

# 18CSC304J/ Compiler Design

Submitted By:- ANANNYA P. NEOG (RA1911003010367)

## Exp-4: Elimination of Ambiguity, Left Recursion and Left Factoring

**Aim:-** To write code for elimination of Ambiguity, Left Recursion and Left Factoring

### Codes:-

- **Elimination of Left Recursion:**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    int n, j, l, i, k;
    int length[10] = {};
    string d, a, b, flag;
    char c;
    cout<<"Enter Parent Non-Terminal: ";
    cin >> c;
    d.push_back(c);
    a += d + "'->";
    d += "->";
    b += d;
    cout<<"Enter productions: ";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout<<"Enter Production ";
        cout<<i + 1<<" :";
        cin >> flag;
        length[i] = flag.size();
        d += flag;
        if (i != n - 1)
        {
            d += "|";
        }
    }
    cout<<"The Production Rule is: ";
    cout<<d<<endl;
    for (i = 0, k = 3; i < n; i++)
    {
        if (d[0] != d[k])
        {
            cout<<"Production: "<< i + 1;
            cout<<" does not have left recursion.";
            cout<<endl;
            if (d[k] == '#')
            {
                b.push_back(d[0]);
                b += "\"";
            }
        }
    }
}
```

```

        else
        {
            for (j = k; j < k + length[i]; j++)
            {
                b.push_back(d[j]);
            }
            k = j + 1;
            b.push_back(d[0]);
            b += "\\|";
        }
    }
    else
    {
        cout<<"Production: "<< i + 1 ;
        cout<< " has left recursion";
        cout<< endl;
        if (d[k] != '#')
        {
            for (l = k + 1; l < k + length[i]; l++)
            {
                a.push_back(d[l]);
            }
            k = l + 1;
            a.push_back(d[0]);
            a += "\\|";
        }
    }
}
a += "#";
cout << b << endl;
cout << a << endl;
return 0;
}

```

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main()
5 {
6     int n, j, l, i, k;
7     int length[10] = {};
8     string d, a, b, flag;
9     char c;
10    cout<<"Enter Parent Non-Terminal: ";
11    cin >> c;
12    d.push_back(c);
13    a += d + "\\->";
14    d += "->";
15    b += d;
16    cout<<"Enter productions: ";
17    cin >> n;
18    for (int i = 0; i < n; i++)
19    {
20        cout<<"Enter Production ";
21        cout<<i + 1<<" :";
22        cin >> flag;
23        length[i] = flag.size();
24        d += flag;
25        if (i != n - 1)
26        {
27            d += "|";
28        }
29    }
30    cout<<"The Production Rule is: ";
31    cout<<d<<endl;
32    for (i = 0, k = 3; i < n; i++)
33    {
34        if (length[i] > 0)
35        {
36            // ... (the logic from the previous block)
37        }
38    }
39    return 0;
40 }

```

```
main.cpp
33 {
34     if (d[0] != d[k])
35     {
36         cout<<"Production: "<< i + 1;
37         cout<<" does not have left recursion.";
38         cout<<endl;
39         if (d[k] == '#')
40         {
41             b.push_back(d[0]);
42             b += "\'";
43         }
44         else
45         {
46             for (j = k; j < k + length[i]; j++)
47             {
48                 b.push_back(d[j]);
49             }
50             k = j + 1;
51             b.push_back(d[0]);
52             b += "\'";
53         }
54     }
55     else
56     {
57         cout<<"Production: "<< i + 1;
58         cout<<" has left recursion";
59         cout<< endl;
60         if (d[k] != '#')
61         {
62             for (l = k + 1; l < k + length[i]; l++)
63             {
64                 a.push_back(d[l]);
65             }
66             k = l + 1;
67             a.push_back(d[0]);
68             a += "\'";
69         }
70     }
71 }
72
73 a += "#";
74 cout << b << endl;
75 cout << a << endl;
76 return 0;
77 }
```

## Output:-

```
input
Enter Parent Non-Terminal: A
Enter productions: 2
Enter Production 1 :A+B
Enter Production 2 :B
The Production Rule is: A->A+B|B
Production: 1 has left recursion
Production: 2 does not have left recursion.
A->BA'|
A'->+BA'|#

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

- **Left Factoring:**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    int n,j,l,i,m;
    int len[10] = {};
    string a, b1, b2, flag;
    char c;
    cout << "Enter the Parent Non-Terminal : ";
    cin >> c;
    a.push_back(c);
```

```

b1 += a + "\\'->";
b2 += a + "\\\''->";
a += "->";
cout << "Enter total number of productions : ";
cin >> n;
for (i = 0; i < n; i++)
{
    cout << "Enter the Production " << i + 1 << " : ";
    cin >> flag;
    len[i] = flag.size();
    a += flag;
    if (i != n - 1)
    {
        a += "|";
    }
}
cout << "The Production Rule is : " << a << endl;
char x = a[3];
for (i = 0, m = 3; i < n; i++)
{
    if (x != a[m])
    {
        while (a[m++] != '|');
    }
    else
    {
        if (a[m + 1] != '|')
        {
            b1 += "|" + a.substr(m + 1, len[i] - 1);
            a.erase(m - 1, len[i] + 1);
        }
        else
        {
            b1 += "#";
            a.insert(m + 1, 1, a[0]);
            a.insert(m + 2, 1, "\\");
            m += 4;
        }
    }
}
}
char y = b1[6];
for (i = 0, m = 6; i < n - 1; i++)
{
    if (y == b1[m])
    {
        if (b1[m + 1] != '|')
        {
            flag.clear();
            for (int s = m + 1; s < b1.length(); s++)
            {
                flag.push_back(b1[s]);
            }
            b2 += "|" + flag;
            b1.erase(m - 1, flag.length() + 2);
        }
        else
    }
}

```

```

        {
            b1.insert(m + 1, 1, b1[0]);
            b1.insert(m + 2, 2, "\\");
            b2 += "#";
            m += 5;
        }
    }
}
b2.erase(b2.size() - 1);
cout << "\nAfter Left Factoring : " << endl;
cout << a << endl;
cout << b1 << endl;
cout << b2 << endl;
return 0;
}

```

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main()
5 {
6     int n,j,l,i,m;
7     int len[10] = {};
8     string a, b1, b2, flag;
9     char c;
10    cout << "Enter the Parent Non-Terminal : ";
11    cin >> c;
12    a.push_back(c);
13    b1 += a + "'->";
14    b2 += a + "\\'->";
15    a += "->";
16    cout << "Enter total number of productions : ";
17    cin >> n;
18    for (i = 0; i < n; i++)
19    {
20        cout << "Enter the Production " << i + 1 << " : ";
21        cin >> flag;
22        len[i] = flag.size();
23        a += flag;
24        if (i != n - 1)
25        {
26            a += "|";
27        }
28    }
29    cout << "The Production Rule is : " << a << endl;
30    char x = a[3];
31    for (i = 0, m = 3; i < n; i++)
32    {
33        if (x != a[m])
34        {
35            while (a[m++] != '|');
36        }
37    }
38 }

```

```
main.cpp
37 else
38 {
39     if (a[m + 1] != '|')
40     {
41         b1 += "|" + a.substr(m + 1, len[i] - 1);
42         a.erase(m - 1, len[i] + 1);
43     }
44     else
45     {
46         b1 += "#";
47         a.insert(m + 1, 1, a[0]);
48         a.insert(m + 2, 1, '\\');
49         m += 4;
50     }
51 }
52 }
53 char y = b1[6];
54 for (i = 0, m = 6; i < n - 1; i++)
55 {
56     if (y == b1[m])
57     {
58         if (b1[m + 1] != '|')
59         {
60             flag.clear();
61             for (int s = m + 1; s < b1.length(); s++)
62             {
63                 flag.push_back(b1[s]);
64             }
65             b2 += "|" + flag;
66             b1.erase(m - 1, flag.length() + 2);
67         }
68         else
69         {
70             b1.insert(m + 1, 1, b1[0]);
71             b1.insert(m + 2, 2, '\\');
72             b2 += "#";
73             m += 5;
74         }
75     }
76 }
77 b2.erase(b2.size() - 1);
78 cout << "\nAfter Left Factoring : " << endl;
79 cout << a << endl;
80 cout << b1 << endl;
81 cout << b2 << endl;
82 return 0;
83 }
```

## Output:-

```
input
Enter the Parent Non-Terminal : a
Enter total number of productions : 4
Enter the Production 1 : a
Enter the Production 2 : aX
Enter the Production 3 : (X)
Enter the Production 4 : aX+X
The Production Rule is : a->a|aX|(X)|aX+X

After Left Factoring :
a->aa'| (X)
a' -> #|Xa''
a'' -> #|+X

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