EX NO 18 :

Write a c program to find FIRST and FOLLOW for predictive parser.

PROGRAM :

#include<stdio.h>

#include<ctype.h>

#include<string.h>

void T4Tutorials(char, int, int);

void CFG\_Follow(char c);

void Searching\_First(char, int, int);

int count, n = 0;

char FirstCalculation[10][100];

char calc\_CFG\_Follow[10][100];

int m = 0;

char production[10][10];

char f[10], first[10];

int k;

char ck;

int e;

int main(int argc, char \*\*argv)

{

int jm = 0;

int km = 0;

int i, choice;

char c, ch;

count = 4;

// The Input grammar

strcpy(production[0], "E=XY");

strcpy(production[1], "X=ilove");

strcpy(production[3], "Y=t4tutorials");

int kay;

char done[count];

int ptr = -1;

// Initializing the FirstCalculation array

for(k = 0; k < count; k++) {

for(kay = 0; kay < 100; kay++) {

FirstCalculation[k][kay] = '!';

}

}

int point1 = 0, point2, xxx;

for(k = 0; k < count; k++)

{

c = production[k][0];

point2 = 0;

xxx = 0;

// Checking if First of c has already been calculated

for(kay = 0; kay <= ptr; kay++)

if(c == done[kay])

xxx = 1;

if (xxx == 1)

continue;

// Function call

Searching\_First(c, 0, 0);

ptr += 1;

// Adding c to the calculated list

done[ptr] = c;

printf("\n First(%c) = { ", c);

FirstCalculation[point1][point2++] = c;

// Printing the First Sets of the grammar

for(i = 0 + jm; i < n; i++) {

int lark = 0, chk = 0;

for(lark = 0; lark < point2; lark++) {

if (first[i] == FirstCalculation[point1][lark])

{

chk = 1;

break;

}

}

if(chk == 0)

{

printf("%c, ", first[i]);

FirstCalculation[point1][point2++] = first[i];

}

}

printf("}\n");

jm = n;

point1++;

}

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

char donee[count];

ptr = -1;

// Initializing the calc\_CFG\_Follow array

for(k = 0; k < count; k++) {

for(kay = 0; kay < 100; kay++) {

calc\_CFG\_Follow[k][kay] = '!';

}

}

point1 = 0;

int land = 0;

for(e = 0; e < count; e++)

{

ck = production[e][0];

point2 = 0;

xxx = 0;

// Checking if CFG\_Follow of ck

// has alredy been calculated

for(kay = 0; kay <= ptr; kay++)

if(ck == donee[kay])

xxx = 1;

if (xxx == 1)

continue;

land += 1;

// Function call

CFG\_Follow(ck);

ptr += 1;

// Adding ck to the calculated list

donee[ptr] = ck;

printf(" CFG\_Follow(%c) = { ", ck);

calc\_CFG\_Follow[point1][point2++] = ck;

// Printing the CFG\_Follow Sets of the grammar

for(i = 0 + km; i < m; i++) {

int lark = 0, chk = 0;

for(lark = 0; lark < point2; lark++)

{

if (f[i] == calc\_CFG\_Follow[point1][lark])

{

chk = 1;

break;

}

}

if(chk == 0)

{

printf("%c, ", f[i]);

calc\_CFG\_Follow[point1][point2++] = f[i];

}

}

printf(" }\n\n");

km = m;

point1++;

}

}

void CFG\_Follow(char c)

{

int i, j;

// Adding "$" to the CFG\_Follow, set of the start symbol

if(production[0][0] == c) {

f[m++] = '$';

}

for(i = 0; i < 10; i++)

{

for(j = 2;j < 10; j++)

{

if(production[i][j] == c)

{

if(production[i][j+1] != '\0')

{

// Calculate the first of the next

// Variable (Non-Terminal ) in the production

T4Tutorials(production[i][j+1], i, (j+2));

}

if(production[i][j+1]=='\0' && c!=production[i][0])

{

// Calculate the CFG\_Follow of the Variable (Non-Terminal )

// in the L.H.S. of the production

CFG\_Follow(production[i][0]);

}

}

}

}

}

void Searching\_First(char c, int q1, int q2)

{

int j;

// The case where we

// encounter a Terminal

if(!(isupper(c))) {

first[n++] = c;

}

for(j = 0; j < count; j++)

{

if(production[j][0] == c)

{

if(production[j][2] == '#')

{

if(production[q1][q2] == '\0')

first[n++] = '#';

else if(production[q1][q2] != '\0'

&& (q1 != 0 || q2 != 0))

{

// Recursion to calculate First of New

// Variable (Non-Terminal ) we encounter after epsilon

Searching\_First(production[q1][q2], q1, (q2+1));

}

else

first[n++] = '#';

}

else if(!isupper(production[j][2]))

{

first[n++] = production[j][2];

}

else

{

// Recursion to calculate First of

// New Variable (Non-Terminal ) we encounter

// at the beginning

Searching\_First(production[j][2], j, 3);

}

}

}

}

void T4Tutorials(char c, int c1, int c2)

{

int k;

// The case where we encounter

// a Terminal

if(!(isupper(c)))

f[m++] = c;

else

{

int i = 0, j = 1;

for(i = 0; i < count; i++)

{

if(FirstCalculation[i][0] == c)

break;

}

//Including the First set of the

// Variable (Non-Terminal ) in the CFG\_Follow of

// the original query

while(FirstCalculation[i][j] != '!')

{

if(FirstCalculation[i][j] != '#')

{

f[m++] = FirstCalculation[i][j];

}

else

{

if(production[c1][c2] == '\0')

{

// Case where we reach the

// end of a production

CFG\_Follow(production[c1][0]);

}

else

{

// Recursion to the next symbol in case we encounter a "#"

T4Tutorials(production[c1][c2], c1, c2+1);

}

}

j++;

}

}

}

OUTPUT :

