CS335 : Milestone 1

Arpit Kumar Rai (200190)
Avi Kumar (200229) Kunwar Preet Singh (200536)

March 2023

Tools and Utilities used

- Flex is used for our lexer, which is integrated with the parser. It returns tokens.
- Bison is used for implementing the parser. Official grammar is rewritten in a way that there is no conflict in lr(1) parsing.
- Make utility is used for automatic tracking for files and compilation.

Features implemented

- We have implemented all the basic features in the description.
- Support for Strings, including a few operations like concatenation, support for printing with println().
- Support for Interfaces
- Static polymorphism via method overloading
- Dynamic polymorphism via method overriding (using @Override as well)
- import statements like import java.util.*
- Class and Interface inheritance
- Exception type

Instructions for Compilation

- Please change directory into *./milestone1/src/*
- Execute *make*

- Execute *./myASTGenerator - -input ./path/input.txt - -output ./path/output.dot* to give input and output.

- Note that - -input should be space separated from input path, as should be - -output from output path.

- The generated output is dot file. This dot file can be used to generate visual ast using graphviz by *dot -Tpdf ./path/out.dot > ./path/out.pdf*

Command line options

- *- -help* : Can be used to get information about the usage of the parser, i.e. ”myASTGenerator”.

- *- -input* : Can be used to set the input to the file that needs to be parsed. The path is expected to be space separated from *- -input.* Path can not be empty, it needs to be set.

- *- -output* : Can be used to set the output, where the dot file will be sent. Any errors that may exist will be shown on the terminal via stderr. The path is expected to be space separated from *- -output.* Path can not be empty, it needs to be set.

- *- -verbose* : Verbose output is sent to *parser.out.* It contains information about all the states, lists if grammar has any conflicts and also contains the actions and goto definitions for all states. The submitted parser.y is conflict-free.

Instructions for Running test cases

- We have provided 25 non-trival programs that may be compiled using the parser. These include examples of interfaces, multi dimensional arrays, basic data types, class inheritance, method and function overloading etc.

- We have provided an easy python script to automate the testing of these testcases.

- Please change directory into *./milestone1/*

- Execute *python run.py*

- The tests are named *./tests/test_[1-25].java*, corresponding outputs include *./outputs/out[1-25].dot* and *./outputs/out[1-25].pdf.* Pdf file stores the AST.