Lecture - 17 22/05/20 ①

Simplification of Grammars

- Even through there is no restriction on the RHS of a production for any CFG, it is better, in fact necessary to eliminate some of the useless symbols and productions.
- In any CFG, some of the symbols or productions may not be used to derive a string.
- Some symbols and productions may never be used while deriving a string
- So these symbols and productions which will never be used are useless and the corresponding productions can be eliminated.

e.g;
$$S \to aA \mid B$$
$$A \to aA \mid a$$

If we use $S \to B$, a string can never be derived $\Rightarrow$ B is useless as a symbol and $S \to B$ is useless as a production. $\Longrightarrow$ need to eliminat

How to eliminate ?

1. symbols in V from which string of terminals cannot be derived.
2. symbols in (V ∪ T) and not appearing in any sentential form
3. ε - productions
4. The productions of the form $A \rightarrow B$ i.e, unit productions.

Substitution (A non-terminal is replaced by the corresponding symbol on the RHS.

Let

$G = (V, T, P, S)$ be a CFG.

Consider the production

$A \rightarrow x_1 B x_2$

$B \rightarrow y_1 | y_2 | y_3 | \cdots y_n$.

The production $A \rightarrow x_1 B x_2$ can be replaced by

$A \rightarrow x_1 y_1 x_2 | x_1 y_2 x_2 | \cdots x_1 y_n x_2$

and the production

$B \rightarrow y_1 | y_2 | \cdots y_n$

can be deleted.

Resulting productions are added to P_1 and variables to V_1.

The language generated by the resulting grammar $G_1 = (V_1, T, P_1, S)$ is same as that generated by G.

i.e, $L(G_1) = L(G)$

Ex: Consider the productions

$A \rightarrow aBa$

$B \rightarrow ab/b$

Simplify the Grammar.

Sol: Consider the production

$A \rightarrow aBa$

- RHS contains a non-terminal B.
- B can be replaced by $B \rightarrow ab/b$.

$\therefore$ resulting A-productions are

$A \rightarrow aaba \mid aba$

~~Left~~ Left Recursion

A grammar G is said to be left recursive if it has non-terminal A such that there is a derivative of the form

$A \overset{+}{\Longrightarrow} A\alpha$ (by appling one or more productions)

where α is a string of terminals and non-terminals.

As 'A' appears again on the LHS of the production, G is said to be left recursive

Left recursion is of two types < Immediate / Indirect (4)

Immediate Left Recursion

$A \rightarrow A\alpha$

Consider
$E \rightarrow E+T \mid T$ — (i)
$T \rightarrow T*F \mid F$ — (ii)
$F \rightarrow (E) \mid id$ — (iii)

In (i) and (ii), the first symbol on the RHS is same as the symbol on the LHS $\Rightarrow$ immediate left recursion.

Indirect Left Recursion

A left recursion involving derivations of two or more steps so that the first symbol on the RHS of the partial derivation is same as the symbol from which the derivation started is called indirect left recursion.

$E \rightarrow T$
$T \rightarrow F$
$F \rightarrow E+T \mid id$

$\Downarrow$

$E \Rightarrow T \Rightarrow F \Rightarrow E+T$] $\Rightarrow$ left recursive.

same

Left Recursion :⇒ substitution method

If a Grammar has left recursion, then it is not desirable, as the parser constructed will enter an infinite loop and system may crash.
⇒ left recursion must be eliminated

Def. A grammar G is said to be left recursive if there is some non-terminal A such that

$$A \rightarrow A\alpha$$

Elimination

Consider the A- production of the form

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid A\alpha_3 \; \cdots \; A\alpha_m \mid \beta_1 \mid \beta_2 \cdots \mid \beta_n$$

where β_i's do not start with A.

Then A production can be replaced by

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \beta_3 A' \mid \cdots \beta_n A'$$

$$A' = \alpha_1 A' \mid \alpha_2 A' \mid \alpha_3 A' \mid \cdots \mid \alpha_m A' \mid \epsilon$$

α_i's do not start with A'

Q)

$S \rightarrow Ab \mid a$ — ①

$A \rightarrow Ab \mid Sa$ — ②

Elimination of left Recursion ⑥

Sol)

$S \rightarrow Ab \Rightarrow Sab$

$S \Rightarrow Sab \rightarrow$ left recursion.

$S \rightarrow Ab \mid a$

$A \rightarrow Ab \mid Aba \mid aa$

Given $A \rightarrow A\alpha \mid \beta$	Substitution	Without left recursion
	α	$A \rightarrow \beta_1 A'$ & $A' = \alpha_1 A' \mid \varepsilon$
$S \rightarrow Ab \mid a$		$S \rightarrow Ab \mid a$
$A \rightarrow Ab \mid Aba \mid aa$	$A = A$	$A \rightarrow aaA'$
	$\alpha_1 = b$	
	$\alpha_2 = ba$	$A' = \alpha_1 A' \mid \alpha_2 A' \mid \varepsilon$
	$\beta_1 = aa$	$= bA' \mid baA' \mid \varepsilon$

After elimination of left recursion

$S \rightarrow Ab \mid a$

$A \rightarrow aaA'$

$A' \rightarrow bA' \mid baA' \mid \varepsilon$

Q/ Eliminate left recursion from the grammar.

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid id$

Sol.

Given $A \rightarrow A\alpha_i \mid \beta_i$	Substitution	Without left recursion, $A \rightarrow \beta_i A'$ and $A' \rightarrow \alpha_1 A' \mid \varepsilon$
$E \rightarrow E+T \mid T$	$A = E$ $\alpha_1 = +T$ $\beta_1 = T$	$E \rightarrow TE'$ $E' \rightarrow +TE' \mid \varepsilon$
$T \rightarrow T*F \mid F$	$A = T$ $\alpha_1 = *F$ $\beta_1 = F$	$T \rightarrow FT'$ $T' \rightarrow *FT' \mid \varepsilon$
$F \rightarrow (E) \mid id$	Not applicable	$F \rightarrow (E) \mid id$

Grammar obtained is

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \varepsilon$

$T \rightarrow FT'$

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

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