CS 4240

Jiateng Xie
Fu Shen
Wenduo Yang

Project Phase 1 Report

Below are the instructions to build the project using only Tiger.g file and the tool. To run the compiler, you can simply run the jar file provided.

Building code.

To build the project, go to the directory containing Tiger.g and antlr-3.5.2-complete.jar. Then compile the code by executing the following command on the command line/ terminal.

```
java -jar antlr-3.5.2-complete.jar Tiger.g
```

Then TigerParser.java and TigerLexer.java will be generated under the same directory.

Running code.

To run the code, you need to generate a jar file using eclipes. We have already included the jar file in the root directory. To manually output the jar file, first put the two java files (lexer file and parser file) under a directory. Then create a java project at this directory using eclipes. Add a main class. Copy the content of our Main.java to your main class. Right click the project title, select build path, and select Add External Archives. Browse to antlr-3.5.2-complete and select it. Finally export the project as a runnable jar file.

Structure of the code & Location of the test program

The grammar file(Tiger.g), the jar file (Tiger.jar) and the Main.java are in the root directory. The parser file and scanner file are in the 4240phase1 folder. The test program and its generated AST and Parser Tree are in the test folder.

Running jar file

To simply run the jar file to parse a sample program, execute:
java -jar Tiger.jar <path to sample code>
To run the jar file with debug flag(i.e. To check the stream of token types), execute:
java -jar Tiger.jar -debug <path to sample code>

Generating AST

To generate the AST for the sample program, please open the ANTLRWorks jar file included in the folder, use ANTLRWorks to open the Tiger grammar file, click on Debug under the menu and insert the sample program, run Debug and the AST will be generated by ANTLRWorks.

In case it is needed to generate AST in a Dot file, please add the following code to the Main method and delete the code “parser.program()” in line 16 of main file.

:”
```
import org.antlr.runtime.tree.*;
import org.antlr.stringtemplate.*;

CommonTree tree = (CommonTree) parser.program().getTree();
DOTTreeGenerator gen = new DOTTreeGenerator();
StringTemplate st = gen.toDOT(tree);
System.out.println(st);”
```
After adding these codes, you will be able to view the dot representation of the AST through the console by running the code.