

# TUTORIAL 9

UI9CS012

1. Do Left factoring in the following grammar:

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

$$B \rightarrow bB \mid b$$

$$\Rightarrow A \rightarrow aA'$$

$$A' \rightarrow AA'' \mid \epsilon$$

$$A'' \rightarrow B \mid \epsilon$$

$$B \rightarrow bB'$$

$$B' \rightarrow B \mid \epsilon$$

$$ii) A \rightarrow aAB \mid aBc \mid aAc$$

$$A \rightarrow aA'$$

$$A' \rightarrow AA'' \mid BC$$

$$A'' \rightarrow B \mid C$$

2. Explain with examples : Recursive Descent Parsing

① A parser that uses a set of recursive procedure to recognize its inputs with no backtracking is called recursive descent parser.

②

For Implementing a recursive descent parser

For a grammar, it must be Left Factored  
and must not be Left recursive.

U19CS012

Example Grammar

$E \rightarrow TE'$

$E' \rightarrow +TE'$

$T \rightarrow FT'$

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

$F \rightarrow (E) \mid id$

Solution - Procedure EC() {

TC();

E'();

}

Procedure E'() {

if input symbol = '+' then

advance();

TC();

E'();

}

Procedure TC() {

EC();

T'();

}

Procedure T'() {

if input symbol = '\*' then

advance();

EC();

T'();

}

Procedure F() {

if input symbol = 'id' then

advance();

else if input symbol = '(' then

advance();

EC();

else if input symbol = ')' then

advance();

else

error;

}



U19CS015

3.7 Calculate First and Follow for the given grammar.

- 1.)
- $$E \rightarrow TE'$$
- $$E' \rightarrow +TE' \mid \epsilon$$
- $$T \rightarrow FT'$$
- $$T' \rightarrow *FT' \mid \epsilon$$
- $$F \rightarrow (E) \mid id$$

SYMBOL	FIRST	FOLLOW
E	{ '(', id }	{ '\$', ')' }
E'	{ '+', '\epsilon' }	{ '\$', ')' }
T	{ '(', 'id' }	{ '\$', ')', '+' }
T'	{ '*', '\epsilon' }	{ '\$', ')', '+' }
F	{ '(', 'id' }	{ '\$', ')', '+', '*' }

- 2.7
- $$S \rightarrow ACB \mid Cbb \mid Ba$$
- $$A \rightarrow da \mid BC$$
- $$B \rightarrow g \mid \epsilon$$
- $$C \rightarrow h \mid \epsilon$$

SYMBOL	FIRST	FOLLOW
S	{ d, g, h, \epsilon, a, b }	{ '\$' }
A	{ d, h, g, \epsilon }	{ '\$', g, h }
B	{ g, \epsilon }	{ '\$', a, g, h }
C	{ h, \epsilon }	{ '\$', b, g, h }

Next Page →

DI9CSQ12

3. >  $S \rightarrow A$  Remove Left recursion

$A \rightarrow aB \mid Ad$   $S \rightarrow A$   $A \rightarrow aBA' \mid d$   $C \rightarrow g$

$B \rightarrow b$   $B \rightarrow b$   $A' \rightarrow dA' \mid \epsilon$

$C \rightarrow g$

SYMBOL	FIRST	FOLLOW
S	{a}	{g}
A	{a}	{g}
B	{b}	{g}
C	{g}	-
A'	{d, \epsilon}	{g}

4. >  $S \rightarrow (L) \mid a$

$L \rightarrow SL'$

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

SYMBOL	FIRST	FOLLOW
S	{ '(', a' }	{ '(', ')', 'a' }
L	{ '(', 'a' }	{ ')', 'a' }
L'	{ ')', '\epsilon' }	{ 'a' }

5. >  $S \rightarrow AaAb \mid BbBa$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

SYMBOL	FIRST	FOLLOW
S	{a, b}	{a, b}
A	{\epsilon}	{a, b}
B	{\epsilon}	{a, b}