Design Of Compilers

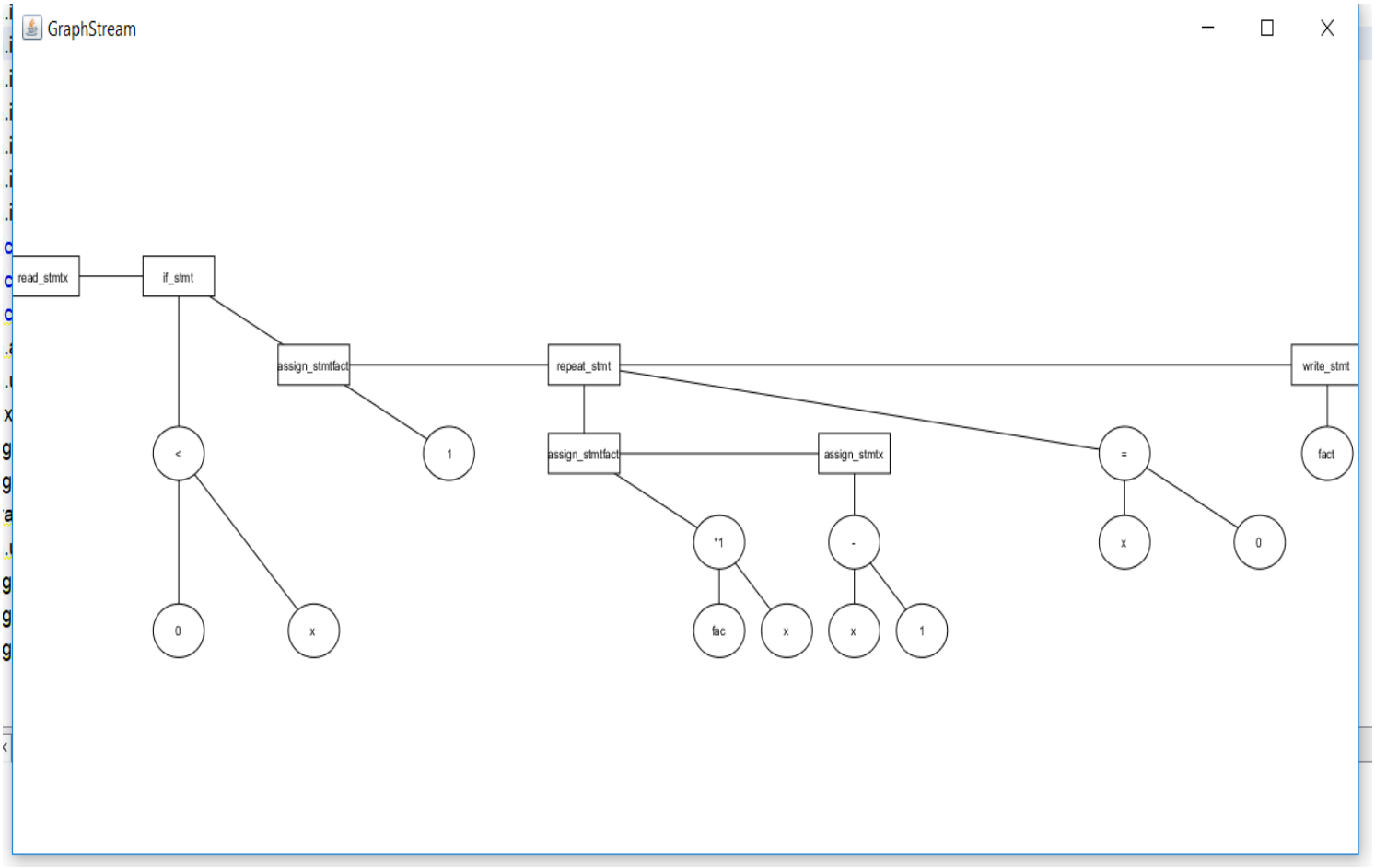
CSE 226

Parser Project

Submitted by:

Engy Samy Salah
(16P3004)
Mayar Wessam Nour
(16P3008)

Parser Tree:


GraphStream
read_stmtx
if_stmt
assign_stmtfact
repeat_stmt
write_stmt
<
1
assign_stmtfact
assign_stmtx
=
fact
0
x
*1
-
0
x
fac
x
x
1

Parser Output File:

```
Program Found
stmt_sequence Found
statement Found
read stmt
Semi Colon Found
statement Found
If-statement Found
If
Exp Found
Simple_exp Found
Term Found
Factor Found
Digit
Comparison stmt
<
Simple_exp Found
Term Found
Factor Found
ID
Then
stmt_sequence Found
statement Found
Assignment_stmt
Exp Found
Simple_exp Found
Term Found
Factor Found
Digit
Semi Colon Found
statement Found
Repeat_statement Found
stmt_sequence Found
statement Found
Assignment_stmt
Exp Found
Simple_exp Found
Term Found
Factor Found
ID
Mulop Found
*
Factor Found
ID
Mulop Found
*
Factor Found
ID
Semi Colon Found
statement Found
Assignment_stmt
Exp Found
Simple_exp Found
Term Found
Factor Found
ID
Addop Found
-
Term Found
Factor Found
Digit
Until Found
Exp Found
Simple_exp Found
Term Found
Factor Found
ID
Comparison stmt
=
Simple_exp Found
Term Found
Factor Found
Digit
Semi Colon Found
statement Found
Write stmt
Exp Found
Simple_exp Found
Term Found
Factor Found
ID
Successfull End if
```

If Statement Function In Parser:

```
456     public static void if_stmt() throws FileNotFoundException, UnsupportedEncodingException
457     {
458        file("If-statement Found",x);
459        ifs=true;
460          match(token[x],"if");
461          file("If",x);
462          if(!repeat)
463     {
464      node if_stmt=new node();
465           counter++;
466             stmt.al.add(if_stmt);
467             stmt.a2.add("if_stmt");
468     }
469     else
470     {
471         node if_stmt=new node();
472           counter1++;
473             stmt.al.get(counter).al.add(if_stmt);
474             stmt.al.get(counter).a2.add("if_stmt");
475     }
476           temp=counter1;
477           counter1=-1;
478           exp();
479
480     counter1=temp;
481           if(!match(token[x],"then"))
482           {
483           file("Expected Then",x);
484           }
```

Assignment Function in Parser:

```
527     public static void reapet_stmt() throws FileNotFoundException, UnsupportedEncodingException
528     {...34 lines }
562     public static void assign_stmt() throws FileNotFoundException, UnsupportedEncodingException
563     {
564         file("Assignment_stmt",x);
565       match( Type[x],"ID");
566       if(!ifs && !repeat)
567     {
568      node assgin_stmt=new node();
569           counter++;
570             stmt.al.add(assgin_stmt);
571             stmt.a2.add("assgin_stmt"+ token[x-1]);
572     }
573     else
574     {
575         node assign_stmt=new node();
576           counter1++;
577             stmt.al.get(counter).al.add(assign_stmt);
578             stmt.al.get(counter).a2.add("assign_stmt"+token[x-1]);
579     }
580       match(token[x],":=");
581       temp=counter1;
582       counter1=-1;
583       exp();
584       counter1=temp;
585
586       //file("Assignment stmt",x);
587     }
588     public static void read_stmt() throws FileNotFoundException, UnsupportedEncodingException
589     {...26 lines }
```

Term Function In Parser:

```
public static void simple_exp() throws FileNotFoundException, UnsupportedEncodingException{...16 lin
public static void addop() throws FileNotFoundException, UnsupportedEncodingException
{...43 lines }
public static void term() throws FileNotFoundException, UnsupportedEncodingException{
   file("Term Found",x);
factor();
 while(token[x].equals("*")||token[x].equals("/"))
   {
      mulop();

     stmt.al.get(counter).al.get(counter1).al.add(stmt.al.get(counter).al.get(counter1-1));
      stmt.al.get(counter).al.get(counter1).a2.add(stmt.al.get(counter).a2.get(counter1-1));
   factor();

   stmt.al.get(counter).al.get(counter1).al.add(stmt.al.get(counter).al.get(counter1));
   stmt.al.get(counter).al.get(counter1).a2.add(stmt.al.get(counter).a2.get(counter1));

   }

}

public static void mulop() throws FileNotFoundException, UnsupportedEncodingException
```

Node Class:

```
class node {
 ArrayList<node> al=new ArrayList<node>();
 ArrayList<String> a2=new ArrayList<String>();

 node(){
 }
public ArrayList<String> getName() {
      return a2;
   }

}
```

DrawTree Function(array 1D containing all the nodes)

```
238    public static void drawtree(String s [] ,int i)
239    {
240
241
       String Symbol []=new String [] {"+","-","=","*","/","<"};
243    int mayar=0;
244
245    while(ind<i)
246    {
247
248
249        if(ind<i&&s[ind].substring(0,1).equals("w") )
250        {
251        tree[ind]=s[ind];
252        ind++;
253        tree[ind]=s[ind];
254        ind++;
255
256        }
257        else if(ind<i &&s[ind].substring(0,1).equals("a"))
258        {
259           tree[ind]=s[ind];
260        ind++;
261           while(ind<i&&!s[ind].substring(0, 1).equals("u")&&!s[ind].substring(0, 1).equals("a")&&!s[ind].substring(0, 1).equals("w")&&!s[ind].substring(0, 1).equals("r")&&!s[ind].substring(0, 1).equals("i"))
262           {
263                       tree[ind]=s[ind];
264                        ind++;
265                      while(ind<i&&!s[ind].substring(0, 1).equals("u")&&!s[ind].substring(0, 1).equals("a")&&!s[ind].substring(0, 1).equals("w")&&!s[ind].substring(0, 1).equals("r")&&!s[ind].substring(0, 1).equals
266                     {
267                       tree[ind]=s[ind+1];
```

DrawApi(drawing the array of nodes using an API):

```
887     public static void DrawApi(Graph g)
888     {
889         String []stmtt=new String[100];
890         int loop=0;
            String []temp= new String [100];
892         int si=0;
            int eng=0;
894     while(en<index)
895     {
896          if(en<index&&tree[en].substring(0, 3).equals("unt"))
897     {
898         break;
899     }
900
901          if(en<index&&tree[en].substring(0, 3).equals("end"))
902     {
903         break;
904     }
905        if(en<index&&tree[en].substring(0, 3).equals("rea"))
906        {
907            if(xx&&yy)
908           {
909              if(cht<chere&&cht==0)
910              {
911                 stmtt[loop]=tree[en]+id;
912                 loop++;
913                  en++;
914                  id++;
915                   cht++;
916              }
917              else if(chere<cht&&chere==0){
921                  id++;
922                   chere++;
923              }
924              else if(cht<chere&&cht>=1)
925              {
926                 Node dd= g.addNode(tree[en]+id);
927                  dd.addAttribute("layout.frozen");
928         dd.addAttribute("xy", xaxis,yaxis);
929         xaxis+=2;
930                  stmtt[loop]=tree[en]+id;
931               chit[cht]=tree[en]+id;
932               g.addEdge(chit[cht-1]+chit[cht],chit[cht-1],chit[cht]);
933                cht++;
934                  loop++;
935                  id++;
936                  en++;
937              }
938              else if(chere<cht&&chere>=1)
939              {
940                 Node dd=g.addNode(tree[en]+id);
941                  dd.addAttribute("layout.frozen");
942         dd.addAttribute("xy", xaxis,yaxis);
943         xaxis+=2;
944                  stmtt[loop]=tree[en]+id;
945               chrep[chere]=tree[en]+id;
946               g.addEdge(chrep[chere-1]+chrep[chere],chrep[chere-1],chrep[chere]);
947                chere++;
948                  loop++;
949                  id++;
950                  en++;
951              }
```