LAB7: Compute First and follow sets of given grammar.

IT057-Kathiriya Darshakkumar

Code:

```
#include <bits/stdc++.h>
using namespace std;

bool cmp(string &x, string &y)
{
    return x.size() > y.size();
}

string filter$(string dar)
{
    for (int i = 0; i < dar.length(); i += 3)
    {
        for (int j = i + 3; j < dar.length(); j += 3)
        {
            if (dar.substr(i, 3) == dar.substr(j, 3))
            {
                dar = dar.substr(0, j) + dar.substr(j + 3);
            }
        }
    }
    return dar;
}

string filterNull(string dar)
{
```

```
    dar = filter$(dar);
    for (int i = 0; i < dar.length(); i += 3)
    {
        if(dar.substr(i,3)=="^, ")
            dar = dar.substr(0,i) + dar.substr(i+3);
    }
    return dar;
}

void solve57()
{
    string input[100][100];
    string nonTerm[100];
    int proCount[100];
    int cnt;
    cout << "Enter total number of expressions: ";
    cin >>cnt;
    for (int i = 0; i <cnt; i++)
    {
        cout << "Enter non-terminal: ";
        cin >> nonTerm[i];
        cout << "Enter number of productions: ";
        cin >> proCount[i];
        int j = 0;
        for (j = 0; j < proCount[i]; j++)
        {
            cout << "Enter production " << j + 1 << " : ";
            cin >> input[i][j];
        }
        sort(input[i], input[i] + j, cmp);
    }
    cout << "\nGiven Input Expression:\n";
    for (int i = 0; i < cnt; i++)
```

```
    {
        cout << nonTerm[i] << " -> ";
        cout << input[i][0];
        for (int j = 1; j < proCount[i]; j++)
        {
            cout << " | " << input[i][j];
        }
        cout << endl;
    }
    string first[cnt];
    for (int dm = 0; dm < cnt; ++dm)
    {
        for (int i = 0; i < cnt; ++i)
        {
            for (int j = 0; j < proCount[i]; j++)
            {
//check for null(^) production
                int m = 0;
                for (; m < input[i][j].length(); ++m)
                {

                    int non_t_index = -1;
                    for (int n = 0; n < cnt; ++n)
                    {
                        if (input[i][j].substr(m, nonTerm[n].length()) == nonTerm[n])
                        {
                            non_t_index = n;
                            break;
                        }
                    }
                    if (non_t_index == -1)
                    {
                        first[i] = filter$(first[i] + input[i][j].substr(m, 1) + ", ");
```

```
                break;
            }
            if (first[non_t_index].find('^') != -1)
            {
                first[i] = filter$(first[i] + first[non_t_index]);
            }
            else
            {
                first[i] = filter$(first[i] + first[non_t_index]);
                break;
            }
        }
    }
  }
}

cout << "\nFirst-set of Given Expression:\n";
for (int i = 0; i < cnt; i++)
{
    cout << nonTerm[i] << " -> ";
    cout << first[i];
    cout << endl;
}
string follow[cnt];
cout << follow[1];
if (cnt)
    follow[0] = "$, ";
for (int dm = 0; dm < cnt; ++dm)
{
    for (int i = 0; i < cnt; ++i)
    {
        for (int j = 0; j < cnt; ++j)
        {
```

```
for (int k = 0; k < proCount[j]; ++k)
{
    for (int l = 0; l < input[j][k].length(); ++l)
    {
        if (input[j][k].substr(l, nonTerm[i].length()) == nonTerm[i])
        {
            string next_string = input[j][k].substr(l + nonTerm[i].length());
            if (next_string.length() == 0)
            {
                follow[i] = filterNull (follow[i] + follow[j]);
                continue;
            }
            else
            {
                int m = 0;
                for (; m < next_string.length(); ++m)
                {
                    int non_t_index = -1;
                    for (int n = 0; n < cnt; ++n)
                    {
                        if (next_string.substr(m, nonTerm[n].length()) == nonTerm[n])
                        {
                            non_t_index = n;
                            break;
                        }
                    }
                    if (non_t_index == -1)
                    {
                        follow[i] = filterNull (follow[i] + next_string.substr(m, 1) + ", ");
                        break;
                    }
                    if (first[non_t_index].find('^') != -1)
                    {
```

```
                    follow[i] = filterNull (follow[i] + first[non_t_index]);
                    continue;
                }
                else
                {
                    follow[i] = filterNull (follow[i] + first[non_t_index]);
                    break;
                }
            }
            if (m == next_string.length())
            {
                follow[i] = filterNull(follow[i] + follow[j]);
            }
        }
    }
  }
 }
}
    cout << "\nFollow-set of Given Expression:\n";
    for (int i = 0; i < cnt; i++)
    {
        cout << nonTerm[i] << " -> ";
        cout << follow[i];
        cout << endl;
    }
}

int main()
{
    int i=1;
    while(i)
```

```
    {
        solve57();
        i--;
    }
    return 0;
}
```

Output1:

```
"D:\00 Study\SEM 6\0LAB\LT\LAB 7\firstndfolow.exe"
Enter total number of expressions: 3
Enter non-terminal: S
Enter number of productions: 2
Enter production 1 : AaAb
Enter production 2 : BbBa
Enter non-terminal: A
Enter number of productions: 1
Enter production 1 : ^
Enter non-terminal: B
Enter number of productions: 1
Enter production 1 : ^

Given Input Expression:
S -> AaAb | BbBa
A -> ^
B -> ^

First-set of Given Expression:
S -> ^, a, b,
A -> ^,
B -> ^,

Follow-set of Given Expression:
S -> $,
A -> a, b,
B -> b, a,

Process returned 0 (0x0)   execution time : 70.095 s
Press any key to continue.
```

Output2:

```
"D:\00 Study\SEM 6\0LAB\LT\LAB 7\firstndfolow.exe"
Enter non-terminal: B
Enter number of productions: 2
Enter production 1 : g
Enter production 2 : ^
Enter non-terminal: C
Enter number of productions: 2
Enter production 1 : h
Enter production 2 : ^

Given Input Expression:
S -> ACB | CbB | Ba
A -> da | BC
B -> g | ^
C -> h | ^

First-set of Given Expression:
S -> d, h, ^, b, g, a,
A -> d, g, ^, h,
B -> g, ^,
C -> h, ^,

Follow-set of Given Expression:
S -> $,
A -> h, g, $,
B -> $, a, h, g,
C -> g, $, b, h,

Process returned 0 (0x0)   execution time : 114.869 s
Press any key to continue.
```