

CSB 353: Compiler Design

LAB 7

Submitted By:

Name: PREM KUMAR

Roll No: 191210037

Branch: CSE

Semester: 6 th

Submitted To: Dr. Shelly Sachdeva

Department of Computer Science and Engineering



NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2019-2023

Ques 1. Write a C program to find First and Follow for all non-terminals.

Code:

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <conio.h>
4  #define max 20
5  char prod[max][10];
6  char ter[10], nt[10];
7  char first[10][10], follow[10][10];
8  int eps[10];
9  int count = 0;
10 int findpos(char ch)
11 {
12     int n;
13     for (n = 0; nt[n] != '\0'; n++)
14         if (nt[n] == ch)
15             break;
16     if (nt[n] == '\0')
17         return 1;
18     return n;
19 }
20
21 int IsCap(char c)
22 {
23     if (c >= 'A' && c <= 'Z')
24         return 1;
25     return 0;
26 }
27 void add(char *arr, char c)
28 {
29     int i, flag = 0;
30     for (i = 0; arr[i] != '\0'; i++)
31     {
32         if (arr[i] == c)
33         {
34             flag = 1;
35             break;
36         }
37     }
```

```

38     if (flag != 1)
39         arr[strlen(arr)] = c;
40 }
41 void addarr(char *s1, char *s2)
42 {
43     int i, j, flag = 0;
44     for (i = 0; s2[i] != '\0'; i++)
45     {
46         flag = 0;
47         for (j = 0;; j++)
48         {
49             if (s2[i] == s1[j])
50             {
51                 flag = 1;
52                 break;
53             }
54             if (j == strlen(s1) && flag != 1)
55             {
56                 s1[strlen(s1)] = s2[i];
57                 break;
58             }
59         }
60     }
61 }
62 void addprod(char *s)
63 {
64     int i;
65     prod[count][0] = s[0];
66     for (i = 3; s[i] != '\0'; i++)
67     {
68         if (!IsCap(s[i]))
69             add(ter, s[i]);
70         prod[count][i - 2] = s[i];
71     }
72     prod[count][i - 2] = '\0';
73     add(nt, s[0]);
74     count++;

```

```

75     }
76 void findfirst()
77 {
78     int i, j, n, k, e, n1;
79     for (i = 0; i < count; i++)
80     {
81         for (j = 0; j < count; j++)
82         {
83             n = findpos(prod[j][0]);
84             if (prod[j][1] == (char)238)
85                 eps[n] = 1;
86             else
87             {
88                 for (k = 1, e = 1; prod[j][k] != '\0' && e == 1; k++)
89                 {
90                     if (!IsCap(prod[j][k]))
91                     {
92                         e = 0;
93                         add(first[n], prod[j][k]);
94                     }
95                     else
96                     {
97                         n1 = findpos(prod[j][k]);
98                         addarr(first[n], first[n1]);
99                         if (eps[n1] == 0)
100                             e = 0;
101                     }
102                 }
103                 if (e == 1)
104                     eps[n] = 1;
105             }
106         }
107     }
108 }
109 void findfollow()
110 {
111     int i, j, k, n, e, n1;

```

```

112     n = findpos(prod[0][0]);
113     add(follow[n], '$');
114     for (i = 0; i < count; i++)
115     {
116         for (j = 0; j < count; j++)
117         {
118             k = strlen(prod[j]) - 1;
119             for (; k > 0; k--)
120             {
121                 if (IsCap(prod[j][k]))
122                 {
123                     n = findpos(prod[j][k]);
124                     if (prod[j][k + 1] == '\0')
125                     {
126                         n1 = findpos(prod[j][0]);
127                         addarr(follow[n], follow[n1]);
128                     }
129                     if (IsCap(prod[j][k + 1]))
130                     {
131                         n1 = findpos(prod[j][k + 1]);
132                         addarr(follow[n], first[n1]);
133                         if (eps[n1] == 1)
134                         {
135                             n1 = findpos(prod[j][0]);
136                             addarr(follow[n], follow[n1]);
137                         }
138                     }
139                     else if (prod[j][k + 1] != '\0')
140                         add(follow[n], prod[j][k + 1]);
141                 }
142             }
143         }
144     }
145 }
146 int main()
147 {
148     char s[max];

```

```

149     int i, j;
150     int prods;
151     printf("\nEnter the number of productions: ");
152     scanf("%d", &prods);
153
154     printf("\nEnter the productions in sequence:\n");
155
156     scanf("%s", s);
157     while (--prods)
158     {
159         addprod(s);
160         scanf("%s", s);
161     }
162     addprod(s);
163     findfirst();
164     findfollow();
165
166     for (i = 0; i < strlen(nt); i++)
167     {
168         printf("\nFirst(%c) = ", nt[i]);
169         for (j = 0; j < strlen(first[i]); j++)
170         {
171             if (j == 0)
172                 printf("{");
173             printf("%c", first[i][j]);
174             if (j != (strlen(first[i]) - 1))
175                 printf(",");
176             else
177                 printf("}\n");
178         }
179         printf("Follow(%c) = ", nt[i]);
180         for (j = 0; j < strlen(follow[i]); j++)
181         {
182             if (j == 0)
183                 printf("{");
184             printf("%c", follow[i][j]);
185             if (j != (strlen(follow[i]) - 1))
186                 printf(",");
187             else
188                 printf("}\n");
189         }
190     }
191     return 0;
192 }

```

Output:

```
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab7> cd "c:\Users\Prem\Desktop\6thSem\CSB353\lab7\"
($?) { .\firstAndFollow }

Enter the number of productions: 6

Enter the productions in sequence:
E->E+T
E->T
T->T*F
T->F
F->(E)
F->i

First(E) = {(,i}
Follow(E) = {$,+,)}

First(T) = {(,i}
Follow(T) = {$,+,*,)}

First(F) = {(,i}
Follow(F) = {$,+,*,)}
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab7> █
```