***

Elimination of Left Recursion

A grammar is left recursive if it has a nonterminal A. Such that there is a derivation $A \rightarrow A\alpha$ for some string α.
Top-down parsing method cannot handle left-recursive grammars, so a transformation is needed to eliminate left recursion.

The left-recursion pair of production

$$A \rightarrow A\alpha \mid \beta$$

Could be replaced by the non-left recursive productions.

$$A \rightarrow \beta A'$$
$$A' \rightarrow \alpha A' \mid \epsilon$$

Immediate left recursion can be eliminated by the following techniques, which works for any number of A-productions.

First, group the production as

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \cdots\cdots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \cdots\cdots \mid \beta_n$$

Where no β_i begins with an A. Then, replace the A-production by.

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \cdots\cdots \mid \beta_n A'$$
$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \cdots \mid \alpha_m A' \mid \epsilon$$

**
*

Algorithm: Eliminating Left Recursion

Input : Grammar 'G' with no cycles or ϵ production

Output : An equivalent grammar with no left recursion

Method :

1> Arrange the non-terminal in some order $A_1, A_2, \cdots\cdots A_n$

2> for (each i from 1 to n)

{

3) for (each j from 1 to i - 1)

{

4] Replace each production of the form $A_i \rightarrow A_j \gamma$ by the Production $A_i \rightarrow \delta_1\gamma \mid \delta_2\gamma \text{ ------ } \mid \delta_k\gamma$, where $A_j \rightarrow \delta_1 \mid \delta_2 \text{ --- } \mid \delta_k$ are all current A_j productions

5] }

6] Eliminate the immediate left recursion among the A_i productions

7] }

Left factoring :-

Left factoring is a grammar transformation that is useful for producing a grammar suitable for predictive or top-down parsing. When the choice between two alternative A production is not clear we may be able to rewrite the productions to ~~def~~ refer the decision until enough of the input has been seen that we can make the right choice

For Eg, if we have the 2 productions

Stmt → if Expr then Stmt else Stmt
| if Expr then Stmt

On seeing the input if we cannot immediatley tell which production to choose to expand Stmt. In general,

$$\text{if } A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$$

are two A-productions and the input begins with a non-empty string derived from α

LEFT RECURSION PROBLEMS

1. Eliminate left recursion from the following grammar.

(a) $E \rightarrow E+T / T$

(b) $T \rightarrow T*F | F$

(c) $F \rightarrow (E) | id$

Solution:-

To Eliminate left recursion we have a rule,

$$\boxed{A \rightarrow A\alpha | \beta}$$

This rule can be Converted in to:

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \\ A' \rightarrow \epsilon \end{array}}$$

(a) consider the grammar,

$$E \rightarrow E+T / T$$

Map this grammar with the rule $A \rightarrow A\alpha | \beta$

$$\boxed{A \rightarrow A\alpha | \beta}$$

$$\underbrace{E}_{A} \rightarrow \underbrace{E}_{A}\underbrace{+T}_{\alpha} | \underbrace{T}_{\beta}$$

$$\boxed{A \rightarrow \beta A'}$$

$$E \rightarrow TE'$$

$$\boxed{A' \rightarrow \alpha A' | \epsilon}$$

$$E' \rightarrow +TE' | \epsilon$$

∴ The grammar without left recursion will be

$$\boxed{\begin{array}{l} E \rightarrow TE' \\ E' \rightarrow +TE' | \epsilon \end{array}}$$

(b) consider the grammar

$$T \rightarrow T * F \mid F$$

Map this grammar with the rule $A \rightarrow A\alpha \mid \beta$

$$A \rightarrow A\alpha \mid \beta$$

$$\underbrace{T}_{A} \rightarrow \underbrace{T}_{A} \underbrace{* F}_{\alpha} \mid \underbrace{F}_{\beta}$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \epsilon$$

∴ The grammar without left recursion is

$$\boxed{\begin{array}{l} T \rightarrow FT' \\ T' \rightarrow *FT' \end{array}}$$

(c) consider the grammar,

$$F \rightarrow (E) \mid id$$

There is no left recursion in the above grammar.

2) Consider the following grammar,

$A \rightarrow ABd \mid Aa \mid a$

$B \rightarrow Be \mid b$

Remove left recursion.

= Solution:-

Rewrite the following grammar as follows

$A \rightarrow ABd \mid a$

$A \rightarrow Aa \mid a$

$B \rightarrow Be \mid b$

To Eliminate left recursion we have the rule

$\boxed{A \rightarrow A\alpha \mid \beta}$

This rule can be converted in to

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}}$$

Consider the grammar,

$A \rightarrow ABd \mid a$

Map this grammar with the rule $A \rightarrow A\alpha \mid \beta$

$A \rightarrow A\alpha \mid \beta$

$\underbrace{A}_{A} \rightarrow \underbrace{A}_{A}\underbrace{Bd}_{\alpha} \mid \underbrace{a}_{\beta}$

$A \rightarrow aA'$

$A' \rightarrow BdA' \mid \epsilon$

Consider the grammar,

$A \rightarrow Aa \mid a$

Map this grammar with the rule $A \rightarrow A\alpha \mid \beta$

$A \rightarrow A\alpha \mid \beta$

$\underbrace{A}_{A} \rightarrow \underbrace{A}_{A}\underbrace{a}_{\alpha} \mid \underbrace{a}_{\beta}$

$A \rightarrow aA'$

$A' \rightarrow aA' \mid \epsilon$

consider the grammar

$B \to Be|b$

Map this grammar with the rule $A \to A\alpha|\beta$

$A \to A\alpha|\beta$

$\underbrace{B}_{A} \to \underbrace{B}_{A}\underbrace{e}_{\alpha}|\underbrace{b}_{\beta}$

$B \to bB'$

$B' \to eB'|\epsilon$

∴ The grammar without left recursion will be

$$\boxed{\begin{array}{l} A \to aA' \\ A' \to BdA'|aA'|\epsilon \\ B \to bB' \\ B' \to eB'|\epsilon \end{array}}$$

3] Eliminate left recursion from the following grammar.

$S \to aB|ac|Sd|Se$

$B \to bBc|f$

$C \to g$

= Solution:-

Rewrite this grammar as follows:

$S \to Sd|aB$

$S \to Se|ac$

$B \to bBc|f$

$C \to g$

The grammar with left recursion can be eliminated using the rule, $\boxed{A \to A\alpha|\beta}$

This can be converted into:

$$\boxed{\begin{array}{l} A \to \beta A' \\ A' \to \alpha A'|\epsilon \end{array}}$$

Consider the grammar,

$S \to Sd \mid aB$

Map this grammar with the rule $A \to A\alpha \mid \beta$

$A \to A\alpha \mid \beta$

$\underbrace{S}_{A} \to \underbrace{S}_{A}\underbrace{d}_{\alpha} \mid \underbrace{aB}_{\beta}$

$S \to aBS'$

$S' \to dS' \mid \epsilon$

Consider the grammar,

$S \to Se \mid ac$

Map this grammar with the rule $A \to A\alpha \mid \beta$

$A \to A\alpha \mid \beta$

$\underbrace{S}_{A} \to \underbrace{S}_{A}\underbrace{e}_{\alpha} \mid \underbrace{ac}_{\beta}$

$S \to acS'$

$S' \to eS' \mid \epsilon$

Consider the grammar

$B \to bBc \mid f$

There is no left recursion in the above grammar

Consider the grammar,

$C \to g$

There is no left recursion in the above grammar.

$\therefore$ The grammar without left recursion is,

$$\boxed{\begin{array}{l} S \to aBS' \mid acS' \\ S' \to dS' \mid eS' \mid \epsilon \\ B \to bBc \mid f \\ C \to g \end{array}}$$

4] Eliminate the left recursion from the following grammar

$A \rightarrow Ac \mid Aad \mid bd \mid c$

Solution:-

Rewrite the grammar as follows:

$A \rightarrow Ac \mid bd$

$A \rightarrow Aad \mid c$

To Eliminate left recursion we have a rule,

$\boxed{A \rightarrow A\alpha \mid \beta}$

This rule can be converted into.

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}}$$

Consider the grammar.

$A \rightarrow Ac \mid bd$

Map this grammar with the rule, $A \rightarrow A\alpha \mid \beta$

$A \rightarrow A\alpha \mid \beta$

$\underbrace{A}_{A} \rightarrow \underbrace{A}_{A}\underbrace{c}_{\alpha} \mid \underbrace{bd}_{\beta}$

$A \rightarrow bdA'$

$A' \rightarrow cA' \mid \epsilon$

Consider the grammar,

$A \rightarrow Aad \mid c$

Map this grammar with the rule $A \rightarrow A\alpha \mid \beta$

$A \rightarrow A\alpha \mid \beta$

$\underbrace{A}_{A} \rightarrow \underbrace{A}_{A}\underbrace{ad}_{\alpha} \mid \underbrace{c}_{\beta}$

$A \rightarrow cA'$

$A' \rightarrow adA' \mid \epsilon$

∴ The grammar without left recursion is,

$$\boxed{\begin{array}{l} A \rightarrow bdA' \mid cA' \\ A' \rightarrow cA' \mid adA' \mid \epsilon \end{array}}$$

5] Eliminate left recursion from the following grammar

$L \rightarrow L,S \mid S$

$S \rightarrow a \mid (L)$

Solution:-

To eliminate left recursion we have the rule,

$$\boxed{A \rightarrow A\alpha \mid \beta}$$

This rule can be converted into

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}}$$

Consider the grammar

$L \rightarrow L,S \mid S$

Map this grammar with the rule

$A \rightarrow A\alpha \mid \beta$

$\underbrace{L}_{A} \rightarrow \underbrace{L}_{A}\underbrace{,S}_{\alpha} \mid \underbrace{S}_{\beta}$

$L \rightarrow SL'$

$L' \rightarrow ,SL' \mid \epsilon$

Consider the grammar $S \rightarrow a \mid (L)$

There is no left recursion in the above grammar

$\therefore$ The grammar without left recursion is,

$$\boxed{\begin{array}{l} L \rightarrow SL' \\ L' \rightarrow ,SL' \mid \epsilon \\ S \rightarrow a \mid (L) \end{array}}$$

6] Eliminate Left recursion from the following grammar

$S \rightarrow a \mid \uparrow \mid (T)$

$T \rightarrow T, S \mid S$

= Solution:

To eliminate left recursion we have the rule

$$\boxed{A \rightarrow A\alpha \mid \beta}$$

This rule can be converted into:

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}}$$

consider the grammar,

$T \rightarrow T, S \mid S$

map this grammar with the rule $A \rightarrow A\alpha \mid \beta$

$A \rightarrow A\alpha \mid \beta$

$$\underbrace{T}_{A} \rightarrow \underbrace{T}_{A} \underbrace{,S}_{\alpha} \mid \underbrace{S}_{\beta}$$

$T \rightarrow ST'$

$T' \rightarrow , ST' \mid \epsilon$

Consider the grammar,

$S \rightarrow a \mid \uparrow \mid (T)$

There is no Left recursion in the following grammar

∴ The grammar without left recursion is

$$\boxed{\begin{array}{l} S \rightarrow a \mid \uparrow \mid (T) \\ T \rightarrow ST' \\ T' \rightarrow , ST' \mid \epsilon \end{array}}$$

LEFT FACTORING PROBLEMS

1. Do the left factoring for the following grammar

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Solution:

To eliminate the left factoring we have the rule,

$$\boxed{\begin{array}{l} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \cdots \mid \alpha\beta_n \mid \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \end{array}}$$

Consider the grammar,

$$\underbrace{S}_{A} \rightarrow \underbrace{iEtS}_{\alpha}\underbrace{\epsilon}_{\beta_1} \mid \underbrace{iEtS}_{\alpha}\underbrace{eS}_{\beta_2} \mid a$$

map this grammar using the rule, $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \cdots \alpha\beta_n \mid$

$S \rightarrow iEtSS'$

$S' \rightarrow eS \mid \epsilon$

Consider the grammar,

$E \rightarrow b$

The grammar doesnot contain left factoring

∴ The grammar with left factoring is

$$\boxed{\begin{array}{l} S \rightarrow iEtSS' \mid a \\ S' \rightarrow eS \mid \epsilon \\ E \rightarrow b \end{array}}$$

2. Do the left factoring in the following grammar

$A \rightarrow aAB \mid aA \mid a$

$B \rightarrow bB \mid b$

Solution:-

For left factoring we have the rule,

$$\boxed{\begin{array}{l} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \cdots \alpha\beta_n \mid \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \end{array}}$$

Consider the grammar,

$A \rightarrow aAB \mid aA \mid a$

map this grammar with the rule $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$

$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$

$A \rightarrow \underbrace{a}_{\alpha}\underbrace{AB}_{\beta_1} \mid \underbrace{a}_{\alpha}\underbrace{A}_{\beta_2} \mid \underbrace{a}_{\alpha}\underbrace{\epsilon}_{\beta_3}$

$A \rightarrow aA'$

$A' \rightarrow AB \mid A \mid \epsilon$

Consider the grammar,

$B \rightarrow bB \mid b$

map this grammar with the rule $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$

$B \rightarrow \underbrace{b}_{\alpha}\underbrace{B}_{\beta_1} \mid \underbrace{b}_{\alpha}\underbrace{\epsilon}_{\beta_2}$

$B \rightarrow bB'$

$B' \rightarrow B \mid \epsilon$

∴ The grammar with left factoring is

$$\boxed{\begin{array}{l} A \rightarrow aA' \\ A' \rightarrow AB \mid A \mid \epsilon \\ B \rightarrow bB' \\ B' \rightarrow B \mid \epsilon \end{array}}$$