

## Implementation of Bottom-Up (Shift-Reduce) Parsing in C++

August 15, 2015 Ankur Mhatre Leave a comment

```
#include<conio.h>
#include<iostream.h>
#include<string.h>

struct grammer{
    char p[20];
    char prod[20];
}g[10];

void main()
{
    int i,stpos,j,k,l,m,o,p,f,r;
    int np,tspos,cr;

    cout<<"\nEnter Number of productions:";
    cin>>np;

    char sc,ts[10];

    cout<<"\nEnter productions:\n";
    for(i=0;i<np;i++)
    {
        cin>>ts;
        strncpy(g[i].p,ts,1);
        strcpy(g[i].prod,&ts[3]);
    }

    char ip[10];

    cout<<"\nEnter Input:";
    cin>>ip;

    int lip=strlen(ip);

    char stack[10];

    stpos=0;
    i=0;

    //moving input
    sc=ip[i];
    stack[stpos]=sc;
    i++;stpos++;

    cout<<"\n\nStack\tInput\tAction";
    do
    {
        r=1;
        while(r!=0)
```

```
        {
            cout<<"\n";
            for(p=0;p<stpos;p++)
            {
                cout<<stack[p];
            }
            cout<<"\t";
            for(p=i;p<lip;p++)
            {
                cout<<ip[p];
            }

            if(r==2)
            {
                cout<<"\tReduced";
            }
            else
            {
                cout<<"\tShifted";
            }
            r=0;

            //try reducing
            getch();
            for(k=0;k<stpos;k++)
            {
                f=0;

                for(l=0;l<10;l++)
                {
                    ts[l]='\0';
                }

                tspos=0;
                for(l=k;l<stpos;l++) //removing first caharcter
                {
                    ts[tspos]=stack[l];
                    tspos++;
                }

                //now compare each possibility with production
                for(m=0;m<np;m++)
                {
                    cr = strcmp(ts,g[m].prod);

                    //if cr is zero then match is found
                    if(cr==0)
                    {
                        for(l=k;l<10;l++) //removing matched part from stack
                        {
                            stack[l]='\0';
                            stpos--;
                        }

                        stpos=k;

                        //concatinate the string
                        strcat(stack,g[m].p);
                        stpos++;
                        r=2;
                    }
                }
            }
        }

        //moving input
        sc=ip[i];
        stack[stpos]=sc;
        i++;stpos++;

    }while(strlen(stack)!=1 && stpos!=lip);

    if(strlen(stack)==1)
    {
        cout<<"\n String Accepted";
    }

    getch();
}

/* OUTPUT


Enter Number of productions:4

Enter productions:
E->E+E
E->E*E
E->(E)
E->a

Enter Input:(a+a)*a
```

```
142
143
144  Stack   Input   Action
145  (       a+a)*a  Shifted
146  (a      +a)*a   Shifted
147  (E      +a)*a   Reduced
148  (E+     a)*a    Shifted
149  (E+a    )*a     Shifted
150  (E+E    )*a     Reduced
151  (E      )*a     Reduced
152  (E)     *a      Shifted
153  E       *a      Reduced
154  E*      a       Shifted
155  E*a             Shifted
156  E*E             Reduced
157  E               Reduced
158   String Accepted
159
160  */
```

**Related**

Implementation of K-Nearest Neighbors Algorithm in C++
June 11, 2015
In "Snippets"

Implementing Round Robin Scheduling Algorithm in C++
May 29, 2015
In "Snippets"

Implementing Lexical Analyser in C++
August 14, 2015
In "Snippets"

C++SPCC

## Post navigation

Previous PostImplementing Lexical Analyser in C++Next PostPlotting Liner and Circular Convolution with MATLAB

### Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.

## Subscribe to Blog via Email

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 34 other subscribers

Email Address

Subscribe

## Discussions

- Madhu A on Implementation of Apriori Algorithm in C++
- Madhu A on Implementation of Apriori Algorithm in C++
- robocraft hack on Google Input Tools : Chrome Extension
- neelam jat on To Implement Character Generation by using Bitmap Method in C++
- Mounika on 8086 Assembly Program to Sort Numbers in Ascending Order

## {developer} > Java > PHP > WordPress > HTML-CSS-JS

- Snippets
- Interesting Resources
- Symfony 2
- WordPress
- Struts 2

- Git

## Recent

- A simple presentation template in CSS and jQuery by CodyHouse December 19, 2015
- Struts 2 Hello World Example (XML Version) October 22, 2015
- Breadcrumbs & Multi-Step Indicator October 22, 2015
- Implementing JPEG Algorithm in Java September 15, 2015
- Implementing Run Length Encoding in Java September 14, 2015

## Site Meta

- Home
- Privacy Policy
- RSS Feeds
- Sitemap

## Top

- Implementing Round Robin Scheduling Algorithm in C++
- 8086 Assembly Program for Addition of Two 8 bit Numbers
- 8086 Assembly Program to Sort Numbers in Ascending Order
- Decompile, Edit, and Recompile Android Apps with APK Studio
- 8086 Assembly Program to Find Largest Number from Given Numbers