# Week 9: Lecture Notes.

**Theorem:**

Let $G = (V, T, P, S)$ be a context free grammar. Then $S \overset{*}{\Rightarrow} \alpha$ iff there is a derivation tree in grammar $G$ with yield $\alpha$.

**Example:**

Consider CFG $G$ whose productions are

$S \to aAS \mid a$, $A \to SbA \mid SS \mid ba$. Show that $S \overset{*}{\Rightarrow} aabbaa$ and construct a derivation tree whose yield is $aabbaa$

**Sol$^{n}$:**

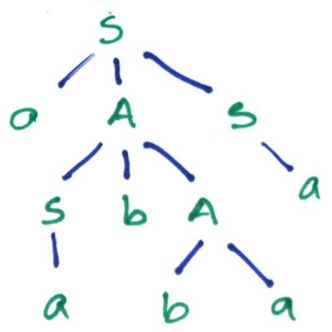
$S \Rightarrow a\underline{A}S \Rightarrow a\underline{S}bAS \Rightarrow aab\underline{A}S \Rightarrow aabba\underline{S} \Rightarrow aabbaa$

another derivation:

$S \Rightarrow aAS \Rightarrow aAa$

$\Rightarrow aSbAa \Rightarrow aabAa \Rightarrow aabbaa$

$\Rightarrow aSbAa \Rightarrow aSbbaa \Rightarrow aabbaa$




## Leftmost and rightmost derivations: ambiguity

- A derivation $A \overset{*}{\Rightarrow} w$ is called a <u>leftmost derivation</u> if we apply a production only to the leftmost variable at every step
- A derivation $A \overset{*}{\Rightarrow} w$ is a <u>rightmost derivation</u> if we apply a production only to the rightmost variable at every step.

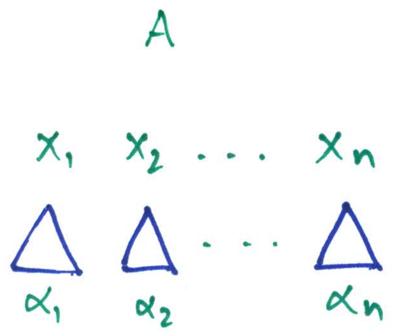
## Theorem:

If $A \overset{*}{\Rightarrow} w$ in $G$, then there is a leftmost derivation of $w$

## Proof:

By induction on the number of steps in $A \overset{*}{\Rightarrow} w$, we can prove it.

- If $w \in L(G)$ for CFG $G$, then $w$ has atleast one parse tree and corresponding to a particular parse tree, $w$ has a unique leftmost and a unique rightmost derivation




$$\left.\begin{aligned} A &\Rightarrow X_1X_2 \ldots X_n \\ &\overset{*}{\Rightarrow} \alpha_1 X_2 \ldots X_n \\ &\overset{*}{\Rightarrow} \alpha_1\alpha_2 \ldots X_n \\ &\;\vdots \\ &\Rightarrow \alpha_1\alpha_2 \ldots \alpha_n \end{aligned}\right\}$$ derivation $X_i \overset{*}{\Rightarrow} \alpha_i$ are recursively made left most

$$\left.\begin{aligned} A &\Rightarrow X_1X_2 \ldots X_n \\ &\overset{*}{\Rightarrow} X_1 X_2 \ldots X_{n-1}\alpha_n \\ &\;\vdots \\ &\Rightarrow \alpha_1\alpha_2 \ldots \alpha_n \end{aligned}\right\}$$ derivation $X_i \overset{*}{\Rightarrow} \alpha_i$ are recursively made right most

- $w$ may have several leftmost or rightmost derivations as they may be more than one parse tree for $w$

## Example:

Let $G$ be the CFG. $S \rightarrow 0B \mid 1A$, $A \rightarrow 010S \mid 1AA$,
$B \rightarrow 1 \mid 1S \mid 0BB$.

For the string 00110101, find

a. the leftmost derivation

b. the rightmost derivation

c. the derivation tree

a). $S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 001SB \Rightarrow 0011AB \Rightarrow 001101AB \Rightarrow$

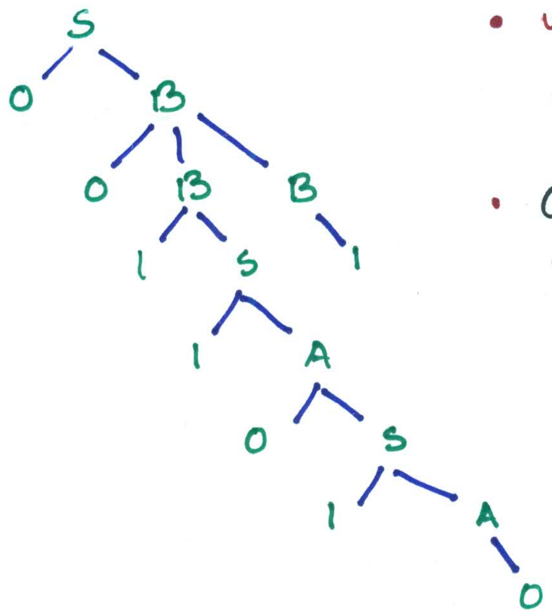
$\Rightarrow 0011010B \Rightarrow 00110101$

b). Corresponding rightmost derivation

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 00B1 \Rightarrow 001S1 \Rightarrow 0011A1 \Rightarrow$

$\Rightarrow 00110S1 \Rightarrow 001101A1 \Rightarrow 00110101$

c). The derivation tree for $w$



- $w$ may have several derivation trees

- Corresponding to each parse tree of $w$ we have a unique leftmost derivation and a unique rightmost derivation

Another leftmost derivation:

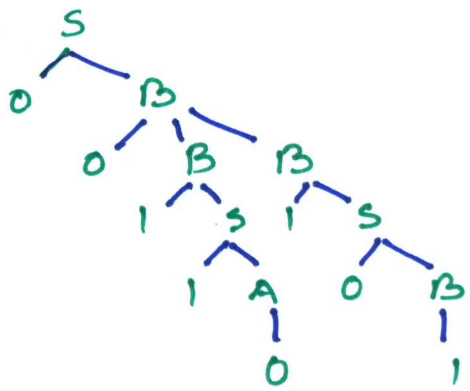
$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 001SB$

$\Downarrow$

$0011015 \Leftarrow 00110B \Leftarrow 0011AB$

$\Downarrow$

$0011010B \Rightarrow 00110101$



Corresponding rightmost derivation

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 00B1S \Rightarrow 00B10B \Rightarrow 00B101$

$\Downarrow$

$00110101 \Leftarrow 0011A101 \Leftarrow 001S101$

# Ambiguity in CFG

**Example:**

"In books selected information is given"

selected $\xrightarrow[\text{refer}]{\text{may}}$ books / information

ambiguous sentence

i.e. sentence may be parsed in two different ways.

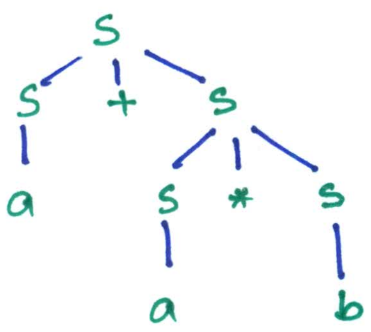
- A terminal string $w \in L(G)$ is ambiguous if $\exists$ two or more derivation trees for $w$.
- A CFG $G$ is ambiguous if $\exists$ some $w \in L(G)$, which is ambiguous.
- A CFL for which every CFG is ambiguous is said to be an inherently ambiguous CFL.

**Example:**

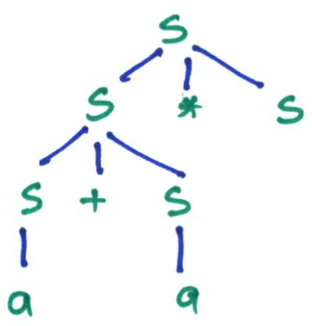
$G = (\{S\}, \{a, b, +, *\}, P, S)$, where $P$ consists of

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

Parse trees for $a + a * b$



$T_1$



$T_2$

The leftmost derivations of $a+a*b$ induced by $T_1, T_2$

- $S \Rightarrow S+S \Rightarrow a+S \Rightarrow a+S*S \Rightarrow a+a*S \Rightarrow a+a*b$
- $S \Rightarrow S*S \Rightarrow S+S*S \Rightarrow a+S*S \Rightarrow a+a*S \Rightarrow a+a*b$

Thus $a+a*b$ is ambiguous

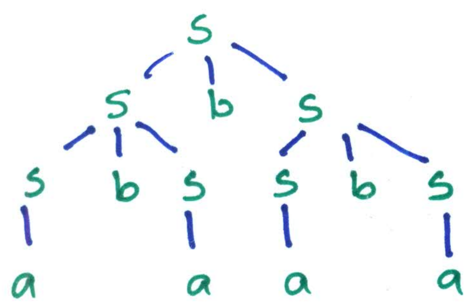
**Example:**

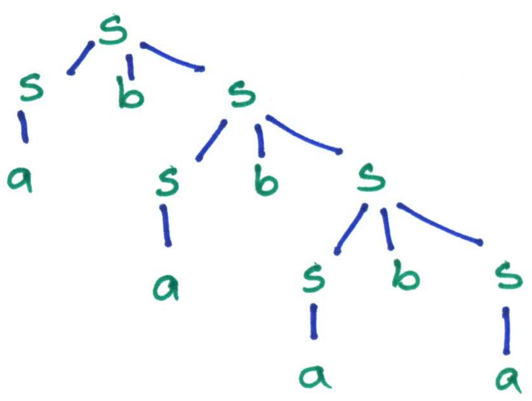
If $G$ is the CFG $S \rightarrow SbS \mid a$. Show that $G$ is ambiguous.

**Sol$^{n}$:**

We have to find a $w \in L(G)$ which is ambiguous.

Let $w = abababa$





# Simplification of CFG's

- $G = (V, T, P, S) \rightarrow$ CFG
- $L(G)$ generation $\rightarrow$ [ may be not all symbols in $V \cup T$ are used
  OR
  may be not all rules in $P$ are used
- Try to eliminate those symbols in $V \cup T$ or rules in $P$ from $G$ and construct a CFG $G'$ that realizes $L(G)$

**Example:**

$G = (\{S, A, B, D, E\}, \{a, b, c\}, P, S)$

where $P = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b, B \rightarrow D, E \rightarrow c|\varepsilon\}$

**Sol$^{n}$:**

Note that $L(G) = \{ab\}$

Let $G' = (\{S, A, B\}, \{a, b\}, P', S)$

where $P' = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b\}$

WHY? [
- eliminate symbols $D, E, c$
- eliminate productions $B \rightarrow D$, $E \rightarrow c|\varepsilon$

1. $D$ doesnot derive any terminal string
2. $E$ and $c$ doesnot appear in any sentential form
3. $E \rightarrow \varepsilon$ is a null production
4. $B \rightarrow D$ simply replaces $B$ by $D$ (unit production)

# Algorithms to construct reduced grammars.

## Construction 1

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

Define $G' = (V', T, P', S)$ as follows:

### Construction of $V'$

We define $W_i \subseteq V$ by recursion:

$W_1 = \{ X \in V \mid$ there exists a production $X \to w$ where $w \in T^* \}$

[if $W_1 = \phi$; then some variable will remain after application of any production and so $L(G) = \phi$]

$W_{i+1} = W_i \cup \{ X \in V \mid$ there exists a production $A \to \alpha$ with $\alpha \in (T \cup W_i)^* \}$

Thus $W_i \subseteq W_{i+1}$ $\forall i$

Also, $W_K = W_{K+1}$ for some $K \leq |V|$ as $V$ is a finite set of variables.

Therefore $W_K = W_{K+j}$ for $j \geq 1$

We define $V' = W_K$

### Construction of $P'$

$P' = \{ A \to \alpha \mid A, \alpha \in (V' \cup T)^* \}$

- $S \in V'$ (Prove it !!)

**Example:**

Let $G = (V, T, P, S)$ be given by the productions $S \to AB$, $A \to a$, $B \to b$, $B \to C$, $E \to d$. Find $G'$ such that every variable in $G'$ derives some terminal string.

**Sol$^n$:**

**Construction of $V'$**

$W_1 = \{A, B, E\}$

$W_2 = W_1 \cup \{X \in V \mid \exists$ some production $X \to \alpha$ with $\alpha \in (W_1 \cup T)^*\}$

$= W_1 \cup \{S\}$

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

$W_3 = W_2 \cup \{X \in V \mid \exists$ some production $X \to \alpha$ with $\alpha \in (W_2 \cup T)^*\}$

$= W_2 \cup \phi$

$= W_2$

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

**Construction of $P'$**

$P' = \{X \to \alpha \mid X, \alpha \in (V' \cup T)^*\}$

$= \{S \to AB, A \to a, B \to b, E \to d\}$

$\therefore G' = (\{S, A, B, E\}, \{a, b, d\}, P', S)$

**Note:**

$S \in V'$ always

## Theorem 1

Given a CFG $G = (V, T, P, S)$, with $L(G) \neq \phi$, we can effectively find an equivalent CFG $G' = (V', T, P', S)$ such that for each $A$ in $V'$ there is some $w$ in $T^*$ for which $A \overset{*}{\Rightarrow} w$

**Proof:**

Let $V', P'$ are constructed from G following the above algorithm.

**Claim:**

i) for each $A \in V'$, $A \overset{*}{\underset{G'}{\Rightarrow}} w$ for some $w \in T^*$

conversely $A \overset{*}{\underset{G'}{\Rightarrow}} w$ implies $A \in V'$

ii) $L(G) = L(G')$

**Proof of claim (i)**

Note that $W_K = W_1 \cup W_2 \cup \ldots \cup W_K$

We prove by induction on $i$ for $i = 1, 2, \ldots K$ that $A \in W_i$ implies $A \underset{G'}{\Rightarrow} w$ for some $w \in T^*$

**Base:**

Let $A \in W_1$

Then $\underline{A \rightarrow w \text{ is in } P}$ for some $\underline{w \in T}^*$

By construction of $\underline{P'}$, $\underline{A \rightarrow w \text{ is in } P'}$

This in turn implies $A \overset{*}{\underset{G'}{\Rightarrow}} w$

**Induction:**

Let the result be true for $i$

Consider $A \in W_{i+1}$

Then, either

a. $A \in W_i$, in which case $A \overset{*}{\underset{G}{\Rightarrow}} w$ for some $w \in T^*$ by induction

b. $\exists$ a production $A \to \alpha$ in $P$ with $\alpha \in (T \cup W_i)^*$ in which case $A \to \alpha$ is in $P'$ by construction of $P'$

Then $\alpha$ may be written as

$\alpha = x_1 x_2 \dots x_n$ where $x_j \in T \cup W_i$

If $x_j \in W_i$, then by induction hypothesis,

$$x_j \overset{*}{\underset{G'}{\Rightarrow}} w_j \text{ for some } w_j \in T^*$$

Hence, $A \overset{*}{\underset{G'}{\Rightarrow}} w_1 w_2 \dots w_n \in T^*$

Conversely we can show that $A \overset{*}{\underset{G'}{\Rightarrow}} w$ implies $A \in V'$

We can prove it by induction on $K$ = number of steps in the derivation $A \overset{*}{\underset{G'}{\Rightarrow}} w$ in a similar manner.

**Base:** $K = 1$

production $A \to w$ is in $P$, so $A \in V'$

## Induction

Let $A \underset{G}{\Rightarrow} X_1 X_2 \ldots X_n \overset{K-1}{\underset{G}{\Rightarrow}} w$ by a derivation of $K$ steps.

We may write $w = w_1 w_2 \ldots w_n$ where $X_i \overset{*}{\Rightarrow} w_i$ for $1 \le i \le n$, by a derivation of fewer than $K$ steps.

$\therefore$ By induction hypothesis $X_i \in V'$, $1 \le i \le n$

Hence, $A \rightarrow X_1 X_2 \ldots X_n$ is in $P'$ by construction of $P'$.

Let $V' = W_1 \cup W_2 \cup \ldots \cup W_l$ and $W_l = W_{l+1}$

Let $j$ be the least integer, $1 \le j \le l$ for which $W_j$ contains all $X_1, X_2, \ldots, X_n$

As $A \rightarrow X_1 X_2 \ldots X_n$ is in $P$, where $X_1, X_2 \ldots X_n \in (T \cup W_j)^*$ we must have $A \in W_{j+1} \subseteq V'$

Thus $A \overset{K}{\underset{G}{\Rightarrow}} w$ implies $A \in V'$

## Proof of claim (ii)

- $L(G') \subseteq L(G)$ as $P' \subseteq P$ and $V' \subseteq V$ ——— (1)
- To prove $L(G) \subseteq L(G')$, we need to prove the following auxilliary result.

$$A \overset{*}{\underset{G'}{\Rightarrow}} w \quad \text{if} \quad A \overset{*}{\underset{G}{\Rightarrow}} w \quad \text{for some } w \in T^*$$

We prove this by induction on $K$ = the number of steps in the derivation $A \overset{*}{\underset{G}{\Rightarrow}} w$

## Base:

$K=1$

Let $A \underset{G}{\Rightarrow} w$ where $w \in T^*$

Then $A \to w$ is in $P'$

$\therefore A \underset{G'}{\Rightarrow} w$

## Induction

Result is true for $\leq K$ steps.

Let $A \overset{K+1}{\underset{G}{\Rightarrow}} w$ where $w \in T^*$. We need to prove $A \overset{K+1}{\underset{G'}{\Rightarrow}} w$

We may write $w = w_1 w_2 \ldots w_n$ and

$$A \underset{G}{\Rightarrow} X_1 X_2 \ldots X_n \overset{K}{\underset{G}{\Rightarrow}} w \quad \text{s.t.} \quad X_j \overset{*}{\underset{G}{\Rightarrow}} w_j, \ 1 \leq j \leq n$$

is in $P$

- If $X_j \in T$ then $X_j = w_j$
- If $X_j \in V$ then $X_j \overset{*}{\underset{G}{\Rightarrow}} w_j$ implies $X_j \in V'$

  by claim (i)

Also $X_j \overset{*}{\underset{G}{\Rightarrow}} w_j$ in at most steps.

Hence by induction hypothesis $X_j \overset{*}{\underset{G'}{\Rightarrow}} w_j$

Also $X_1, X_2 \ldots X_n \in (V' \cup T)^*$ implies

$A \to X_1 X_2 \ldots X_n$ is in $P'$

Thus $A \underset{G'}{\Rightarrow} X_1 X_2 \ldots X_n \overset{K}{\underset{G'}{\Rightarrow}} w_1 w_2 \ldots w_n$

i.e. $A \overset{K+1}{\underset{G'}{\Rightarrow}} w$

In particular, $S \overset{*}{\Rightarrow} w$ for $w \in T^*$

This implies $L(G) \subseteq L(G')$ —— (ii)

From (i) and (ii), we get $L(G) = L(G')$

## Theorem 2.

Given a CFG $G = (V, T, P, S)$, we can effectively find an equivalent CFG $G' = (V', T', P', S)$ such that for each $X \in V' \cup T'$, there exists $\alpha, \beta \in (V' \cup T')^*$ for which

$$S \stackrel{*}{\Rightarrow} \alpha X \beta$$

**Proof:**

The set $V' \cup T'$ of symbols appearing in sentential form of $G$ is constructed by an iterative algorithm.

Algorithm 2.

Construction 2.

a. Place $S$ in $V'$

b. If $A$ is placed in $V'$ and $A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$, then add all variables of $\alpha_1, \alpha_2 \dots \alpha_n$ to the set $V'$ and all terminals of $\alpha_1, \alpha_2 \dots \alpha_n$ to $T'$

c. $P'$ is the set of productions of $P$ containing only symbols of $V' \cup T'$

**Example:**

Consider $G = (\{S, A, B, E\}, \{a, b, c\}, P, S)$ where $P$ consists of $S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow b$, $E \rightarrow c$

Construct an equivalent grammar $G' = (V', T', P', S)$ such that every symbol in $V' \cup T'$ appears in some sentential form of $G'$

**Sol$^{\underline{n}}$:**

0. Initially set $V' = \{S\}$

$$S \rightarrow AB$$

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

   $S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow b$

2. $V' = \{S, A, B\}$, $T' = \{a, b\}$

No further modification is possible as $S \to AB$, $A \to a$, $B \to b$.

$\therefore \quad V' = \{S, A, B\}, \quad T' = \{a, b\}$

$P' = \{ S \to AB,\ A \to a,\ B \to b \}$

**Algorithm 1:**

input: CFG $G$ with $L(G) \neq \phi$

output: equivalent CFG $G'$ such that every variable in $G'$ derives some terminal string

**Algorithm 2:**

input: CFG $G = (V, T, P, S)$

output: equivalent CFG $G' = (V', T', P', S)$ such that every symbol in $V' \cup T'$ appears in some sentential form.

**Definitions:**

- $G$ is said to be reduced or non-redundant if every symbol in $V \cup T$ appears in the course of the derivation of some terminal string

i.e. for every $X \in V \cup T$, $\exists$ some $\alpha, \beta \in (V \cup T)^*$ such that $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$, where $w \in T^*$

- We say $X$ is useful in the derivation of terminal string.

## Theorem 3.

For every CFG $G$, there exists a reduced grammar $G'$ which is equivalent to $G$

**Proof:**

We construct the reduced grammar in two steps:

Algorithm 3

Construction 3

Step 1:
Construct a CFG $G_1$ equivalent to a $G$ using Algorithm 1, so that every variable in $G_1$ derives some terminal string

Step 2:
Construct a CFG $G'$ equivalent to $G_1$ using Algm. 2 so that every symbol (variable as well as terminal symbol) in $G'$ appears in some sentential form of $G'$

By step 2, every symbol $X$ in $G'$ appears in some sentential form, say $\alpha X \beta$

By step 1, every symbol in $\alpha X \beta$ derives some terminal string.

Note that all symbols of $G'$ are symbols of $G$

It follows that $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$ for some terminal string $w$

Hence $G'$ is reduced.

**NOTE:**

To get a reduced grammar, we must first apply Algorithm 1 and then Algorithm 2. For if Algorithm 2 is applied first and then Algorithm 1, we may not get a grammar with it's most reduced form

**Example:**

Find a reduced grammar equivalent to the grammar $G$ whose productions are

$$S \to AB \mid CA, \; B \to BC \mid AB, \; A \to a, \; C \to aB \mid b$$

**Sol$^{\underline{n}}$:**

$$G = (V, T, P, S) \xrightarrow[1]{\text{Algm}} G_1 = (V', T', P', S) \xrightarrow[2]{\text{Algm}} G_2 = (V'', T'', P'', S)$$

$\underline{G_1}$

$W_1 = \{A, C\}$ as $A \to a$, $C \to b$ are productions with a terminal string on right hand side

$W_2 = \{A, C\} \cup S$ as $S \to CA$ is a production with R.H.S. $\in \{T \cup W_1\}^*$

$W_3 = \{A, C, S\} \cup \phi$ as $\{X \in V \mid X \to \alpha \text{ with } \alpha \in (T \cup W_2)^*\}$ is empty

$\therefore V' = \{A, C, S\}, \; P' = \{A \to a, C \to b, S \to CA\}$

$\underline{G_2}$

0. $V'' = \{S\}$

$S \to CA$

1. $V'' = \{S, C, A\}$

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

2. $V'' = \{S, C, A\}, T'' = \{a, b\}$

$P'' = \{S \to CA, C \to b, A \to a\}$

$\therefore$ The reduced equivalent grammar is

$$G' = (\{S, A, C\}, \{a, b\}, \{S \to CA, C \to b, A \to a\}).$$

## Example:

Construct a reduced grammar with no useless symbol equivalent to the grammar

$S \to aAa$, $A \to Sb \mid bcc \mid DaA$, $C \to abb \mid DD$, $E \to aC$,

$D \to aDA$

**Solⁿ:**

**Step 1:** $W_1 = \{C\}$ as $C \to abb$ is the only production with a terminal string on the R.H.S.

$W_2 = \{C\} \cup \{A, E\}$ as $A \to bCC$ and $E \to aC$ are productions with R.H.S. in $(T \cup W_1)^*$

$W_3 = W_2 \cup \{S\}$ as $S \to aAa$ and $aAa \in (T \cup W_2)^*$

$W_4 = W_3 \cup \phi$

$\therefore \quad V' = W_4 = \{A, E, C, S\}$

$P' = \{A_1 \to \alpha \mid \alpha \in (V' \cup T)^*\}$

$= \{S \to aAa,\ A \to Sb \mid bcc,\ C \to abb,\ E \to aC\}$

**Step 2:**

0. $V'' = \{S\}$

$S \to aAa$

1. $V'' = \{S, A\}$, $T'' = \{a\}$

$S \to aAa$, $A \to Sb \mid bCC$

2. $V'' = \{S, A, C\}$, $T'' = \{a, b\}$

$S \to aAa$, $A \to Sb \mid bCC$, $C \to abb$

3. $V'' = \{S, A, C\}$, $T'' = \{a, b\}$

$P'' = \{S \to aAa,\ A \to Sb \mid bC,\ C \to abb\}$

Algorithm 1: Transforms G to an equivalent G' where every variable derives some terminal strings.

Algorithm 2: Transforms G to an equivalent G' where every symbol appears in some sentential form.

Algorithm 3: Transforms G to an equivalent G' where G' is reduced, i.e. have no useless symbols.

**Example:**

$S \to aAa,\ A \to Sb \mid bCC \mid DaA,\ C \to abb \mid DD,\ E \to aC$

$D \to aDA$

**Algorithm 1:**

$G'$ : $V'$

$W_0 = \{C\}$

$W_1 = \{C\} \cup \{A, E\} = \{C, A, E\}$

$W_2 = \{A, C, E\} \cup \{S\}$

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

$P'$ : $S \to aAa,\ A \to Sb \mid bCC,\ C \to abb,\ E \to aC$

**Algorithm 2:**

$G''$ :

$V'' = \{S\},\quad S \to aAa$

$V'' = \{S, A\},\ T'' = \{a\},\quad A \to Sb \mid bCC$

$V'' = \{S, A, C\},\ T'' = \{a, b\},\quad C \to abb$

$V'' = \{S, A, C\},\ T'' = \{a, b\}.$

$P'' = \{S \to aAa,\ A \to Sb \mid bCC,\ C \to abb\}$

**Algorithm 4:** Transforms $G$ to $G'$ with no null production where $L(G') = L(G) - \{\varepsilon\}$

**Algorithm 5:** Transforms $G$ to an equivalent $G'$ with no null productions or unit productions

**Algorithm 6:** Transforms $G$ to an equivalent reduced $G'$ with no null productions or unit productions

## Algorithm 4

$G$ $\qquad G_1 = (V_1, T_1, P_1, S_1)$

$L(G) = L(G) - \{\varepsilon\}$

$\downarrow$

$G_2$ (no null production)

$L(G_2) = L(G)$

$\downarrow$

$G_3$ (no unit production)

$\downarrow$ Algorithm 1

$G_4$ (all variables derive some terminal string)

$\downarrow$ Algorithm 2

$G_5$ (every symbol appears in some sentential form)

- if $\varepsilon \notin L(G)$, then $L(G_1) = L(G)$ set $G_2 = G_1$
- if $\varepsilon \in L(G)$ then $L(G_2) = L(G)$ where

$$G_2 = (V_2, T_2, P_2, S_2), \quad V_2 = V_1 \cup \{S_2\}$$

$$P_2 = P_1 \cup \{S_2 \to \varepsilon \mid S_1\}$$

## Reduction to CNF

1. Eliminate null productions and unit productions
2. Elimination of terminals on R.H.S. of each productions
   i. $A \to a$ $\quad A \to BC$ type productions are retained
   ii. $A \to X_1 X_2 \ldots X_n$ type productions:
   if some $X_i = a_i$, a terminal, introduce a new variable $C_{a_i}$ and introduce a production $C_{a_i} \to a_i$

3. Restricting the number of variables on R.H.S.

$A \to A_1 A_2 \ldots A_n$, $n \geq 3$ all $A_i$ variables.

$A \to A_1 C_1$, $C_1 \to A_2 C_2$, $\ldots$, $C_{n-1} \to A_{n-1} A_n$

## Elimination of Null productions ($\varepsilon$- productions)

- A production of the form $A \to \varepsilon$ is called a null production where $A$ is a variable

A variable $A$ in a CFG is nullable if

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

## Theorem 4

If $L = L(G)$ $(\varepsilon \notin L)$ for some CFG $G = (V, T, P, S)$ then $L - \{\varepsilon\}$ is $L(G')$ for a CFG $G'$ with no null productions

**Proof:**

We construct $G' = (V, T, P', S)$ as follows:

Construction 4.

We find the nullable variables recursively:

(i) $W_1 = \{A \in V \mid A \to \varepsilon \text{ is in } P\}$

(ii) $W_{i+1} = W_i \cup \{A \in V \mid \text{there exists a production } A \to \alpha \text{ with } \alpha \in W_i^*\}$

**Step 1:** Construction of the set of nullable variables

By definition, $W_i \subseteq W_{i+1}$ $\forall i$

As $V$ is finite, $W_{k+1} = W_k$ for some $k < |V|$

$\therefore$ $W_{k+j} = W_k$ $\forall j$

Let $W = W_k$

$W$ is set of all nullable variables

i.e. for every variable $X$ in $W$ we have

$$X \overset{*}{\underset{G}{\Rightarrow}} \varepsilon$$

Step 2: Construction of $P'$

- Any production whose R.H.S. doesnot have any nullable variable is included in $P'$
- If $A \to X_1 X_2 \ldots X_n$ is in $P$, then add all productions $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ in $P'$ where

  i. $\alpha_i = X_i$ if $X_i \notin W$

  ii. $\alpha_i = X_i$ or $\varepsilon$ if $X_i \in W$

  iii. $\alpha_1 \alpha_2 \ldots \alpha_n \neq \varepsilon$ i.e. not all $\alpha_i$'s are $\varepsilon$

Thus productions of $P'$ are obtained by

- either not erasing any nullable variables on the R.H.S. of $A \to X_1 X_2 \ldots X_n$ (in $P$)
- or erasing some or all nullables variables provided some symbol appear on the R.H.S. of $A \to X_1 X_2 \ldots X_n$ after erasing.

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

Claim: $G'$ has no null productions

i.e. $\forall\ A \in V,\ w \in T^*$

$$A \overset{*}{\underset{G'}{\Rightarrow}} w \text{ iff } w \neq \varepsilon \text{ and } A \overset{*}{\underset{G}{\Rightarrow}} w$$

Example:

Consider the grammar $G$ whose productions are

$S \to aS \mid AB, \quad A \to \varepsilon, \quad B \to \varepsilon, \quad D \to b$

Construct a grammar $G'$ without null productions generating $L(G) - \{\varepsilon\}$.

Sol$^n$

Step 1: Construction of the set $W$ of all nullable variables

$W_1 = \{A, B\}$

$W_2 = \{W_1 \cup \{A_1 \in V \mid A_1 \to \alpha \text{ is in } P \text{ with } \alpha \in W_1^*\}$

$= \{A, B, S\}$ as $S \to AB$ is in $P$ with $AB \in W_1^*$

$W_3 = W_2 \cup \phi = W_2$

Thus, $W = \{A, B, S\}$

Step 2: Construction of $P'$

1. $D \to b$ is included in $P'$
2. $S \to aS$ gives rise to $S \to aS$ and $S \to a$
3. $S \to AB$ gives rise to $S \to AB$, $S \to A$, $S \to B$

$\therefore$ The required CFG without null productions is

$G' = (\{S, A, B\}, \{a, b\}, P', S)$

where $P'$ consists of

$D \to b, \quad S \to aS, \quad S \to a, \quad S \to AB$

$S \to A, \quad S \to B$

To prove $L(G') = L(G) - \{\varepsilon\}$ we prove an auxilliary result given by the following relation

$\forall\ A \in V$ and $w \in T^*$

$$A \overset{*}{\underset{G'}{\Rightarrow}} w \text{ iff } w \neq \varepsilon \text{ and } A \overset{*}{\underset{G}{\Rightarrow}} w$$

Proof:

(if) Let $A \overset{*}{\underset{G}{\Rightarrow}} w$ and $w \neq \varepsilon$

We prove that $A \overset{*}{\underset{G'}{\Rightarrow}} w$ by induction on the number of steps in the derivation of $A \overset{*}{\underset{G}{\Rightarrow}} w$

Basis:

If $A \underset{G}{\Rightarrow} w$ and $w \neq \varepsilon$ then $A \to w$ is in $P'$ and

so $\quad A \underset{G'}{\Rightarrow} w$

Induction:

Assume the result for derivation in at most $i$ steps.

Let $A \overset{i+1}{\underset{G}{\Longrightarrow}} w$ and $w \neq \varepsilon$

We can split the derivation as

$$A \underset{G}{\Rightarrow} X_1 X_2 \cdots X_n \overset{i}{\underset{G}{\Rightarrow}} w_1 w_2 \cdots w_n$$

where

$$w = w_1 w_2 \cdots w_n$$

and $\quad X_j \overset{*}{\underset{G}{\Rightarrow}} w_j$

As $w \neq \varepsilon$, not all $w_j$'s are $\varepsilon$

- if $w_j \neq \varepsilon$, then by induction hypothesis $X_j \overset{*}{\underset{G}{\Rightarrow}} w_j$ and $w_j \neq \varepsilon$ implies $X_j \overset{*}{\underset{G'}{\Rightarrow}} w_j$
- if $w_j = \varepsilon$ then $X_j \in W$ (i.e. $X_j$ is a nullable variable)

so using production $A \to X_1 X_2 \ldots X_n$ in $P$ we construct production $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ in $P'$ where

$$\alpha_j = \begin{cases} X_j & \text{if } w_j \neq \varepsilon \\ \varepsilon & \text{if } w_j = \varepsilon \text{ (i.e. } X_j \in W) \end{cases}$$

Since $w \neq \varepsilon$ not all $\alpha_j$'s are $\varepsilon$

$$\therefore A \underset{G'}{\Rightarrow} \alpha_1 \alpha_2 \ldots \alpha_n \overset{*}{\underset{G'}{\Rightarrow}} w_1 \alpha_2 \ldots \alpha_n$$

$$\overset{*}{\underset{G'}{\Rightarrow}} w_1 w_2 \alpha_3 \ldots \alpha_n$$

$$\vdots$$

$$\overset{*}{\underset{G'}{\Rightarrow}} w_1 w_2 \ldots w_n = w.$$

(Only if)

We prove 'only if' part by induction on the number of steps in the derivation of $A \overset{*}{\underset{G'}{\Rightarrow}} w$

<u>Basis</u> If $A \underset{G'}{\Rightarrow} w$, then $A \to w$ is in $P'$

By construction of $P'$, $A \to w$ is obtained from some production $A \to X_1 X_2 \dots X_n$ in $P$ by erasing some (or none of the) nullable variables.

Hence $A \underset{G}{\Rightarrow} X_1 X_2 \dots X_n \overset{*}{\underset{G}{\Rightarrow}} w$.

<u>Induction:</u>

Assume the result for derivation in at most $i$ steps

Let $A \overset{j+1}{\underset{G'}{\Rightarrow}} w$

This can be split as

$$A \underset{G'}{\Rightarrow} X_1 X_2 \dots X_n \overset{*}{\underset{G'}{\Rightarrow}} w_1 w_2 \dots w_n = w$$

where $X_i \overset{*}{\underset{G'}{\Rightarrow}} w_i$ in fewer than $j+1$ steps.

$\therefore$ By induction hypothesis $X_i \overset{*}{\underset{G}{\Rightarrow}} w_i$

The first production $A \to X_1 X_2 \dots X_n$ in $P'$ is obtained from some production $A \to \alpha$ in $P$ by erasing some (or none of the) nullable variables in $\alpha$

$\therefore \; A \underset{G}{\Rightarrow} \alpha \overset{*}{\underset{G}{\Rightarrow}} X_1 X_2 \ldots X_n$

- if $X_i \in T$ then $X_i \overset{0}{\underset{G}{\Rightarrow}} X_i = w_i$

  if $X_i \in V$ then by induction hypothesis

$$X_i \overset{*}{\underset{G}{\Rightarrow}} w_i$$

$\therefore \; A \overset{*}{\underset{G}{\Rightarrow}} X_1 X_2 \ldots X_n \overset{*}{\underset{G}{\Rightarrow}} w_1 w_2 \ldots w_n = w$

Thus by the principle of induction whenever

$A \underset{G'}{\Rightarrow} w$, we have $A \underset{G}{\Rightarrow} w$ and $w \neq \varepsilon$

- Applying the claim to $S$, we have $w \in L(G')$ iff $w \in L(G)$ and $w \neq \varepsilon$

  $\Rightarrow \; L(G') = L(G) - \{\varepsilon\}$

## Corollary 1:

An algorithm to decide whether $\varepsilon \in L(G)$ for a CFG:

i. Construct $W$, the set of the nullable variables. at most $|V|$ steps

ii. test whether $S \in W$

## Corollary 2

If $G = (V, T, P, S)$ is a CFG, we can find an equivalent CFG $G_1 = (V_1, T, P_1, S_1)$ without null productions except $S_1 \to \varepsilon$ when. $\varepsilon \in L(G)$. When $S_1 \to \varepsilon$ is in $P_1$, $S_1$ doesnot appear on the R.H.S. of any production in $P_1$.

**Proof:**

By corollary 1, we can decide whether $\varepsilon \in L(G)$

**Case 1:**

$\varepsilon \notin L(G)$

Then $G_1$ obtained by using Theorem 4 is the required equivalent grammar.

**Case 2:**

$\varepsilon \in L(G)$

Construct $G' = (V, T, P', S)$ using Theorem 4

Then $L(G') = L(G) - \{\varepsilon\}$

Define $G_1 = (V \cup \{S_1\}, T, P_1, S_1)$

where $P_1 = P' \cup \{S_1 \to S,\ S_1 \to \varepsilon\}$

$S_1$ doesnot appear on the R.H.S. of any production in $P_1$

$\therefore$ $G_1$ is the required CFG with $L(G_1) = L(G)$

## Elimination of Unit productions

- A unit production (or a chain rule) in a CFG $G$ is a production of the form $A \to B$ where $A$ and $B$ are variables in $G$.

**Example:** $G$ is the grammar: $S \to A$, $A \to B$, $B \to C$, $C \to a$

$S \to A$, $A \to B$, $B \to C$, are useful just to replace $S$ by $C$

Thus $L(G) = \{a\}$

If $G'$ is the grammar $S \to a$, then $L(G') = L(G)$

## Theorem 5

If $G$ is a CFG, we can find a CFG $G_1$ which has no null productions or unit productions such that

$$L(G_1) = L(G)$$

**Proof:**

We can apply Corollary 2 of Theorem 4 to grammar

Algorithm 5 / Construction 5: $G$ to get a CFG $G' = (V, T, P, S)$ without null productions such that

$$L(G') = L(G)$$

Let $A \in V$

Step 1: Construction of the set of variables derivable from $A$

Define $W_i(A)$ recursively as follows:

$$W_0(A) = \{A\}$$

$$W_{i+1}(A) = W_i(A) \cup \{B \in V \mid C \to B \text{ is in } P \text{ with } C \in W_i(A)\}$$

By definition of $W_i(A)$, $W_i(A) \subseteq W_{i+1}(A)$

As $V$ is finite, $W_{k+1}(A) = W_k(A)$ for some $k < |V|$

$$\therefore \quad W_{k+j}(A) = W_k(A) \quad \forall j \geq 0$$

Let $W(A) = W_k(A)$, then $W(A)$ is the set of all variables derivable from $A$.

Note: $B \in W(A)$ means $A \overset{*}{\underset{G'}{\Rightarrow}} B$

Step 2: Construction of $A$-productions in $G_1$

The $A$-productions in $G_1$ are

either i. the non-unit $A$-productions in $G'$

or ii. $A \to \alpha$ whenever $B \to \alpha$ is in $G'$

Actually (ii) covers (i) as $A \in W(A)$

Now, we define $G_1 = (V, T, P_1, S)$ where $P_1$ is constructed using Step 2 for every $A \in V$

Claim: $G_1$ is the required grammar.

**Example:**

Let $G$ be $S \to AB$, $A \to a$, $B \to C|b$, $C \to D$, $D \to E$, $E \to a$

Eliminate unit productions and get an equivalent grammar.

**Soln:**

Step 1: $W_0(S) = \{S\}$

$S \to AB$

$W_1(S) = W_0(S) \cup \{A_1 \in V \mid A_2 \to A_1 \text{ is in } P \text{ with } A_2 \in W_0(S)\}$

$= W_0(S) \cup \phi$

$= \{S\}$

$\Rightarrow \boxed{W(S) = \{S\}}$

Similarly $\boxed{W(A) = \{A\}}$

$W_0(B) = \{B\}$

$B \to C|b$

$W_1(B) = W_0(B) \cup \{C\} = \{B, C\}$

$C \to D$

$W_2(B) = W_1(B) \cup \{D\} = \{B, C, D\}$

$D \to E$

$W_3(B) = W_2(B) \cup \{E\} = \{B, C, D, E\}$

$\boxed{W(B) = \{B, C, D, E\}}$

$W_0(C) = \{C\}$

$C \to D$

$W_1(C) = \{C, D\}$

$D \to E$

$W_2(C) = \{C, D, E\}$

$\boxed{W(C) = \{C, D, E\}}$

$W_0(D) = \{D\}$

$D \to E$

$W_1(D) = \{D, E\}$

$\boxed{W(D) = \{D, E\}}$

Step 2:

The productions of $G'$ are

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

(as $\{E \to a$ and $a \notin V\}$ is in $G$ with $E \in W(B)$
$E \in W(C)$
$E \in W(D)$)

Step 3.

To complete the proof, we have to show that

$$L(G') = L(G_1)$$

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

$G_1 : (V, T, P_1, S)$ where consists of

i. all non-unit productions of $G'$

ii. productions $A \to \alpha$ whenever $B \to \alpha$ is in $P$ with $B \in W(A)$ and $\alpha \notin V$

i.e. if $A \to \alpha$ is in $P_1 - P$, then it is induced by $B \to \alpha$ in $P$ with $B \in W(A)$ and $\alpha \notin V$

Now, $B \in W(A)$ implies $A \overset{*}{\underset{G'}{\Rightarrow}} B$

$\therefore \quad A \overset{*}{\underset{G'}{\Rightarrow}} B \underset{G'}{\Rightarrow} \alpha$

So, if $A \overset{*}{\underset{G_1}{\Rightarrow}} \alpha$, then $A \overset{*}{\underset{G'}{\Rightarrow}} \alpha$

Thus, $L(G_1) \subseteq L(G')$

To prove $L(G') \subseteq L(G_1)$

Suppose that $w \in L(G')$ and consider a leftmost derivation of $w$ in $G'$, say

$$S \underset{G'}{\Rightarrow} \alpha_1 \underset{G'}{\Rightarrow} \alpha_2 \underset{G'}{\Rightarrow} \cdots \underset{G'}{\Rightarrow} \alpha_n = w$$

Let $i$ be the smallest index such that $\alpha_i \underset{G'}{\Rightarrow} \alpha_{i+1}$ is obtained by a unit production and $j$ be the smallest index greater than $i$ such that $\alpha_j \underset{G'}{\Rightarrow} \alpha_{j+1}$ is obtained by a non-unit production.

So, $S \underset{G_1}{\Rightarrow} \alpha_i$ and $\alpha_j \underset{G'}{\Rightarrow} \alpha_{j+1}$ can be written as

$$\alpha_i = w_i A_i \beta_i \Rightarrow w_i A_{i+1} \beta_i \Rightarrow$$

$$\cdots \Rightarrow w_i A_j \beta_i \Rightarrow w_i \gamma \Rightarrow \alpha_{i+1}$$

when $A_j \in W(A_i)$ and $A_j \to \alpha$ is a non-unit production.

Hence $A_j \to \gamma$ is a production in $P_1$

$\therefore \alpha_j \overset{*}{\underset{G_1}{\Rightarrow}} \alpha_{j+1}$ and we have $S \overset{*}{\underset{G_1}{\Rightarrow}} \alpha_{j+1}$

Repeating the arguments whenever some unit production occurs in the remaining part of the derivation

$$S \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_i \Rightarrow \alpha_{i+1} \Rightarrow \cdots \Rightarrow \alpha_j \Rightarrow \alpha_{j+1} \Rightarrow \cdots \Rightarrow \alpha_n = w$$

We can prove that $S \overset{*}{\underset{G_1}{\Rightarrow}} \alpha_n = w$

Hence $L(G') \subseteq L(G_1)$

## Corollary

If $G$ is a CFG, we can construct an equivalent grammar $G'$ which is reduced (i.e. has no useless symbols) and has no null productions or unit productions

## Proof:

We construct $G'$ in the following way:

Algorithm 6
- Eliminate null productions to get $G_1$
- Eliminate unit productions to get $G_2$

Construction 6
- Construct a reduced grammar $G'$ equivalent to $G_2$

$G'$ is the required equivalent grammar.

**Note:**

We have to apply the constructions only in the order given above. Otherwise we maynot get the grammar in the most simplified form.