

18CSC304J/ Compiler Design

Submitted By:- ANANNYA P. NEOG (RA1911003010367)

Exp-8: Computation of LEADING AND TRAILING

Aim:- To write code to compute LEADING AND TRAILING

Codes:-

```
#include<iostream>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
using namespace std;
int vars,terms,i,j,k,m,rep,count,temp=-1;
char var[10],term[10],lead[10][10],trail[10][10];
struct grammar
{
    int prodno;
    char lhs,rhs[20][20];
}

gram[50];
void get()
{
    cout<<"\n----- LEADING AND TRAILING ----- \n";
    cout<<"\nEnter the no. of variables : ";
    cin>>vars;
    cout<<"\nEnter the variables : \n";
    for(i=0;i<vars;i++)
    {
        cin>>gram[i].lhs;
        var[i]=gram[i].lhs;
    }

    cout<<"\nEnter the no. of terminals : ";
    cin>>terms;
    cout<<"\nEnter the terminals : ";
    for(j=0;j<terms;j++)
    cin>>term[j];
    cout<<"\n----- PRODUCTION DETAILS ----- \n";
    for(i=0;i<vars;i++)
    {
        cout<<"\nEnter the no. of production of "<<gram[i].lhs<<": ";
        cin>>gram[i].prodno;
        for(j=0;j<gram[i].prodno;j++)
        {
            cout<<gram[i].lhs<<"->";
            cin>>gram[i].rhs[j];
        }
    }
}
```

```

    }
}
}

void leading()
{
    for(i=0;i<vars;i++)
    {
        for(j=0;j<gram[i].prodno;j++)
        {
            for(k=0;k<terms;k++)
            {
                if(gram[i].rhs[j][0]==term[k])
                    lead[i][k]=1;
                else
                {
                    if(gram[i].rhs[j][1]==term[k])
                        lead[i][k]=1;
                }
            }
        }
    }
    for(rep=0;rep<vars;rep++)
    {
        for(i=0;i<vars;i++)
        {
            for(j=0;j<gram[i].prodno;j++)
            {
                for(m=1;m<vars;m++)
                {
                    if(gram[i].rhs[j][0]==var[m])
                    {
                        temp=m;
                        goto out;
                    }
                }
            }
            out:
            for(k=0;k<terms;k++)
            {
                if(lead[temp][k]==1)
                    lead[i][k]=1;
            }
        }
    }
}
}

void trailing()
{

```

```

for(i=0;i<vars;i++)
{
    for(j=0;j<gram[i].prodno;j++)
    {
        count=0;
        while(gram[i].rhs[j][count]!='\x0')
            count++;
        for(k=0;k<terms;k++)
        {
            if(gram[i].rhs[j][count-1]==term[k])
                trail[i][k]=1;
            else
            {
                if(gram[i].rhs[j][count-2]==term[k])
                    trail[i][k]=1;
            }
        }
    }
}
for(rep=0;rep<vars;rep++)
{
    for(i=0;i<vars;i++)
    {
        for(j=0;j<gram[i].prodno;j++)
        {
            count=0;
            while(gram[i].rhs[j][count]!='\x0')
                count++;
            for(m=1;m<vars;m++)
            {
                if(gram[i].rhs[j][count-1]==var[m])
                    temp=m;
            }
            for(k=0;k<terms;k++)
            {
                if(trail[temp][k]==1)
                    trail[i][k]=1;
            }
        }
    }
}

void display()
{
    for(i=0;i<vars;i++)
    {
        cout<<"\nLEADING("<<gram[i].lhs<<") = ";
        for(j=0;j<terms;j++)

```

```

{
    if(lead[i][j]==1)
        cout<<term[j]<<" ";
}
}
cout<<endl;
for(i=0;i<vars;i++)
{
    cout<<"\nTRAILING(";<<gram[i].lhs<<" ) = ";
    for(j=0;j<terms;j++)
    {
        if(trail[i][j]==1)
            cout<<term[j]<<" ";
    }
}
}
}

int main()
{
    get();
    leading();
    trailing();
    display();
    getch();
}

```

```
1 #include<iostream>
2 #include<conio.h>
3 #include<stdio.h>
4 #include<string.h>
5 #include<stdlib.h>
6 using namespace std;
7 int vars,terms,i,j,k,m,rep,count,temp=-1;
8 char var[10],term[10],lead[10][10],trail[10][10];
9 struct grammar
10 {
11     int prodno;
12     char lhs,rhs[20][20];
13 }
14 gram[50];
15 void get()
16 {
17     cout<<"\n----- LEADING AND TRAILING ----- \n";
18     cout<<"\nEnter the no. of variables : ";
19     cin>>vars;
20     cout<<"\nEnter the variables : \n";
21     for(i=0;i<vars;i++)
22     {
23         cin>>gram[i].lhs;
24         var[i]=gram[i].lhs;
25     }
26
27     cout<<"\nEnter the no. of terminals : ";
28     cin>>terms;
29     cout<<"\nEnter the terminals : ";
30     for(j=0;j<terms;j++)
31     {
32         cin>>term[j];
33         cout<<"\n----- PRODUCTION DETAILS ----- \n";
34         for(i=0;i<vars;i++)
35         {
36             cout<<"\nEnter the no. of production of "<<gram[i].lhs<<": ";
37             cin>>gram[i].prodno;
38             for(j=0;j<gram[i].prodno;j++)
39             {
```

```
main.cpp
40 cout<<gram[i].lhs<<"->";
41 cin>>gram[i].rhs[j];
42 }
43 }
44 }
45
46 void leading()
47 {
48     for(i=0;i<vars;i++)
49     {
50         for(j=0;j<gram[i].prodno;j++)
51         {
52             for(k=0;k<terms;k++)
53             {
54                 if(gram[i].rhs[j][0]==term[k])
55                     lead[i][k]=1;
56                 else
57                 {
58                     if(gram[i].rhs[j][1]==term[k])
59                         lead[i][k]=1;
60                 }
61             }
62         }
63     }
64     for(rep=0;rep<vars;rep++)
65     {
66         for(i=0;i<vars;i++)
67         {
68             for(j=0;j<gram[i].prodno;j++)
69             {
70                 for(m=1;m<vars;m++)
71                 {
72                     if(gram[i].rhs[j][0]==var[m])
73                     {
74                         temp=m;
75                         goto out;
76                     }
77                 }
78                 out:
```

```
main.cpp
79         for(k=0;k<terms;k++)
80         {
81             if(lead[temp][k]==1)
82                 lead[i][k]=1;
83         }
84     }
85 }
86 }
87 }
88
89 void trailing()
90 {
91     for(i=0;i<vars;i++)
92     {
93         for(j=0;j<gram[i].prodno;j++)
94         {
95             count=0;
96             while(gram[i].rhs[j][count]!='\x0')
97                 count++;
98             for(k=0;k<terms;k++)
99             {
100                 if(gram[i].rhs[j][count-1]==term[k])
101                     trail[i][k]=1;
102                 else
103                 {
104                     if(gram[i].rhs[j][count-2]==term[k])
105                         trail[i][k]=1;
106                 }
107             }
108         }
109     }
110     for(rep=0;rep<vars;rep++)
111     {
112         for(i=0;i<vars;i++)
113         {
114             for(j=0;j<gram[i].prodno;j++)
115             {
116                 count=0;
117                 while(gram[i].rhs[j][count]!='\x0')
```

Run

Debug

Stop

Share

Save

Beautify

Language C++

```
main.cpp
118     count++;
119     for(m=1;m<vars;m++)
120     {
121         if(gram[i].rhs[j][count-1]==var[m])
122             temp=m;
123     }
124     for(k=0;k<terms;k++)
125     {
126         if(trail[temp][k]==1)
127             trail[i][k]=1;
128     }
129 }
130 }
131 }
132 }
133
134 void display()
135 {
136     for(i=0;i<vars;i++)
137     {
138         cout<<"\nLEADING("<<gram[i].lhs<<" ) = ";
139         for(j=0;j<terms;j++)
140         {
141             if(lead[i][j]==1)
142                 cout<<term[j]<<",";
143         }
144         cout<<endl;
145         for(i=0;i<vars;i++)
146         {
147             cout<<"\nTRAILING("<<gram[i].lhs<<" ) = ";
148             for(j=0;j<terms;j++)
149             {
150                 if(trail[i][j]==1)
151                     cout<<term[j]<<",";
152             }
153         }
154     }
155 }
156
157 int main()
158 {
159     get();
160     leading();
161     trailing();
162     display();
163     getch();
164 }
```

input

Output:-

```
input
----- LEADING AND TRAILING -----
Enter the no. of variables : 3

Enter the variables :
E
T
F

Enter the no. of terminals : 5

Enter the terminals : )
(
*
+
i

----- PRODUCTION DETAILS -----

Enter the no. of production of E:2
E->E+T
E->T

Enter the no. of production of T:2
T->T*F
T->F

Enter the no. of production of F:2
F->(E)
F->i

LEADING(E) = (,*,+,i,
LEADING(T) = (,*,i,
LEADING(F) = (,i,

TRAILING(E) = ),*,+,i,
TRAILING(T) = ),*,i,
TRAILING(F) = ),i,

...Program finished with exit code 0
Press ENTER to exit console.□
```