Aditi Bhatia

D12A

06

SPCC Lab 03

AIM: Implementation of first and follow in compiler design

THEORY:

FIRST and FOLLOW are two functions associated with grammar that helps us fill in the entries of an M-table

FIRST()

It is a function that gives the set of terminals that begin the strings derived from the production rule

A symbol c is in FIRST(α) if and only if $\alpha \Rightarrow c\beta$ for some sequence β of grammar symbols

A terminal symbol a is in FOLLOW (N) if and only if there is a derivation from the start symbols of the grammar such that $S \Rightarrow \alpha N \alpha \beta$ where α and β are a sequence of grammar symbols

In other words a terminal c is in FOLLOW(N) if c can follow N at some point in a derivation

Benifit of FIRST and FOLLOW

→ It can be used to prove the LL(K) characteristic of grammar

→ It can be used to promote in the construction of predictive parsing tables

→ It provides selection information for recursive descent parsers

If the input string is $T \rightarrow *FT' | \epsilon$

Here we find out that T has two productions $T \rightarrow *FT'$ and $T \rightarrow \epsilon$

After viewing this we found out that first of T in both the production statements is * and ε

∴ First of the string is {*, ε}

FOR EDUCATIONAL USE

Rules to find First( )

To find the first( ) of the grammar symbol then we have to apply the following set of rules to the given grammar :-

- If X is a terminal, then First(X) is {X}
- If X is a non terminal and X tends to aa is production then add 'a' to the first of X. if X → ε add null to the First(X)
- If X → YZ then if FIRST(Y) = ε then First(X) = { First(Y) - ε } ∪ First(Z)
- If X → YZ then if First(X) = Y then First(Y) = terminal but null then First(X) = First(Y) = terminals

If the input string is E → TE', F → (E) | id

Here we found that on the right hand side of the production statement where E occurs, we found E in the production F → (E) | id through which we can find follow of E

Follow of E = {) }

Rules to find Follow( ):

- $ is a follow of S (start symbol)
- If A → aB B != ε then first(β) is in follow(B)
- If A → aB or A → aBβ where first(β) = ε then everything in follow(A) is in follow(B)

For Eg: For the grammar

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid id$

find out the First and Follow

Answer :

Remove left recursion

i.e $A \rightarrow A\alpha | \beta \Rightarrow A \rightarrow \beta A'$

$A' \rightarrow \alpha A' | \varepsilon$

$\therefore$ $E \rightarrow TE'$

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

$T \rightarrow FT'$

$T \rightarrow \varepsilon | *FT'$

$\therefore$ FIRST and FOLLOW of above production is :

	FIRST SET	FOLLOW SET
$E \rightarrow TE'$	{ (, id }	{ $,) }
$E' \rightarrow +TE' \| \varepsilon$	{ +, ε }	{ $,) }
$T \rightarrow T'$	{ (, id }	{ +, $) }
$T \rightarrow *FT' \| \varepsilon$	{ *, ε }	{ + $) }
$F \rightarrow (E) \| id$	{ (, id }	{ * + $) }

conclusion :

FIRST SET is a concept used in syntax analysis specifically in the content of LL and LR parsing algorithms. It is a set of terminals that can appear immediately after a given non terminal in a grammar. FOLLOW set is a set of terminals that can appear immediately after a given non terminal in a grammar.
It is used to construct predictive parsing tables and is used to prove LL(K) characteristic of grammars.