Compiler Design Lab

NAME: Priyanka Srinivas
REGISTER NUMBER: RA1911026010014
SECTION: K1

Exercise 5: FIRST and FOLLOW

AIM: To compute the FIRST and FOLLOW of a grammar

INTRODUCTION:

FIRST and FOLLOW are two functions associated with grammar that help 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 α ⇒ cβ 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 symbol S of the grammar such that S ⇒ αNαβ, where α and β are a (possibly empty) 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.

Computation of FIRST

FIRST (α) is defined as the collection of terminal symbols which are the first letters of strings derived from α.
FIRST (α) = {α |α →∗ αβ for some string β }
If X is Grammar Symbol, then First (X) will be −

- If X is a terminal symbol, then FIRST(X) = {X}
- If X → ε, then FIRST(X) = {ε}
- If X is non-terminal & X → a α, then FIRST (X) = {a}
- If X → Y1, Y2, Y3, then FIRST (X) will be

(a) If Y is terminal, then
FIRST (X) = FIRST (Y1, Y2, Y3) = {Y1}
(b) If Y1 is Non-terminal and
If Y1 does not derive to an empty string i.e., If FIRST (Y1) does not contain ε then, FIRST (X) = FIRST (Y1, Y2, Y3) = FIRST(Y1)
(c) If FIRST (Y1) contains ε, then.
FIRST (X) = FIRST (Y1, Y2, Y3) = FIRST(Y1) − {ε} ∪ FIRST(Y2, Y3)
Similarly, FIRST (Y2, Y3) = {Y2}, If Y2 is terminal otherwise if Y2 is Non-terminal then

- FIRST (Y2, Y3) = FIRST (Y2), if FIRST (Y2) does not contain ε.
- If FIRST (Y2) contain ε, then
- FIRST (Y2, Y3) = FIRST (Y2) − {ε} ∪ FIRST (Y3)

Similarly, this method will be repeated for further Grammar symbols, i.e., for Y4, Y5, Y6 … . YK.
Computation of FOLLOW
Follow (A) is defined as the collection of terminal symbols that occur directly to the right of A.
FOLLOW(A) = {a|S ⇒* αAaβ where α, β can be any strings}
Rules to find FOLLOW

- If S is the start symbol, FOLLOW (S) ={$}
- If production is of form A → α B β, β ≠ ε.

(a) If FIRST (β) does not contain ε then, FOLLOW (B) = {FIRST (β)}
Or
(b) If FIRST (β) contains ε (i. e., β ⇒* ε), then
FOLLOW (B) = FIRST (β) − {ε} ∪ FOLLOW (A)
∵ when β derives ε, then terminal after A will follow B.

- If production is of form A → αB, then Follow (B) ={FOLLOW (A)}.

ALGORITHM FOR FIRST:

1. If X is a terminal then FIRST(X) = {X}
   Example: F -> I | id We can write it as FIRST(F) -> { (, id)
2. If X is a non-terminal like E -> T then to get FIRST substitute T with other productions until you get a terminal as the first symbol.
3. If X -> ε then add ε to FIRST(X).

ALGORITHM FOR FOLLOW:

1. Always check the right side of the productions for a non-terminal, whose FOLLOW set is being found. (never see the left side).
2. (a) If that non-terminal (S,A,B…) is followed by any terminal (a,b…,*,+,(,)…) , then add that terminal into the FOLLOW set.
   (b) If that non-terminal is followed by any other non-terminal then add FIRST of other non-terminal into the FOLLOW set.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int n,m=0,p,i=0,j=0;
char a[10][10],f[10];
void follow(char c);
void first(char c);
int main(){
int i,z;
```

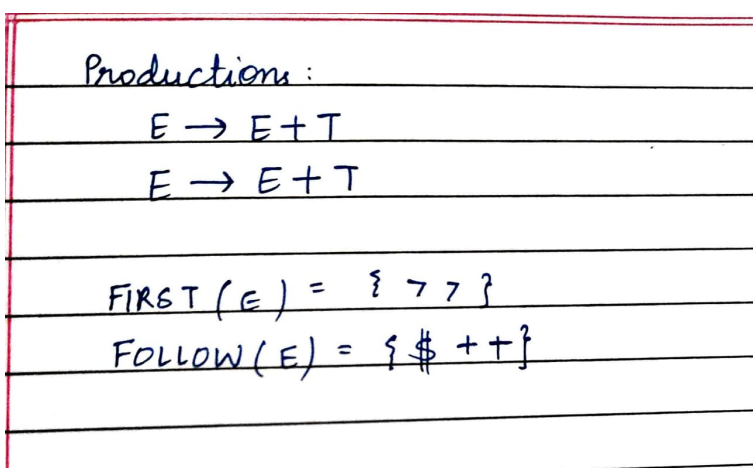
```
char c,ch;
printf("Enter the no of prooductions:\n");
scanf("%d",&n);
printf("Enter the productions:\n");
for(i=0;i<n;i++)
scanf("%s%c",a[i],&ch);
do{
m=0;
printf("Enter the elemets whose fisrt & follow is to be found:");
scanf("%c",&c);
first(c);
printf("First(%c)={",c);
for(i=0;i<m;i++)
printf("%c",f[i]);
printf("}\n");
strcpy(f," ");
m=0;
follow(c);
printf("Follow(%c)={",c);
for(i=0;i<m;i++)
printf("%c",f[i]);
printf("}\n");
printf("Continue(0/1)?");
scanf("%d%c",&z,&ch);
}while(z==1);
return(0);
}
void first(char c){
int k;
if(!isupper(c))
f[m++]=c;
for(k=0;k<n;k++){
if(a[k][0]==c){
if(a[k][2]=='$')
follow(a[k][0]);
else if(islower(a[k][2]))
f[m++]=a[k][2];
else first(a[k][2]);
}
```

```
}
}
void follow(char c){
if(a[0][0]==c)
f[m++]='$';
for(i=0;i<n;i++){
for(j=2;j<strlen(a[i]);j++){
if(a[i][j]==c){
if(a[i][j+1]!='\0')
first(a[i][j+1]);
if(a[i][j+1]=='\0' && c!=a[i][0])
follow(a[i][0]);
}
}
}
}
```

INPUT:

E->E+T
E->E+T

Manual Output:



System Output:

```
Command Prompt - ff

C:\Users\priya\Desktop\Coding\compiler-design-lab\lab-5>gcc firstfollow.c -o ff.exe

C:\Users\priya\Desktop\Coding\compiler-design-lab\lab-5>ff
Enter the no of prooductions:
2
Enter the productions:
E->E+T
E->E+T
Enter the elemets whose fisrt & follow is to be found:E
First(E)={>>}
Follow(E)={$++}
Continue(0/1)?
```

RESULT:

FIRST and FOLLOW for a set of productions has been computed.