

Lab7: Compute First and follow sets of given grammar

Aim: Write a program to Compute First and follow sets of given grammar.

Program-2:

First and follow sets of your project related grammar.

Project no: 5

IT082-Nakrani Dhruvil
IT088-Pankhania Aanandi

Following is a valid sentence in a layman friendly “comparison ” program. Generate its appropriate language description and compiler-

- Valid sentences in language :
 1. Which is bigger number 7 or 2?
 2. From 7 and 2 which is larger ?
 3. Is 7 bigger than 2?
 4. Is 2 smaller than 4?
 5. Which is smaller between 4 and 9?

Which is smaller number 4 or 2?

C code: n1.c

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
typedef struct P{
    char lhs[3], rhs[10][10], first[20], follow[20], rfirst[10][20];
    int length[10], SP;
}P;
P P1[20];
int nop;
```

```

int f_Index(char ch){
    for(int i=0; i<nop; i++)
        if(P1[i].lhs[0] == ch)
            return i;
    return -1;
}

```

```

int IsNull(char *s){
    for(int i=0; i<strlen(s); i++)
        if(s[i]=='^')
            return 1;
    return 0;
}

```

```

void add_here(char ch, char *first){
    int j=0;
    for( ; j<strlen(first); j++)
        if(first[j]==ch)
            return;
    first[j++]=ch;
}

```

```

void fn_print(char *a1, char *a2, int null){
    if(null){
        for(int m=0; m<strlen(a1); m++)
            if(a1[m] != '^')
                add_here(a1[m], a2);
    }
    else{
        for(int m=0; m<strlen(a1); m++)
            add_here(a1[m], a2);
    }
}

```

```

void ForFirstSet(int a){
    int b;
    for(b=0; b<P1[a].SP; b++){
        if(P1[a].rhs[b][0] == '^'){
            add_here(P1[a].rhs[b][0], P1[a].first);
            add_here(P1[a].rhs[b][0], P1[a].rfirst[b]);
        }
        else if(!isupper(P1[a].rhs[b][0])){
            add_here(P1[a].rhs[b][0], P1[a].first);
        }
    }
}

```

```

        add_here(P1[a].rhs[b][0], P1[a].rfirst[b]);
    }
    else{
        int l = 0, k, m;
        while(l < strlen(P1[a].rhs[b])){
            if(!isupper(P1[a].rhs[b][l])){
                add_here(P1[a].rhs[b][l],P1[a].first);
                add_here(P1[a].rhs[b][l],P1[a].rfirst[b]);
                break;
            }
            k = f_Index(P1[a].rhs[b][l]);
            if(IsNull(P1[k].first)){
                fn_print(P1[k].first, P1[a].first, 1);
                fn_print(P1[k].first, P1[a].rfirst[b],1);
                l++;
            }
            else{
                fn_print(P1[k].first, P1[a].first, 0);
                fn_print(P1[k].first, P1[a].rfirst[b],0);
                break;
            }
        }
        if(l == strlen(P1[a].rhs[b])){
            add_here('^', P1[a].first);
            add_here('^', P1[a].rfirst[b]);
        }
    }
}
}
}

```

[illegible]

```

0);
                                fn_print(P1[b].follow,P1[a].follow,
                                break;
                                }
                                else{
                                    if(!isupper(P1[b].rhs[c][d+1])){
                                        add_here(P1[b].rhs[c][d+1],P1[a].follow);
                                        break;
                                    }
                                    int pos =
f_Index(P1[b].rhs[c][d+1]);
                                    if(IsNull(P1[pos].first)){
                                        fn_print(P1[pos].first,P1[a].follow, 1);
                                        d++;
                                    }
                                    else{
                                        fn_print(P1[pos].first,P1[a].follow, 0);
                                        break;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

int main(){
    int i, j;
    printf("\n*** Program to find First and Follow set ***\n\n");
    printf("-----");
    printf("\nEnter Total No of Productions : ");
    scanf("%d", &nop);

    for(i=0; i<nop; i++){
        memset(P1[i].lhs, 0, sizeof(P1[i].lhs));
        memset(P1[i].rhs, 0, sizeof(P1[i].rhs));
        memset(P1[i].first, 0, sizeof(P1[i].first));
        memset(P1[i].rfirst, 0, sizeof(P1[i].rfirst));
        memset(P1[i].follow, 0, sizeof(P1[i].follow));
        memset(P1[i].length, 0, sizeof(P1[i].length));
    }
}

```

```

}
for(i=0; i<nop; i++){
    printf("LHS of Production [%d] : ", i+1);
    scanf("%s", P1[i].lhs);
    printf("Enter Total No of Subproductions of P[%d] : ", i+1);
    scanf("%d", &P1[i].SP);
    for(j=0; j<P1[i].SP; j++){
        printf("RHS of subproduction[%d] : ", j+1);
        scanf("%s", P1[i].rhs[j]);
        P1[i].length[j]=strlen(P1[i].rhs[j]);
    }
}

printf("-----");
printf("\nGrammar entered: \n");
for(i=0; i<nop; i++){
    printf("%s -> ", P1[i].lhs);
    for(j=0; j<P1[i].SP; j++){
        if(j!=0)
            printf(" | %s", P1[i].rhs[j]);
        else
            printf("%s", P1[i].rhs[j]);
    }
    printf("\n");
}

for(i=(nop-1); i>(-1); i--)
    ForFirstSet(i);

for(i=0; i<nop; i++)
    ForFollowSet(i);
printf("-----");

printf("\nFIRST Sets : \n");
for(i=0; i<nop; i++){
    printf("FIRST(%s) : ", P1[i].lhs);
    for(j=0; j<strlen(P1[i].first); j++){
        if(j != 0)
            printf(", %c", P1[i].first[j]);
        else
            printf(" { %c", P1[i].first[j]);
    }
    printf(" }\n");
}

```

```

    }
    printf("-----");
    printf("\nFOLLOW sets : \n");
    for(i=0; i<nop; i++){
        printf("FOLLOW(%s) : ", P1[i].lhs);
        for(j=0; j<strlen(P1[i].follow); j++){
            if(j != 0)
                printf(", %c", P1[i].follow[j]);
            else
                printf("{ %c", P1[i].follow[j]);
        }
        printf(" }\n");
    }
    printf("\n");
    return 0;
}

```

Output Screenshots:

```

aanandi@Aanandi: /mnt/f/IT088/LT/Lab7 (First And Follow)
aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow)$ gcc n1.c
aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow)$ ./a.out

*** Program to find First and Follow set ***

-----
Enter Total No of Productions : 6
LHS of Production [1] : S
Enter Total No of Subproductions of P[1] : 1
RHS of subproduction[1] : aBDh
LHS of Production [2] : B
Enter Total No of Subproductions of P[2] : 1
RHS of subproduction[1] : cC
LHS of Production [3] : C
Enter Total No of Subproductions of P[3] : 2
RHS of subproduction[1] : bC
RHS of subproduction[2] : ^
LHS of Production [4] : D
Enter Total No of Subproductions of P[4] : 1
RHS of subproduction[1] : EF
LHS of Production [5] : E
Enter Total No of Subproductions of P[5] : 2
RHS of subproduction[1] : g
RHS of subproduction[2] : ^
LHS of Production [6] : F
Enter Total No of Subproductions of P[6] : 2
RHS of subproduction[1] : f
RHS of subproduction[2] : ^
-----

```

```

-----
Grammar entered:
S -> aBDh
B -> cC
C -> bC | ^
D -> EF
E -> g | ^
F -> f | ^
-----

FIRST Sets :
FIRST(S) : { a }
FIRST(B) : { c }
FIRST(C) : { b, ^ }
FIRST(D) : { g, f, ^ }
FIRST(E) : { g, ^ }
FIRST(F) : { f, ^ }
-----

FOLLOW sets :
FOLLOW(S) : { $ }
FOLLOW(B) : { g, f, h }
FOLLOW(C) : { g, f, h }
FOLLOW(D) : { h }
FOLLOW(E) : { f, h }
FOLLOW(F) : { h }

aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow)$

```

```

aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow) $ gcc n1.c
aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow) $ ./a.out

```

```

*** Program to find First and Follow set ***

```

```

-----
Enter Total No of Productions : 5
LHS of Production [1] : S
Enter Total No of Subproductions of P[1] : 1
RHS of subproduction[1] : A
LHS of Production [2] : A
Enter Total No of Subproductions of P[2] : 1
RHS of subproduction[1] : aBX
LHS of Production [3] : X
Enter Total No of Subproductions of P[3] : 2
RHS of subproduction[1] : dX
RHS of subproduction[2] : ^
LHS of Production [4] : B
Enter Total No of Subproductions of P[4] : 1
RHS of subproduction[1] : b
LHS of Production [5] : C
Enter Total No of Subproductions of P[5] : 1
RHS of subproduction[1] : g
-----

```

```
-----  
Grammar entered:  
S -> A  
A -> aBX  
X -> dX | ^  
B -> b  
C -> g  
-----  
FIRST Sets :  
FIRST(S) : { a }  
FIRST(A) : { a }  
FIRST(X) : { d, ^ }  
FIRST(B) : { b }  
FIRST(C) : { g }  
-----  
FOLLOW sets :  
FOLLOW(S) : { $ }  
FOLLOW(A) : { $ }  
FOLLOW(X) : { $ }  
FOLLOW(B) : { d, $ }  
FOLLOW(C) : { }  
aanandi@Aanandi:/mnt/f/IT088/LT/Lab7 (First And Follow)$
```

Reference for above o/p:

First---Problem-1: <https://www.gatevidyalay.com/first-and-follow-compiler-design/>

Second---Problem-2: <https://www.gatevidyalay.com/first-and-follow-compiler-design/>