College Of Engineering Trivandrum

Compiler Design Lab

CS431

Submitted By:
Neethu S
S6 CSE Roll No:42

TVE18CS043

Department of Computer Science

January 18, 2021

Contents

CS431 - Compiler Design Lab · 2021 ·

Cycle 2 Experiment 2

1 FIRST and FOLLOW

1.1 Aim

Write a program to find the First and Follow of any given grammar symbols.

1.2 Theory

First: If the compiler would have come to know in advance, that what is the "first character of the string produced when a production rule is applied", and comparing it to the current character or token in the input string it sees, it can wisely take decision on which production rule to apply.

Follow: FOLLOW can make a Non-terminal to vanish out if needed to generate the string from the parse tree.

1.3 Algorithm

```
1 Start
2 FIRST ( X ) for all grammar symbols X
    1. If X is terminal , FIRST ( X ) = { X }.
    2. If X -> e is a production , then add e to FIRST ( X ).
    3. If X is a non - terminal , and X -> Y1 Y2 ... Yk is a production ,
       and e is in all of FIRST ( Y1 ) , ... , FIRST ( Yk ) , then add e to
       FIRST ( X ).
    4. If X is a non - terminal , and X -> Y1 Y2 ... Yk is a production ,
       then add a to FIRST ( X ) if for some i , a is in FIRST ( Yi ) , and e
       is in all of FIRST ( Y1 ) , ... , FIRST ( Yi -1 ).
3 FOLLOW ( A ) for all non - terminals A
    1. If $ is the input end - marker , and S is the start symbol , $
       element of FOLLOW ( S ).
    2. If there is a production , A -> aBb , then ( FIRST ( b ) - e ) subset
       of FOLLOW ( B ).
    3. If there is a production , A -> aB , or a production A -> aBb ,
       where e element of FIRST ( b ), then FOLLOW ( A ) subset of FOLLOW ( B ).
4 Stop
```

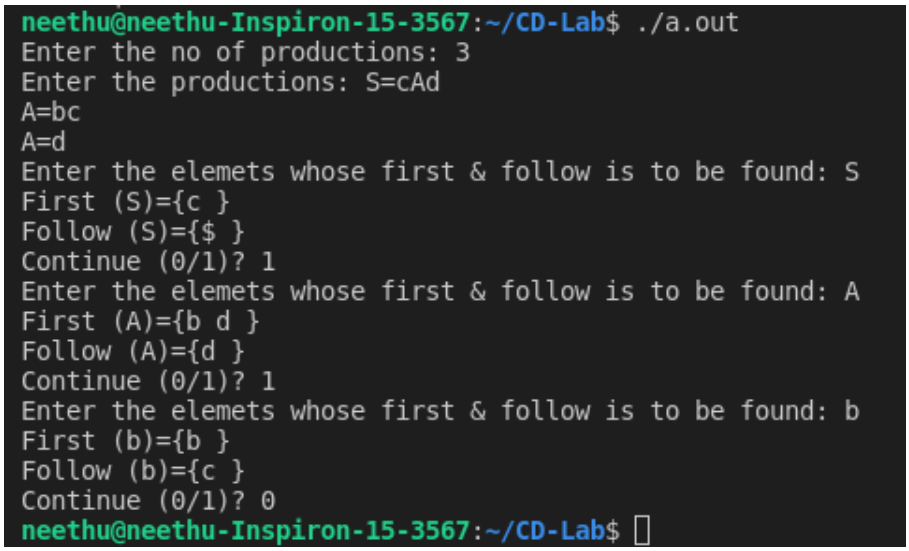
1.4 Code

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.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;
    // clrscr ();
    printf("Enter the no of productions: ");
    scanf("%d", &n);
    printf("Enter the productions: ");
    for (i = 0; i < n; i++)
        scanf("%s%c", a[i], &ch);
    do
    {
        m = 0;
        printf("Enter the elemets whose first & 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, " ");
        // flushall ();
        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]);
            }
        }
    }
}
```

1.5 Output



1.6 Result

Implemented the program for finding the first and follow of any grammer symbols using C language in Ubuntu 20.04 with kernel and the above outputs were obtained.