

Compiler Design

Leading and Trailing

EXPERIMENT - 8

Roehit Ranganathan | RA1911033010017
15 March 2022

Aim:

To Build a program that calculates the lead and trail of a grammar in C/C++/Java.

Program:

```
#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 << "\nLEADING 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 << "\nPRODUCTION 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++)
        {
            int count = 0;
            while (gram[i].rhs[j][count] != '\x00')

```

```

        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++)
        {
            int 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();
}

```

Sample Input & Output:

```
PS D:\SRM\SEM 6\Compiler Design Lab\EXP-8> cd "d:\SRM\SEM 6\Compiler Design Lab\EXP-8\" ; if ($?) { g++ exp8.cc -o exp8 } ; if ($?) { .\exp8 }

LEADING AND TRAILING

Enter the no. of variables : 3

Enter the variables :
A
B
C

Enter the no. of terminals : 5

Enter the terminals : (
)
+
*
i

PRODUCTION DETAILS

Enter the no. of production of A:2
A->A+B
A->B

Enter the no. of production of B:2
B->B*C
B->C

Enter the no. of production of C:2
C->A)
C->i

LEADING(A) = ),+,*,i,
LEADING(B) = ),*,i,
LEADING(C) = ),i,

TRAILING(A) = ),+,*,i,
TRAILING(B) = ),*,i,
TRAILING(C) = ),i,
PS D:\SRM\SEM 6\Compiler Design Lab\EXP-8> █
```

Result:

The Program was successfully executed.