Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Formal Languages and Compiler Lab

CSE 4130

Session - Spring 2020

Assignment 05

Submitted To

Md. Aminur Rahman
Assistant Professor
CSE, AUST

Submitted By

Fairuz Nawar
17.01.04.122
Group - C1

March 22, 2021

Contents

1. Question 01

Implement the following CFG in the way shown above.
A → aXd X → bbX X → bcX X → ϵ

1.1 Input File :

input.txt

```
ad
abbd
abbbbd
abcd
abcbbd
```

1.2 Answer Program :

```
#include<stdio.h>
#include<string.h>
char s[30];
int len,i=0,f=0;

void A()
{
    if(s[i] == 'a')
    {
        i++;
        X();
        if(f==1)
        {
            if(s[i-1] == 'd')
                f = 1;
            else
                 f = 0;

            return;
        }
    }
}

void X()
{
    if(len-1 == i)
    {
        i++;
        f = 1;
        return;
    }
    else
    {
        if(s[i] == 'b')
        {
            i++;
            if(s[i] == 'b' || s[i] == 'c')
            {
                i++;
                X();

            }
        }
        else
        {
            f = 0;
            return;
        }
    }
}

int main(void)
{
   printf("CFG: \n");
    printf("A -> aXd\nX -> bbX | bcX | epsilon\n\n\n");

   char c[100];
   FILE *p1,*p2;
   p2 = fopen("output.txt","w");
   if ((p1 = fopen("input.txt","r")) == NULL){
       printf("File Not Found.");
       exit(1);
   }
   while(fscanf(p1,"%s",&s)>0)
   {
        i = 0;
        len = strlen(s);
        if(len>=1)
            {
                A();
            }
        else{
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
        }

        if(len == i && f == 1){
            fprintf(p2,"The String %s is Valid.\n",s);
            printf("The String %s is Valid.\n",s);
        }

         else{
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
         }

   }
   fclose(p1);
   fclose(p2);
   return 0;
}
```

1.3 Output File :

output.txt

```
The String ad is Valid.
The String abbd is Valid.
The String abbbbd is Valid.
The String abcd is Valid.
The String abcbbd is Valid.
```

2. Question 02

<Exp> → <Term> + <Term> | <Term> - <Term> | <Term>

<Term>→<Factor> ∗ <Factor> | <Factor> / <Factor> | <Factor>

<Factor>→(<Exp>) | ID | NUM

ID → a|b|c|d|e

NUM→ 0|1|2|...|9

Non-terminal symbols:

<Exp>, <Term>, <Factor>

Terminal symbols:

+, -, ∗, /, (,), a, b, c, d,e, 0, 1, 2, 3, ..., 9

Start symbol:

<Exp>

Implement the CFG shown above for generating simple arithmetic expressions.

2.1 Input File :

input.txt

```
(a+b)∗(a-2)
(7+b)/(4-2)
a+6
a∗7
b/4
(7+2)∗b
```

2.2 Answer Program :

```
#include<stdio.h>
#include<string.h>

void Factor(void);
void Term(void);
void Exp(void);


int i=0,f=0,l;

char s[10];

void Term()
{
    Factor();
    if(f && i<l && (s[i]=='*' || s[i]=='/'))
    {
        i++;
        Factor();
    }

}

void Factor()
{
    if(i<l && s[i]=='(')
    {
        i++;
        f=1;
        Exp();
        if(f && s[i]==')')
            i++;
        else
            f=0;

    }

    else if(i<l && s[i]>='0' && s[i]<='9')
    {
        i++;
        f=1;
    }
    else if(i<l && s[i]>='a' && s[i]<='e')
    {
        i++;
        f=1;
    }
    else
```

```
        f=0;
}

void Exp()
{
    Term();
    if(f && i<l &&(s[i]=='+' || s[i]=='-'))
    {
        i++;
        Term();
    }

}

int main(void)
{
    printf("\nCFG:\n");
    printf("Exp -> Term+Term | Term-Term | Term \nTerm -> Factor*Factor | Factor/Fa
    char c[100];
    FILE *p1,*p2;
    p2 = fopen("output.txt","w");
    if ((p1 = fopen("input.txt","r")) == NULL)
    {
        printf("File Not Found.");
        exit(1);
    }
    while(fscanf(p1,"%s",&s)>0)
    {
        i = 0;
        l = strlen(s);
        if (l>=1)
            Exp();
        else{
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
        }

        if (l == i && f ){
            fprintf(p2,"The String %s is Valid.\n",s);
            printf("The String %s is Valid.\n",s);
        }

        else{
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
        }

    }

    fclose(p1);
    fclose(p2);
    return 0;

}
```

2.3 Output File :

```
The String (a+b)*(a-2) is Valid.
The String (7+b)/(4-2) is Valid.
The String a+6 is Valid.
The String a*7 is Valid.
The String b/3 is Valid.
The String (7+2)*b is Valid.
```

3. Question 03

Implement the following grammar in C.
<stat>→<asgn_stat>|<dscn_stat>|<loop_stat>
<asgn_stat>→id = <expn>
<expn>→<smpl_expn> <extn>
<extn>→<relop> <smpl_expn> | epsilon
<dcsn_stat>→ if (<expn>) <stat> <extn1>
<extn1>→ else <stat> | epsilon
<loop_stat>→while (<expn>) <stat> | for (<asgn_stat> ; <expn> ; <asgn_stat>) <stat>
<relop>→ ==|!=|<=|>=|>|<

3.1 Input File :

```
input.txt

a=2
if(a+b)d=a+b
a=(a+3)>=1-2
if(e)
if(a+d)b
while(a+b)
while(1)
while(b)
while(a<b)
```

3.2 Answer Program :

```
#include<stdio.h>
#include<string.h>
```

```c
#include <stdbool.h>

void Factor(void);
void Term(void);
void Exp(void);
void stat();

int f=0,i=0,l;
char s[100];

void Term()
{
    Factor();
    if(f && i<l && (s[i]=='*' || s[i]=='/'))
    {
        i++;
        Factor();
    }

}


void Factor()
{
    if(i<l && s[i]=='(')
    {
        i++;
        f=1;
        Exp();
        if(f && s[i]==')')
        {
            i++;
        }
        else
        {
            f=0;
        }

    }
    else if(i<l && s[i]>='a' && s[i]<='e')
    {
        i++;
        f=1;
    }
    else if(i<l && s[i]>='0' && s[i]<='9')
    {
        i++;
        f=1;
    }
    else
```

```
        f=0;
}


void Exp()
{
    Term();
    if(f && i<l &&(s[i]=='+' || s[i]=='-'))
    {
        i++;
        Term();
    }

}


void extn()
{
    if(f && relop())
    {
        Exp();
    }
}


void expn()
{
    Exp();
    if(f)
    {
        extn();
    }
}

bool relop()
{
    if(f && i<l && (s[i]=='=' || s[i]=='!'))
    {
        i++;
        if(i<l && s[i]=='=')
        {
            i++;
            return true;
        }


        else
            return false;
    }
    else if(f && i<l && (s[i]=='<' || s[i]=='>'))
    {
        i++;
        if(i<l && s[i]=='=')
        {
            i++;
        }

        return true;
    }
    return false;
}


void extn1()
{
    if(f && i+3<l && s[i]=='e' && s[i+1]=='l' && s[i+2]=='s' && s[i+3]=='e')
    {
        i=i+4;
        stat();
    }
}


void dscn_stat()
{
    if(i+1<l && s[i]=='i' && s[i+1]=='f')
    {
        i=i+2;
        if(s[i]=='(')
        {
            i++;
            expn();
            if(f && i<l && s[i]==')')
            {
                i++;
                stat();
                if(f)
                {
                    extn1();
                }
            }
            else
                f=0;
        }
        else
            f=0;
    }
}

void asgn_stat()
```

```
{
    if(i<l && 'a'<=s[i] && s[i]<='e')
    {
        i++;
        if(s[i]=='=')
        {
            i++;
            expn();
        }
    }
}


void loop_stat()
{
    if(i+4<l && s[i]=='w' && s[i+1]=='h' && s[i+2]=='i' && s[i+3]=='l' && s[i+4]=='
    {
        i=i+5;
        if(i<l && s[i]=='(')
        {
            i++;
            expn();
            if(f && i<l && s[i]==')')
            {
                i++;
                stat();
            }
            else
                f=0;
        }
        else
            f=0;
    }

    else if(i+2<l && s[i]=='f' && s[i+1]=='o' && s[i+2]=='r')
    {
        i++;
        if(i<l && s[i]=='(')
        {
            i++;
            expn();
            i++;
            if(s[i]==';')
            {
                i++;
                expn();
                i++;
                if(s[i]==';')
                {
                    i++;
                    expn();
                    if(f && i<l && s[i]==')')
                    {
                        i++;
                        stat();
                    }
                }
            }
        }
        else
            f=0;

    }

}


void stat()
{
    asgn_stat();
    dscn_stat();
    loop_stat();
}

int main()
{
    printf("\nCFG:");
    printf("\n<stat> -> <asgn_stat> | <dscn_stat> | <loop_stat>");
    printf("\n<asgn_stat> -> id = <expn>");
    printf("\n<expn> -> <smpl_expn> <extn>");
    printf("\n<extn> -> <relop> <smpl_expn> | epsilon");
    printf("\n<dcsn_stat> -> if (<expn> ) <stat> <extn1>");
    printf("\n<extn1> -> else <stat> | epsilon");
    printf("\n<loop_stat> -> while (<expn>) <stat> | for (<asgn_stat> ; <expn> ; <a
    printf("\n<relop> -> == | != | <= | >= | > | <\n\n");

    char c[100];
    FILE *p1,*p2;
    p2 = fopen("output.txt","w");
    if ((p1 = fopen("input.txt","r")) == NULL)
    {
        printf("File Not Found.");
        exit(1);
    }

    while(fscanf(p1,"%s",&s)>0)
    {
        strtok(s, "\n");
        i = 0;
        l = strlen(s);
        if (l>=1)
        {
            stat();
        }
        else
        {
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
        }
        if (l == i && f )
        {
            fprintf(p2,"The String %s is Valid.\n",s);
            printf("The String %s is Valid.\n",s);
        }
        else
        {
            fprintf(p2,"The String %s is Not Valid.\n",s);
            printf("The String %s is Not Valid.\n",s);
        }
        strcpy(s,"");

    }

    fclose(p1);
    fclose(p2);
    return 0;
}
```

3.3 Output File :

output.txt

```
The String a=2 is Valid.
The String if(a+b)d=a+b is Valid.
The String a=(a+3)>=1-2 is Valid.
The String if(e) is Valid.
The String if(a+d)b is Valid.
The String while(a+b) is Valid.
The String while(1) is Valid.
The String while(b) is Valid.
The String while(a<b) is Valid.
```