

FIRST & FOLLOW

Course Name: Compiler Design

Course Code: CSE331

Level:3, Term:3

Department of Computer Science and Engineering

Daffodil International University

Rules of FIRST:

1. If x is a terminal, then $\text{FIRST}(x) = \{ 'x' \}$
2. If $x \rightarrow \epsilon$, is a production rule, then add ϵ to $\text{FIRST}(x)$.
3. If $X \rightarrow Y_1 Y_2 Y_3 \dots Y_n$ is a production,
 - a. $\text{FIRST}(X) = \text{FIRST}(Y_1)$
 - b. If $\text{FIRST}(Y_1)$ contains ϵ then $\text{FIRST}(X) = \{ \text{FIRST}(Y_1) - \epsilon \} \cup \{ \text{FIRST}(Y_2) \}$
 - c. If $\text{FIRST}(Y_i)$ contains ϵ for all $i = 1$ to n , then add ϵ to $\text{FIRST}(X)$.

Rules of FOLLOW:

- 1) $\text{FOLLOW}(S) = \{ \$ \}$, where S is the starting Non-Terminal
- 2) If $A \rightarrow pBq$ is a production, where p , B and q are any grammar symbols, then everything in $\text{FIRST}(q)$ except ϵ is in $\text{FOLLOW}(B)$.
- 3) If $A \rightarrow pB$ is a production, then everything in $\text{FOLLOW}(A)$ is in $\text{FOLLOW}(B)$.
- 4) If $A \rightarrow pBq$ is a production and $\text{FIRST}(q)$ contains ϵ , then $\text{FOLLOW}(B)$ contains $\{ \text{FIRST}(q) - \epsilon \} \cup \text{FOLLOW}(A)$

Terminal & Non-terminal

- All the capital letters are non-terminal (A,B,C,D etc).
- Except capital letters, all other characters and operators are terminal (*,+,-,(,),a,b,c,d etc).

Example:

Production rules are given: $E \rightarrow TE'$

$$E' \rightarrow +T E' \mid \epsilon$$
$$T \rightarrow F T'$$
$$T' \rightarrow *F T' \mid \epsilon$$
$$F \rightarrow \text{id} \mid (E)$$

Now find sets of FIRST & FOLLOW

FIRST:

Production	FIRST
$E \rightarrow TE'$	$\{ (, id \}$
$E' \rightarrow +T E' \mid \epsilon$	$\{ +, \epsilon \}$
$T \rightarrow F T'$	$\{ (, id \}$
$T' \rightarrow *F T' \mid \epsilon$	$\{ *, \epsilon \}$
$F \rightarrow id \mid (E)$	$\{ id, (\}$

FOLLOW:

Production	FIRST	FOLLOW
$E \rightarrow TE'$	{ (, id }	{ \$,) }
$E' \rightarrow +T E' \mid \epsilon$	{ +, ϵ }	{ \$,) }
$T \rightarrow F T'$	{ (, id }	{ + , \$,) }
$T' \rightarrow *F T' \mid \epsilon$	{ *, ϵ }	{ + , \$,) }
$F \rightarrow id \mid (E)$	{ id, (}	{ *, +, \$,) }

LL(1) Parsing

A top-down parser builds the parse tree from the top down, starting with the non-terminal.

There are two types of Top Down Parsers:

1. Top Down Parser with Backtracking
2. Top Down Parsers without Backtracking

Top Down Parsers without Backtracking can further be divided into two parts:

1. Recursive Descent
2. Non-Recursive Descent

Non-Recursive Descent which is also known as LL(1) Parser.

LL (1) Parsing:

Here the 1st L represents that the scanning of the Input will be done from Left to Right manner and 2nd L shows that in this Parsing technique we are going to use Left most Derivation Tree. and finally the **1** represents the number of look ahead, means how many symbols are you going to see when you want to make a decision.

Note: The knowledge of Top-down parser, FIRST, FOLLOW is required.

Construction of LL(1) Parsing Table

To construct the Parsing table, we have two functions: FIRST(), FOLLOW()

Now, after computing the First and Follow set for each Non-Terminal symbol we have to construct the Parsing table. In the table Rows will contain the Non-Terminals and the column will contain the Terminal Symbols.

All the Null Productions of the Grammars will go under the Follow elements and the remaining productions will lie under the elements of First set.

Example:

Production	FIRST	FOLLOW
$E \rightarrow TE'$	{ (, id }	{ \$,) }
$E' \rightarrow +TE' \mid \epsilon$	{ + , ϵ }	{ \$,) }
$T \rightarrow FT'$	{ (, id }	{ + , \$,) }
$T' \rightarrow *FT' \mid \epsilon$	{ * , ϵ }	{ + , \$,) }
$F \rightarrow id \mid (E)$	{ id , (}	{ * , + , \$,) }

Constructing the LL(1) parsing table:

Non-terminal	id	*	+	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'			$E' \rightarrow +TE'$		$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow *FT'$	$T' \rightarrow \epsilon$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

When ϵ is found in any production, then it needs to make union of FIRST & FOLLOW sets of that particular production.

More Exercises:

Production rules are given: $S \rightarrow Bb \mid cd$

$B \rightarrow aB \mid \epsilon$

$C \rightarrow cC \mid \epsilon$

Now find sets of FIRST & FOLLOW

More Exercises:

Production rules are given:

$S \rightarrow ABCDE$

$A \rightarrow a \mid \epsilon$

$B \rightarrow b \mid \epsilon$

$C \rightarrow c$

$D \rightarrow d \mid \epsilon$

$E \rightarrow e \mid \epsilon$

Now find sets of FIRST & FOLLOW

Thank You