

Milestone 1 - CS335A

Course : CS335A - Compiler Design
Indian Institute of Technology, Kanpur

Rajat Gattani : 210813, Mohak Singh Rana : 210614, Dhruv Garg : 210339

March 3, 2024

Tools and Utilities used

- The lexer is written in Flex and it returns the tokens to the parser
- Bison is used for implementing the parser. Official grammar is rewritten by removing ambiguous and irrelevant rules such that there are no conflicts in LALR(1) parsing
- Make utility is used for automatic tracking for files and compilation

Features Implemented

- We have implemented all the basic features mentioned in the description
- Leaf nodes are square red colored blocks while non leaf nodes are black oval blocks in the AST
- Implicit and explicit line joining as in python
- Proper indentation and dedent pattern
- Support for classes and methods, including multilevel inheritance and constructors.
- Static polymorphism via method overloading
- Error handling which raises error with line number and offending character/token on invalid character, invalid syntax and improper indentation according to python rules

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. "myAST-Generator".
- `-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 testcases

- We have provided 5 non-trivial programs that may be compiled using the parser. These include examples of basic data types, recursion, class inheritance, method and function overloading etc.
- We have provided a bash script to automate the testing of these testcases.
- Please change directory into `./milestone1/`
- Execute `./checking_script.sh`
- The tests are named `./tests/test [1-5].py` , corresponding outputs include `./outputs/test[1-5].dot` and `./outputs/test[1-5].pdf` . Pdf file stores the AST.