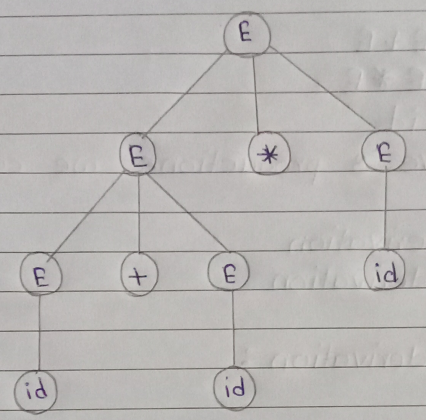
① Leftmost derivation
② Rightmost derivation

① Leftmost derivation :

$E \rightarrow E+E$



② Rightmost derivation :



A CFG is ambigous if for atleast one word in its CFL there are two possible derivation of the word that correspond to two different syntax or parse tree

A grammar is said to be ambigous if its language containing some string that has two different parse tree.

A grammar is called unambigous if its language has exactly one parse tree

Lucky Page No.:
Date: / /

* Simplification of CFG :

Simplification of CFG :

① Eliminating useless symbol

② Eliminating unit productions

③ Eliminate $\wedge$ production / ϵ

* Useful and useless symbol :

Useful	Useless
① Derives a string of terminals	Does not derive a string of terminals
② It is used in derivation of atleast one ω in L (CFG)	It does not occur in derivation sequence of any ω in L (CFG)

Ex : find useful and useless symbol in

$G = (V, \Sigma, S, P)$

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

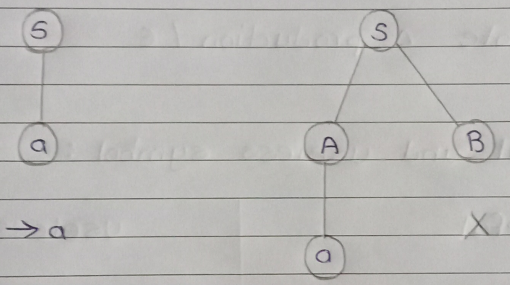
$T / \Sigma = \{a\}$

$P \Rightarrow \{ S \rightarrow a \mid AB$ and

$A \rightarrow a \}$

The starting symbol S can be terminate at ① $S \to a$

② If we take ② $S \to AB$
Hence A & B are not useful are useless symbol



① $S \to a$

useful symbol : S
useless symbol : A, B ② $S \to AB$
final production will $S \to a$

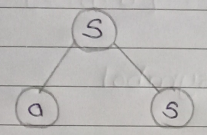
Ex ② $G = (V, T, P, S)$
$V = \{S, A, B, C\}$
$T/\Sigma = \{a, b\}$
S = starting symbol

$P : \Rightarrow$ $S \to aS \mid A \mid C$
$A \to a$
$B \to aa$
$C \to acb$

As S is starting symbol. The starting symbol is always useful.

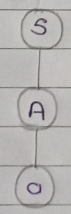
Now check reachability from start state

① $S \rightarrow aS$



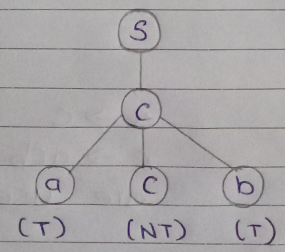
$S \rightarrow aS$ is useful

② $A \rightarrow a$



$A \rightarrow a$ is useful

③ $S \rightarrow C$
$C \rightarrow acb$
$B \rightarrow aa$



Not useful
$S \rightarrow C$
$C \rightarrow acb$
$B \rightarrow aa$

Ex ③ : P ⇒ $S \to aS$

$S \to A$

$S \to C$

$A \to a$

$B \to aa$

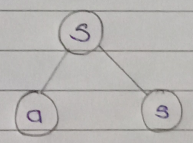
$C \to aCb$

S = starting symbol

$V = \{A, B, C, S\}$ $T / \Sigma = \{a, b, s\}$

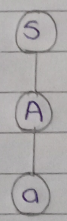
→ useful and useless symbol

① starting symbol always useful



$S \to aS$ is useful

② $S \to A$

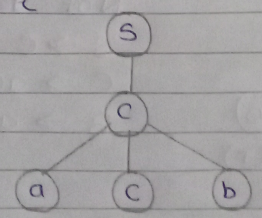


A is useful

$S \to A$

$A \to a$

③ $S \to C$



$S \to C$

$C \to aCb$

are useless

$B \to aa$ is useless

Lucky Page No.:
Date: / /

Ex: $G = (V, \Sigma, S, P)$

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

$T = \{a, b\}$

$P \Rightarrow S \rightarrow AB \mid CA$

$B \rightarrow BC \mid AB$

$A \rightarrow a$

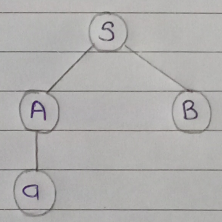
$C \rightarrow aB \mid b$

S = starting symbol

S is starting symbol. The starting symbol is useful variable.

① $S \rightarrow AB$

$B \rightarrow AB$



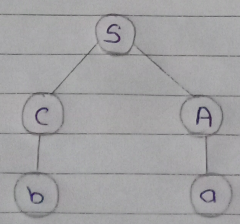
$S \rightarrow AB$

$B \rightarrow AB$ are useless

② $S \rightarrow CA$

$A \rightarrow a$

$C \rightarrow b$



$S \rightarrow CA$

$A \rightarrow a$

$C \rightarrow b$ are useful

Lucky Page No.:
Date: / /

* Elimination of Λ productions :

The partation of the form $A \to \Lambda$ is called null production. It may also represent like,

$$A \to \Lambda$$
$$A \to \epsilon$$

Ex :

$G = (V, \Sigma, S, P)$
$V = \{S, A, B, C, D\}$
$\Sigma = \{a, b, \Lambda\}$
S = starting symbol
Find nullable variable :

① $S \to ABCBCDA$
$A \to CD$
$B \to Cb$
$C \to a | \Lambda$
$D \to bb | \Lambda$

$\to$ ① $C \to \Lambda$ and $D \to \Lambda$ so C and D are nullable variable

② $S \to ABCBCDA$
$\to AB\Lambda B\Lambda\Lambda A$
$\to ABBA$
$\to CDcbcbCD$
$\to \Lambda\Lambda\Lambda b\Lambda b\Lambda\Lambda$
$S \to bb$
S is not nullable

Lucky Page No.:
Date : / /

③ $A \to CD$
$A \to \Lambda\Lambda$
$A \to \Lambda$
A is nullable

④ $B \to cb$
$B \to \Lambda b$
$B \to b$
B is not nullable

Finally
Nullable variable = A, C, D
Not nullable variable = B, S

Ex :
$S \to ACB \mid cbB \mid Ba$
$A \to da \mid BC$
$B \to gC \mid \Lambda$
$C \to ha \mid \epsilon$

Find : ① Remove null production
② Find nullable variable

→ Null productions are $B \to \Lambda$
$C \to \Lambda$

$S \to ACB \mid cbB \mid Ba$

Lucky Page No.:
Date: / /

$S \rightarrow ACB \mid AC \mid AB \mid A$
$S \rightarrow cbB \mid cb$
$S \rightarrow Ba \mid a$

② $A \rightarrow da \mid BC$
$A \rightarrow da$
$A \rightarrow Bc \mid B \mid C$

③ $B \rightarrow gc \mid \Lambda$
$B \rightarrow gc \mid g$

④ $C \rightarrow ha$

Nullable variables.

$V = \{S, A, B, C\}$
$B \rightarrow \Lambda$ and $C \rightarrow \Lambda$
B and C are nullable

$S \rightarrow ACB \mid cbB \mid Ba$
$S \rightarrow A\Lambda\Lambda \mid cb\Lambda \mid \Lambda a$
$S \rightarrow A \mid cb \mid a$
S is not nullable

$C \rightarrow ha \mid \epsilon$
C is nullable

Lucky Page No.:
Date : / /

$A \rightarrow da \mid BC$
$A \rightarrow da \mid nn$
$A \rightarrow da$

A is not nullable

Nullable are : B and C

Not nullable are : A and S

Lucky Page No.:
Date: / /

* Remove Null / Epsilon Product

Ex : $S \rightarrow AB$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid \epsilon$

Step I : Now we have epsilon symbol

$B \rightarrow \epsilon$
Initially
$S \rightarrow AB$
$A \rightarrow aA \mid a$
$B \rightarrow bB$

Step II : Now replace $B \rightarrow \epsilon$ whenever applicable at right side

$S \rightarrow A\epsilon$
$A \rightarrow aA \mid a$
$B \rightarrow b\epsilon$

i.e $S \rightarrow A$
$A \rightarrow aA \mid a$
$B \rightarrow b$

Hence final result will be

$S \rightarrow AB \mid A$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid b$

Lucky Page No.:
Date: / /

* Removing Unit Production:

If $G = (V, \Sigma, S, P)$ is CFG and we have a production such a way that

$$V_1 \rightarrow V_2$$

Where V_1 and V_2 are non terminal symbols then $V_1 \rightarrow V_2$ called unit production.

$$V_1, V_2 \in V$$

Ex: ①

①	$S \rightarrow A$	Unit production
②	$S \rightarrow a$	Not unit production
③	$S \rightarrow \wedge \mid \epsilon$	Null production
④	$S \rightarrow Aa$	Not unit production

Ex ②:

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow c \mid b$
$C \rightarrow D$
$D \rightarrow E$
$E \rightarrow a$

Identify unit and non unit production

unit production :

1 N.T → 1 N.T symbol symbol

unit production :

B → C
C → D
D → E

Non unit production :

S → AB
A → a
B → b
E → a

Ex ③ : S → ABC / Λ
B → a/b
C → D
E → aB
F → AB

unit production	Null production	non unit production
C → D	S → Λ	S → ABC B → a/b E → aB F → AB

Lucky Page No.:
Date: / /

* Removing Unit production :

Production of the form $A \to B$, $B \to S$ are called unit production

Ex :

$S \to A \mid bb$ — ①
$A \to B \mid b$ — ②
$B \to S \mid a$ — ③

from ① and ②

$B \to bb \mid a$ — ④

from ② and ④

$A \to bb \mid a \mid b$ — ⑤

from ① and ② and ⑤

$S \to b \mid bb$

The final result will be

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

Lucky Page No.:
Date: / /

* Eliminating useless variable from a Context free grammar:

A context free grammar is $\{V, \Sigma, S, P\}$ combines some Nonterminal (V) and terminal (Σ / T) symbol

If the given Nonterminal symbol is not in use called as useless symbol or variable

Now consider following grammar

$G = \{V, \Sigma, S, P\}$
$V = \{A, B, S\}$
$\Sigma = \{a, b\}$

$S \rightarrow AA \mid 1A0 \mid B \mid 0A1 \mid 01$ —— ①

A and B are nonterminals and 0 and 1 are terminal symbol.

$S \rightarrow AA \mid 1A0 \mid 0A1 \mid 01$ —— ②

B is useless symbol

both ① and ② are identical

Remove those string whose string contain least one non used string.

Ex : $G = \{V, T, S, P\}$

$V = \{S, C, D\}$

$T = \{d\}$

$P \Rightarrow S \rightarrow CD \mid d$

$D \rightarrow d$

→ From above production we can say C is useless

So, we can remove C

$S \rightarrow d$

$D \rightarrow d$

$G = \{\{S\}, \{d\}, \{S\}, \{P\}\}$

Now we observe $S \rightarrow d$ doesn't contain any nonterminal symbol D in it so it become,

$S \rightarrow d$ only by removing $D \rightarrow d$

Ex ② : $S \rightarrow a \mid aA \mid B \mid C$

$A \rightarrow aB \mid \epsilon$

$B \rightarrow Aa$

$C \rightarrow cCD$

$D \rightarrow fff$

find : ① unit production ④ ∧ Production

② useful variable ⑤ Nullable variable

③ useless variable

Lucky Page No.:
Date: / /

① unit Production :

$G = (V, \Sigma, S, P)$ and we have production such a way that

$$\boxed{\perp NT \rightarrow \perp NT}$$

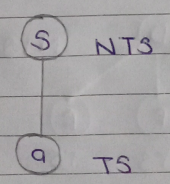
$$\boxed{NT \in V}$$

∴ unit production :- $S \rightarrow B \mid C$

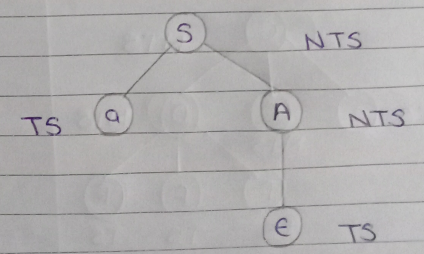
Non Unit production :- $S \rightarrow a \mid aA$
$A \rightarrow aB$
$B \rightarrow Aa$
$C \rightarrow cCD$
$D \rightarrow fff$

② useful and useless variable :

① $S \rightarrow a$

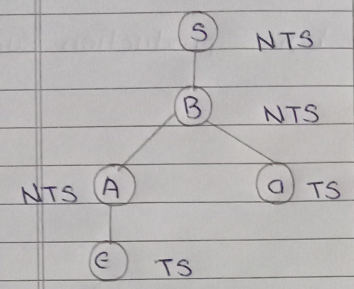


② $S \rightarrow aA$

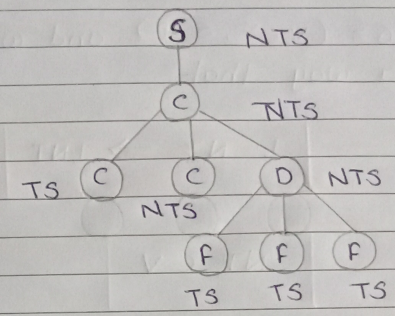


Lucky Page No.:
Date: / /

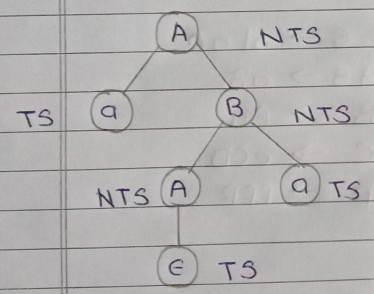
③ $S \rightarrow B$



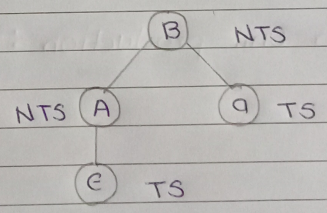
④ $S \rightarrow C$



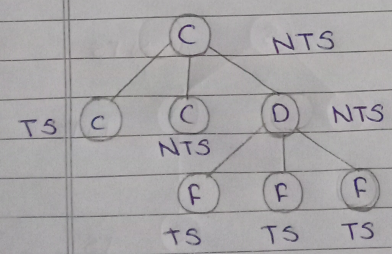
⑤ $A \rightarrow aB$



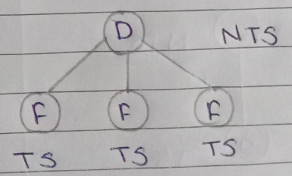
⑥ $B \rightarrow Aa$



⑦ $C \rightarrow cCD$

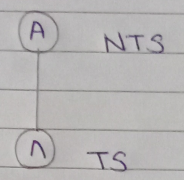


⑧ $D \rightarrow fff$



Lucky Page No.:
Date: / /

⑨ $A \rightarrow \epsilon \mid \wedge$



useful variable : S, A, B

useless variable : C, D

③ Null production :

$A \longrightarrow \epsilon$

N.T symbol — Null / Epsilion Symbol

In given production $A \rightarrow \epsilon$ is null production

④ Nullable variable :

$A \rightarrow \epsilon$ ∴ A is nullable

∴ Nullable variable : A

Not Nullable variable : S, B, C, D

Lucky | Page No.:
Date: / /

* Chomskey Normal form :

In a CFG the production is of the form

$A \rightarrow BC$ $\quad$ $A \rightarrow 2NT$

$A \rightarrow a$ $\quad$ $S \rightarrow TS$

i.e.,

Nonterminal $\rightarrow$ string of two non terminals

Nonterminal $\rightarrow$ Terminal symbol

then they called chomskey normal form

Ex: If two productions $A \rightarrow aAb$ and

$B \rightarrow ab$ and

replace $x_a \rightarrow a$, $x_b \rightarrow b$

$A \rightarrow x_a A x_b$

$B \rightarrow x_a x_b$

Also

$A \rightarrow BCDBGE$ would be replace by

$A \rightarrow BY_1$

$Y_1 \rightarrow CY_2$

$Y_2 \rightarrow DY_3$

$Y_3 \rightarrow BY_4$

$Y_4 \rightarrow CE$

Lucky Page No.:
Date: / /

Hence, Now variable Y_1, Y_2, Y_3, Y_4 are production. so it is CNF

If CFG $G = (V, \Sigma, S, P)$ be in CNF and $G' = (V', \Sigma, S, P')$ then

$$L(G') = L(G) - \{\Lambda\}$$

Ex ②:

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

① $S \rightarrow bA$

$S \rightarrow bA$ —— ①

Take $b = S_1$ in eq^n ①

b = terminal symbol
S_1 = Non terminal symbol

$\therefore S \rightarrow S_1 A$ CNF

② $S \rightarrow aB$

put $a = S_2$

$S \rightarrow S_2 B$ CNF

Lucky Page No.:
Date: / /

③ $A \rightarrow bAA$
$A \rightarrow S_1AA$ —②

put $S_3 = S_1A$ in ②

$\therefore A \rightarrow S_3A$ CNF

④ $A \rightarrow aS$
$A \rightarrow S_2S$ CNF

⑤ $A \rightarrow a$ already CNF

⑥ $B \rightarrow aBB$
$B \rightarrow S_2BB$

$S \rightarrow SB$ CNF

⑦ $B \rightarrow bS$
$B \rightarrow S_1S$ CNF

⑧ $B \rightarrow a$ already CNF

Required CNF

$V = \{S, A, B, S_1, S_2, S_3\}$

$T/\Sigma = \{a, b\}$

$P = \{ S \rightarrow S_1A \mid S_2B$ S = starting symbol
$A \rightarrow S_3A \mid S_2S \mid a$
$B \rightarrow SB \mid S_1S \mid a \}$

Lucky Page No.:
Date : / /

* Converting CFG to CNF

Let G be the grammar of the production

Ex: $S \rightarrow AACD$
$A \rightarrow aAb \mid \Lambda$
$C \rightarrow aC \mid a$
$D \rightarrow aDa \mid bDb \mid \Lambda$
convert into CNF

① Eliminating Λ production:

In above example nullable variable are A & D

$S \rightarrow AACD \mid ACD \mid AAC \mid CD \mid AC \mid C$

$A \rightarrow aAb \mid ab$

$C \rightarrow aC \mid a$

$D \rightarrow aDa \mid bDb \mid aa \mid bb$

② Eliminating unit production:

Now from production ① and ③ it is

$S \rightarrow aC \mid a$
where we delete $S \rightarrow C$

Lucky Page No.:
Date: / /

③ Restricting the right side of production to single terminals or string of two or more variable

$S \rightarrow AACD \mid ACD \mid AAC \mid CD \mid AC \mid X_aC \mid a$

$A \rightarrow X_aAX_b \mid X_aX_b$

$C \rightarrow X_aC \mid a$

$D \rightarrow X_aDX_a \mid X_bDX_b \mid X_aX_a \mid X_bX_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

④ Final step to CNF

$S \rightarrow AT_1 \qquad T_1 \rightarrow AT_2 \qquad T_2 \rightarrow CD$

$S \rightarrow AU_1 \qquad U_1 \rightarrow CD$

$S \rightarrow AV_1 \qquad V_1 \rightarrow AC$

$S \rightarrow CD \mid AC \mid X_aC \mid a$

$A \rightarrow X_a\omega_1 \qquad \omega_1 \rightarrow AX_b$

$A \rightarrow X_aX_b$

$C \rightarrow X_a C | a$

$D \rightarrow X_a Y_a \qquad Y_1 \rightarrow D X_a$

$D \rightarrow X_b Z_1 \qquad Z_1 \rightarrow D X_b$

$D \rightarrow X_a X_b | X_b X_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

Ex ②: $S \rightarrow ASA$
$S \rightarrow BSB$
$S \rightarrow AA$
$S \rightarrow BB$
$S \rightarrow a$
$S \rightarrow b$
$A \rightarrow a$
$B \rightarrow b$

① $S \rightarrow AX_1$
where $X_1 \rightarrow SA$

② $S \rightarrow BX_2$
where $X_2 \rightarrow SB$

Hence required CNF is

Lucky Page No.:
Date : / /

$S \to AX_1$
$S \to BX_2$
$S \to AA$
$S \to BB$
$S \to a$
$S \to b$
$A \to a$
$B \to b$

③ Ex :

$S \to ABC$
$A \to a|b$
$B \to Bb|bb$
$C \to aC|cc|ba$

$G = (V, \Sigma, S, P)$

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

$\Sigma = \{a, b\}$

S = starting symbol

→ CNF : 1NT → 2NT
1NT → 1TS

① Take $S \to ABC$ —①
put $S_1 \to AB$ in ①
$S \to S_1C$ CNF

Lucky Page No.:
Date: / /

② $A \rightarrow a|b$ CNF

③ $B \rightarrow Bb|bb$

take $A \rightarrow a|b$

$B \rightarrow BA|AA$ CNF

③ $C \rightarrow ac|cc|ba$ —— ③

where $A \rightarrow a|b$ in ①

$C \rightarrow Ac|cc|AA$ CNF

Non terminal symbols.

$V = \{S, S_1, A, B, C\}$