Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

CSE 4130
Formal Languages & Compilers Lab
Session: 05
Assignment: 05

Submitted By:

Name: Syed Sanzam
ID: 16.01.04.042
Lab Group: A2

Date of Submission: **September 26, 2019**

Questions:

1. Implement the following CFG, in the way shown above or with the help of a user defined stack, that is, in the way it may work in an implementation of the transition function of a PDA.

$$A \rightarrow aXd$$
$$X \rightarrow bbX$$
$$X \rightarrow bcX$$
$$X \rightarrow \varepsilon$$

2. Implement the CFG for generating simple arithmetic expressions.

Solution 1:

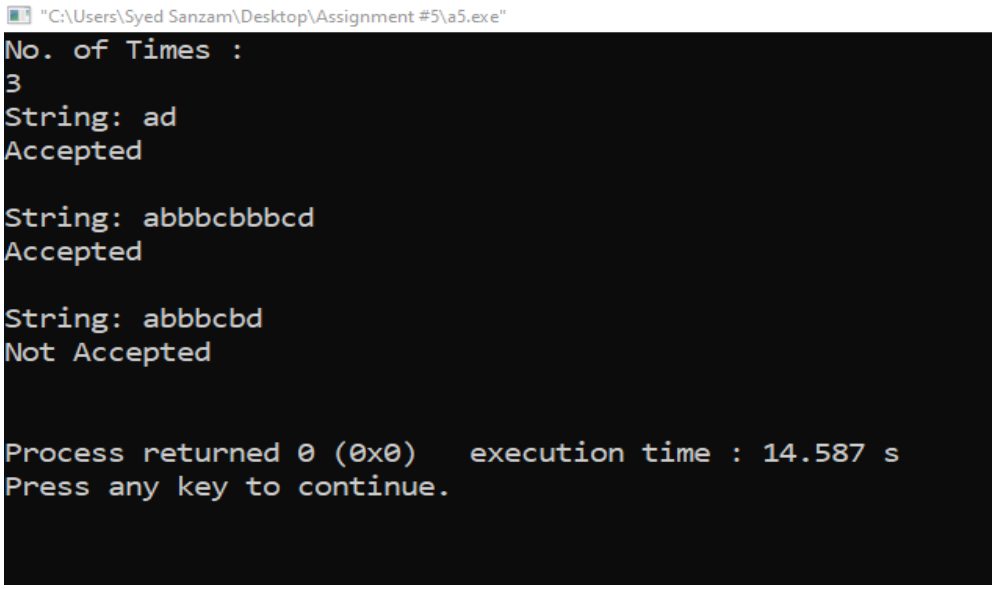
Source Code:

```
/*
    AUTHOR: Syed Sanzam
    Topic : Context Free Grammar
    DATE: 23.09.19

    The Code follows the following CFG,
        A --> a X d
        X --> b b X
        X --> b c X
        X --> Empty
*/


#include <bits/stdc++.h>

using namespace std;


int len;
int read = 0;

bool X(string str)
{
    bool checkLastChar = true;
    while(read < len - 1)
    {
        if(str[read] == 'b')
        {
            if(str[read + 1] == 'b')
            {
                read = read + 2;
            }
            else if(str[read + 1] == 'c')
            {
```

```
                read = read + 2;
            }
            else
            {
                checkLastChar = false;
                break;
            }
        }

        else
        {
            checkLastChar = false;
            break;
        }
    }
    return checkLastChar;
}

void A(string str)
{
    len = str.length();
    if(str[0] == 'a')
    {
        read++;
        if(len == 2)
        {
            if(str[len - 1] == 'd')
            {
                cout << "Accepted" << endl;
            }
            else
            {
                cout << "Not Accepted" << endl;
            }
        }
        else
        {
            if(X(str))
            {
                if(str[len - 1] == 'd')
                {
                    cout << "Accepted" << endl;
                }
                else
                {
                    cout << "Not Accepted" << endl;
                }
            }
            else
            {

                cout << "Not Accepted" << endl;
            }
        }
    }
}

void init()
{
    int noOfTimes = 0;
    string s = " ";
```

```
96.
97.     cout << "No. of Times :" << endl;
98.     cin >> noOfTimes;
99.
100.            for(int i = 0; i < noOfTimes; i++)
101.            {
102.                cout << "String: ";
103.                cin >> s;
104.                A(s);
105.                len = 0;
106.                read = 0;
107.                cout << endl;
108.            }
109.        }
110.
111.        int main()
112.        {
113.            init();
114.            return 0;
115.        }
```

Inputs and Outputs:



Solution 2:

Source Code:

```
1.  /*
2.      AUTHOR: Syed Sanzam
3.      Topic : Context Free Grammar (Evaluation of Arithmetic Expression)
4.      DATE: 25.09.19
5.
6.      The Code follows the following CFG,
7.
8.          E -> E + T | E - T | T
```

```
9.
10.         T -> T * F | T / F | F
11.
12.         F -> (E) | n
13. */
14.
15. #include <bits/stdc++.h>
16.
17. using namespace std;
18.
19. int len;
20. int read = 0;
21. bool check;
22.
23. void factor(string s);
24.
25. bool isNumber(string s)
26. {
27.     bool isNum = false;
28.     while(true)
29.     {
30.         if(isdigit(s[read]))
31.         {
32.             read++;
33.             isNum = true;
34.         }
35.
36.         else
37.             break;
38.     }
39.
40.     return isNum;
41.
42. }
43.
44. bool isChar(string s)
45. {
46.     bool isChar = false;
47.     while(true)
48.     {
49.         if(isalpha(s[read]))
50.         {
51.             read++;
52.             isChar = true;
53.         }
54.
55.         else
56.             break;
57.     }
58.
59.     return isChar;
60.
61. }
62. void term(string s)
63. {
64.     factor(s);
65.     if(read < len)
66.     {
67.         if(check)
68.         {
```

```
            if(s[read] == '+'||s[read] == '-'||s[read] == '*'||s[read] == '/' || s[read] == '=')
            {
                check = true;
                read++;
                term(s);
            }
            else
            {
                check = false;
                return;
            }
        }
    }
    else
        return;
}

void factor(string s)
{
    check = false;
    if(read < len)
    {
        if(isNumber(s) || isChar(s))
        {
            check = true;
            return;
        }
        else if(s[read] == '(')
        {
            read++;
            term(s);
                    if(s[read] == ')')
                    {
                        read++;
                        check = true;
                        return;
                    }
                    else
                        check = false;
                }

                else
                {
                    read++;
                    check = false;
                    return ;
                }
            }
            else
                return;
        }

        void expression(string s)
        {
            if(s.length() > 1)
            {
                term(s);
                if(check)
                    cout << "Accepted!" << endl;

```

```
129.                else
130.                    cout << "Not Accepted!" << endl;
131.            }
132.            else
133.                cout << "Accepted!" << endl;
134.        }
135.
136.        void init()
137.        {
138.            int noOfTime = 0;
139.            cout << "No. of Times: " << endl;
140.            cin >> noOfTime;
141.            for(int i = 0; i < noOfTime; i++)
142.            {
143.                string s;
144.                cout << "Expression: " << endl;
145.                cin >> s;
146.                len = s.length();
147.                expression(s);
148.
149.                read = 0;
150.            }
151.        }
152.
153.        int main()
154.        {
155.            init();
156.            return 0;
157.        }
```

Inputs and Outputs: