## Ambiguous Grammar

A Grammar is said to be Ambiguous if there exists two or more derivation tree for a string w( that means two or more left derivation trees)

Example:. $G = (\{S\}, \{a+b, +, *\}, P, S\}$ where P consists of

$$S \to S+S \mid S*S \mid a \mid b$$

The string $a+a*b$ can be generated as:

| | |
|---|---|
| $S \to S+S$ | $S \to S*S$ |
| $\to a+S$ | $\to S+S*S$ |
| $\to a+S*S$ | $\to a+S*S$ |
| $\to a+a*S$ | $\to a+a*S$ |
| $\to a+a*b$ | $\to a+a*b$ |

Thus, this Grammar is Ambiguous.

## Simplification of Context free Grammar

In CFG, sometimes all the production rules and symbols are not needed for the derivation of strings. Besides this, there may also be some NULL productions and UNIT productions. Elimination of these productions and symbols is called simplification of CFG.

Simplification consists of the following steps:-

1) Reduction of CFG 2) Removal of Unit Productions 3) Removal of Null Productions.

### Reduction of CFG

CFG are reduced in two phases

phase1:- Derivation of an equivalence grammar 'G', from the CFG, G, such that each variable ~~devices~~ derives some terminal string

Derivation Procedure:-

Step1:- Include all symbols $W_1$, that derives some terminal and initialize $i=1$

Step 2: Include symbols $W_{i+1}$, that derives $W_i$

Step3: Increment i and repeat step 2, until $W_{i+1} = W_i$

Step 4:- Include all production rules that have $W_i$ in it.

**Phase 2:-** Derivation of an equivalent grammar "G''", from the CFG, 'G'', such that each symbol appears in a sentential form.

**Derivation Procedures:-**

Step 1: Include the start symbol in $Y_1$ and initialize $i=1$

Step 2: Include all symbols $Y_{i+1}$, that can be derived from $Y_i$ and include all production rules that have been applied.

Step 3: Increment $i$ and repeat step 2, until $Y_{i+1} = Y_i$

---

**Example:-** Find a reduced grammar equivalent to the Grammar G, having production rules

P: $S \rightarrow AC \mid B$, $A \rightarrow a$, $C \rightarrow c \mid BC$, $E \rightarrow aA \mid e$

**Phase 1:**

$T = \{a, c, e\}$

$W_1 = \{A, C, E\}$ — set derive terminals

$W_2 = \{A, C, E, S\}$ — set of element of $W_1$

$W_3 = \{A, C, E, S\}$

$G' = \{(A, C, E, S), \{a, c, e\}, P, (S)\}$

P: $S \rightarrow AC$, $A \rightarrow a$, $C \rightarrow c$, $E \rightarrow aA \mid e$

**phase 2:-**

$Y_1 = \{S\}$

$Y_2 = \{S, A, C\}$

$Y_3 = \{S, A, C, a, c\}$

$Y_4 = \{S, A, C, a, c\}$

$G'' = \{(A, C, S), \{a, c\}, P, \{S\}\}$

P:- $S \rightarrow AC$, $A \rightarrow a$, $C \rightarrow c$

# Simplification of Context Free Grammar

## Removal of Unit Productions

Any production Rule of the form $A \to B$ where $A, B \in$ Non Terminals is called Unit Production.

**Procedure for Removal:-**

Step 1:- To remove $A \to B$, add production $A \to x$ to the grammar rule whenever $B \to x$ occurs in the grammar. [$x \in$ Terminal, $x$ can be Null]

Step 2:- Delete $A \to B$ from the grammar.

Step 3:- Repeat from step1 until all Unit Productions are removed

**Example:-** Remove Unit Productions from the Grammar whose production rule is given by

P: $S \to XY$, $X \to a$, $Y \to Z|b$, $Z \to M$, $M \to N$, $N \to a$

$Y \to Z$, $Z \to M$, $M \to N$

1) Since $N \to a$, we add $M \to a$

P: $S \to XY$, $x \to a$, $Y \to Z|b$, $Z \to M$, $M \to a$, $N \to a$

2) Since $M \to a$, we add $Z \to a$

P: $S \to XY$, $x \to a$, $Y \to Z|b$, $Z \to a$, $M \to a$, $N \to a$

3) Since $Z \to a$, we add $Y \to a$

P: $S \to xY$, $x \to a$, $Y \to a|b$, $Z \to a$, $M \to a$, $N \to a$

→ from Start Symbol S we can not reach to Z, M, N.

Remove the unreachable symbols

P: $S \to XY$, $x \to a$, $Y \to a|b$

# Simplification of Context Free Grammar

## Removal of Null Productions

In a CFG, a Non-Terminal Symbol 'A' is a nullable variable if there is a production $A \to \epsilon$ or there is a derivation that starts at 'A' and leads to $\epsilon$. (Like $A \to \ldots\ldots \to \epsilon$)

### Production for Removal:-

Step1:- To remove $A \to \epsilon$, look for all productions whose right side contains A

Step 2:- Replace each occurences of 'A' in each of these productions with $\epsilon$

Step 3:- Add the resultant productions to the Grammar.

Example :- Remove Null Productions from the following Grammar.

$S \to ABAC,\ A \to aA|\epsilon,\ B \to bB|\epsilon,\ C \to c$

$A \to \epsilon$ , $B \to \epsilon$

1) To eliminate $A \to \epsilon$

$S \to ABAC$

$S \to AB\epsilon C$

$\to ABC \mid BAC \mid BC$

$\to$

$A \to aA$

$A \to a$

New production: $S \to ABAC \mid ABC \mid BAC \mid BC$

$A \to aA \mid a$ , $B \to bB|\epsilon$ , $C \to c$

2) To eliminate $B \to \epsilon$

$S \to AAC \mid AC \mid C$ , $B \to b$

New production : $S \to ABAC \mid ABC \mid BAC \mid BC \mid AAC \mid AC \mid C$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to c$

# Normal Forms

① Chomsky Normal form
② Greibach Normal Form

## ① Chomsky Normal Form

In Chomsky Normal Form (CNF) we have a restriction on length of RHS; which is; elements in RHS should either be two variables or a Terminal.

A CFG is in Chomsky Normal Form if the productions are in the following forms:

$$A \rightarrow a$$

$$A \rightarrow BC$$

Where A, B and C are non-terminals and a is a terminal.

**Steps to convert a given CFG to Chomsky Normal Form:**

Step 1:- If the Start Symbol S occurs on some right side, create a new start Symbol S' and a new Production $S' \rightarrow S$.

Step 2:- Remove Null Productions. (Using the Null Production Removal discussed in previous Lecture).

Step 3:- Remove Unit Productions. ( —— " —— )

Step 4:- Replace each Production $A \rightarrow B_1 \ldots B_n$ where $n > 2$, with $A \rightarrow B_1 C$ where $C \rightarrow B_2 \ldots B_n$. Repeat this step for all Production having two or more symbols on the right side.

Step 5:- If the right side of any Production is in the form $A \rightarrow aB$ where 'a' is a terminal and A and B are non-terminals, then the Production is replaced by $A \rightarrow XB$ and $X \rightarrow a$.
Repeat this step for every Production Which is of the form $A \rightarrow aB$.

# Conversion of CFG to Chomsky Normal Form

Convert the following CFG to CNF : P: $S \to ASA \mid aB$, $A \to B \mid S$, $B \to b \mid \epsilon$

1) Since S appears in RHS, we add new state S' and $S' \to S$ is added to the production

P: $S' \to S$, $S \to ASA \mid aB$, $A \to B \mid S$, $B \to b \mid \epsilon$

---

2) Remove the Null Productions : $B \to \epsilon$ and $A \to \epsilon$ :

After removing $B \to \epsilon$ : P: $S' \to S$, $S \to ASA \mid aB \mid a$, $A \to B \mid S \mid \epsilon$, $B \to b$

After removing $A \to \epsilon$ : P: $S' \to S$, $S \to ASA \mid aB \mid a \mid AS \mid SA \mid S$,

$A \to B \mid S$, $B \to b$

---

3) Remove the Unit Productions : $S \to S$, $S' \to S$, $A \to B$, $A \to S$ :

After Removing $S \to S$ : P: $S' \to S$, $S \to ASA \mid aB \mid a \mid AS \mid SA$, $A \to B \mid S$, $B \to b$

---

After Removing $S' \to S$ : P: $S' \to ASA \mid aB \mid a \mid AS \mid SA$,

$S \to ASA \mid aB \mid a \mid AS \mid SA$,

$A \to B \mid S$, $B \to b$

---

After Removing $A \to B$ : P: $S' \to ASA \mid aB \mid a \mid AS \mid SA$,

$S \to ASA \mid aB \mid a \mid AS \mid SA$,

$A \to b \mid S$, $B \to b$

---

After Removing $A \to S$ : $S' \to ASA \mid aB \mid a \mid AS \mid SA$,

$S \to ASA \mid aB \mid a \mid AS \mid SA$,

$A \to b \mid ASA \mid aB \mid a \mid AS \mid SA$,

$B \to b$

---

4) Now find out the productions that has more than Two variables in RHS

$S' \to ASA$, $S \to ASA$ and $A \to ASA$

After removing these, we get : P: $S' \to AX \mid aB \mid a \mid AS \mid SA$,

$S \to AX \mid aB \mid a \mid AS \mid SA$,

$A \to b \mid AX \mid aB \mid a \mid AS \mid SA$,

$B \to b$,

$X \to SA$

5) Now change the productions $S' \to aB$, $S \to aB$ and $A \to aB$

Finally we get :

P : $S' \to AX \mid YB \mid a \mid AS \mid SA$, $( \ldots Y \to a)$

$S \to AX \mid YB \mid a \mid AS \mid SA$,

$A \to b \mid AX \mid YB \mid a \mid AS \mid SA$,

$B \to b$,

$X \to SA$,

$Y \to a$

Which is the required chomsky Normal Form for the given CFG.

Chey

# Greibach Normal Form

A CFG is in Greibach Normal Form if the productions are in the following forms:

$$A \rightarrow b$$

$$A \rightarrow bC_1C_2 \ldots\ldots C_n$$

Where $A, C_1, \ldots, C_n$ are Non-Terminals and $b$ is a Terminal.

## Steps to convert a Given CFG to GNF :-

Step 1: Check if the given CFG has any Unit Productions or Null Productions and Remove if there are any (using the Unit & Null Productions removal techniques discussed in the previous lecture).

Step 2: Check whether the CFG is already in chomsky Normal Form (CNF) and convert it to CNF if it is not. (using the CFG to CNF Conversion techniques discussed in the previous lecture)

Step 3:- change the Names of the Non-Terminal Symbols into some $A_i$ in ascending order of $i$

---

Example :-

$S \rightarrow CA \mid BB$ Replace S with $A_1$

$B \rightarrow b \mid SB$ C with $A_2$

$C \rightarrow b$ A with $A_3$

$A \rightarrow a$ B with $A_4$

We get:

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$

$$A_4 \rightarrow b \mid A_1A_4$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

---

Step 4:: Alter the rules so that the Non-Terminals are in ascending order, such that If the Production is of the form $A_i \rightarrow A_j x$, then, $i < j$ and should never be $i \geq j$

$$A_4 \rightarrow b \mid \underline{A_1} A_4$$

$$A_4 \rightarrow b \mid \underline{A_2} A_3 A_4 \mid A_4A_4A_4$$

$$A_4 \rightarrow b \mid b A_3 A_4 \mid A_4 A_4 A_4$$

Left Recursion.

Step 5: Remove Left Recursion

(Greibach Normal Form)

(Conversion of CFG to CNF - Removal of Left Recursion)

$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$

$A_4 \rightarrow b \mid A_1 A_4 \longrightarrow A_4 \rightarrow b \mid b A_3 A_4 \mid A_4 A_4 A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Left Recursion

Step 5 :- Remove Left Recursion

Introduce a New Variable to remove the Left Recursion

$A_4 \rightarrow b \mid b A_3 A_4 \mid A_4 A_4 A_4$

$A_4 \rightarrow A_4 A_4 Z \mid A_4 A_4$ ---(new production one with Z and one without Z)

$A_4 \rightarrow b \mid b A_3 A_4 \mid b Z \mid b A_3 A_4 Z$ ...(it is in GNF)

Now the grammar is :

$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$

$A_4 \rightarrow b \mid b A_3 A_4 \mid b Z \mid b A_3 A_4 Z$

$Z \rightarrow A_4 A_4 \mid A_4 A_4 Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

~~$A_1 \rightarrow b A_3 \mid A\, b \mid b A_3 A_4 \mid b Z \mid b A_3 A_4 Z$~~

$A_1 \rightarrow b A_3 \mid b A_4 \mid b A_3 A_4 A_4 \mid b Z A_4 \mid b A_3 A_4 Z A_4$

---- (Replacing $A_2 \rightarrow b$) ($A_4 \rightarrow b$)

$A_4 \rightarrow b \mid b A_3 A_4 \mid b Z \mid b A_3 A_4 Z$

$Z \rightarrow b A_4 \mid b A_3 A_4 A_4 \mid b Z A_4 \mid b A_3 A_4 Z A_4 \mid$

---(Replacing $A_4 \rightarrow b$)

$b A_4 \mid b A_3 A_4 A_4 Z \mid b Z A_4 Z \mid b A_3 A_4 Z A_4 Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

# Pumping Lemma (for Context free Languages):

Pumping Lemma (for CFL) is used to prove that a language is NOT context Free.

## Context Free Language

In formal language theory, a context Free Language is a language generated by some context Free Grammar.
The set of all CFL is identical to the set of languages accepted by pushdown Automata.

Context Free Grammar is identified by 4 tuples as $G = \{V, \Sigma, S, P\}$
where

$V$ = set of Variables or Non-Terminal Symbols

$\Sigma$ = set of Terminal Symbols.

$S$ = Start Symbol

$P$ = Production Rule.

Context Free Grammar has Production Rule of the form

$$A \to a$$

where, $a = \{V \cup \Sigma\}^*$ and $A \in V$

If A is a Context Free Language, then, A has a Pumping Length 'p' such that any string 's', where $|s| \geq P$ may be divided into 5 Pieces $S = uvxyz$ Such that the following conditions must be true:

1) $u v^i x y^i z$ is in A for every $i \geq 0$
2) $|vy| > 0$
3) $|vxy| \leq P$

To Prove that a Language is Not context Free using Pumping Lemma (for CFL) follow the steps below: (we prove using CONTRADICTION)

→ Assume that A is context Free

→ It has to have a Pumping Length (say $P$)

→ All strings longer than P can be pumped $|s| \geq P$

→ Now find a string 's' in A such that $|s| \geq P$

→ Divide S into $uvxyz$

→ Show that $uv^ixy^iz \notin A$ for some $i$

→ Then consider the ways that S can be divided into $uvxyz$

→ show that none of these can satisfy all the 3 pumping conditions at the same time.

→ S cannot be pumped == CONTRADICTION