Comp 2140: Lab Assignment 4: Syntax Directed Translation

1 Submission

You should submit your programs at the submission site (you can click the left highlighted words to go to the submission site). The due date is Friday before the last lab–note that the due time is not the same as the last lab assignment. It is worth 8 points. We will use your last submission for your final marks for this assignment.

You must test your program on our sample inputs and other cases devised by you before submission. Passing the sample inputs will not guarantee a high mark. You must fully understand the grammar and implement that grammar correctly so that your parser can cope with all situations.

Useful links

- Submission site
- Tiny Language Grammar and links for JavaCup and JLex
- Step-by-step instructions

2 Task

You will use JavaCup to translate Tiny programs to Java programs. The translated Java program will be able to run and produce correct result. You need to write a JavaCUP file and a JLex file so that a translator for Tiny language can be generated. We will run the following commands to generate the translator. You need to modify the lex file and cup file in assignment 3.

```
>del *.class A4.java A4.output
>java JLex.Main A4.lex
>java java_cup.Main -parser A4Parser -symbols A4Symbol < A4.cup
>javac A4.lex.java A4Parser.java A4Symbol.java A4User.java
>java A4User
>javac A4.java
>java A4
>more A4.output
```

`A4.lex` and `A4.cup` files are what you are going to write. By running those commands, you will generate the scanner A4Scanner and the parser (or the translator) A4Parser. A4User is listed as below. It invokes the parser and generate the translated java program called A4.java. After A4.java is generated, you compile and run it. A4.class will produce a file called A4.output, which holds the correct result.

```java
import java.io.*;
import java_cup.runtime.Symbol;
class A4User {
  public static void main(String[] args) throws Exception {
    File inputFile = new File ("A4.tiny");
    A4Parser parser= new A4Parser(new A4Scanner(new FileInputStream(inputFile)));
    String javaProgram =(String)parser.debug_parse().value;
    FileWriter fw=new FileWriter(new File("A4.java"));
    fw.write(javaProgram);
    fw.close();
  }
}
```

Even if your program passed all the test cases in A3, it may fail our new test cases. For example,

```
INT main f() BEGIN STRING X:="test"; @#$ END
```

is not a valid TINY program but YOUR parser may think that it is. To catch the error, your lexer need to return some error token when it sees any OTHER token that is not listed.

If your lex and cup files are correct, all of those commands, especially A4User, will run smoothly without any error report. An A4.java program will be generated. A4.java will be compiled without error and the result of A4.java will be written into A4.output. If your programs (lex and cup files) are not correct, during the process there will be some error messages.

Here are a sample A4.tiny file and the corresponding generated A4.java file. The corresponding A4.output file consists of the result f running A4.java. Note that you can generate your own A4.java as long as A4.java can produce the correct result. In this java program A4.java, just the same as in TINY program A4.tiny, it reads integers from A41.input and A42.input. Note that file names such as A41.input, A42.input and A4.output are obtained from the TINY program A4.input. You should not hard code those file names into your javaCup or JLex files.

The main method in TINY program will be translated into a main method in Java program. We assume that there will be only one main method in a TINY program.

Note that you don't need to write any Java programs. The parser and the scanner are generated from your cup and lex specifications. Also, we will test your program on our own data.

3 What to submit

You need to turn in 2 files: the JLex file, named A4.lex, which can be used to generate the scanner; the javaCUP file, named A4.cup, which can be used to generate the translator.

4 Marking scheme

```
yourMark=0;
if ( A4.lex and A4.cup  files not submitted or file names are incorrect) return;
if (A4.lex.java && A4Parser.java && A4Symbol.java are generated and compiled correctly) {
   for (each of the 10 A4.tiny files) {
     if (A4.tiny is valid  and it is a correct program
        && A4.java is generated
        && A4.output contains correct answer)
        // suppose A4.tiny has a write statement that write the result to the file A4.output.
                yourMark+=0.8;
     if (A4.tiny is valid program wrt the grammar but not a correct program
           (such as missing variable declaration)
        && A4.java is generated
        && A4.java has compilation error )
               yourMark+=0.8;
     if (A4.tiny is not a valid Tiny program
        && A4.java is not generated )
               yourMark+=0.8;
    }
 }
if (your submission is late within 24 hours) yourMark=yourMark-2;
if (your submission is late more than 24 hours) yourMark=0;

yourMark+= (size of your lex and cup file<800)? 0.5*800/fileSize:0;
// code length counting is different from previous assignments.
```

Tips for command window

You can type faster in command window if you utilize:

Arrow key use up-arrow to find your previously typed commands. Then type enter to run that command.

Tab key Use Tab to auto-complete file names and command names.