CS 335 Semester 2022–2023-II: Project Milestone 1

Group members:
Priyanka Jalan (190649)
Piyush Agarwal (190600)
Gargi Naladkar (200371)

February 28, 2023

Using Lex and Bison, we developed a scanner and parser that accepts any valid Java program with the features listed below as input and outputs a DOT script representing the AST(abstract syntax tree) of the input program. The DOT script, when processed by the Graphviz tool called dot, will produce a postscript file with the diagram of the parse tree. We incorporated the scanner and parser specifications that are stated in the issue statement. The specifications of our implementation are listed below:

Compilation and Execution:

1. Download the cs335 project zip file. Extract the contents.
2. Move to the src directory by using the following command in your teminal.

```
    $ cd milestone1/src
```

3. To generate executable and compile the lexer and parser files, use the following command:

```
    $ make
```

   The above command generates the following files:

   (a) myASTGenerator : an executable file.
   (b) parser.output : descriptions of various states of LR.
   (c) parser.tab.c : parsing program in C
   (d) lex.yy.c : lexer program in C
   (e) parser.tab.h : header file

   Makefile contains the following commands:

   (a) For parsing using ”make” command:

```
    bison -dtv parser.y
    flex lexer.l
    g++ -w lex.yy.c parser.tab.c -o out
```

   (b) For cleaning using ”make clean” command:

```
    rm myASTGenerator
    rm parser.tab.c parser.tab.h parser.output
    rm lex.yy.c
```

4. A typical session for running the test file with address "../tests/test_1.java" and outputing the generated dot file to "out.dot", use the following command:

```
    ./myASTGenerator ../tests/test_1.java --output=test_1.dot
```

5. The following options are supported for running the executable file:

```
usage: ./myASTGenerator [options] file
Options:
--help                To print this message.
--input=<file name>   You can use it to specify the input file name.
                      By default, the name of the input file can be
                                                              specified
                                                              directly.
--output=<file name>  You can use it to specify the output file name.
                      By default, the name of the output file is 'output.
                                                              dot'
Do not use --input=<file name> flag if you specify the input file name
                                        directly.
--verbose             It outputs the stack trace
```

6. To Render the AST generated, use the following command:

```
    dot -Tps out.dot -o image.ps
```

Features:

Our parser parses for all the basic features which were mandatory. The following **optional features** are added to the JAVA parser. Some of the features have corresponding test cases to validate the correctness.

- Support for strings including concatenation
- Multidimensional arrays (testcase 4)
- Try Catch and Throw Statements (testcase 8)
- Do while and switch constructs (testcase 9)
- Record and enum classes (testcase 3)
- Support for <> operator
- Generics (testcase 8)
- Support for lambda expressions (testcase 2)