CS335A: Compiler Design (Assignment 2: PARSER)

- Source Language: *Python*
- Target Language: *MIPS Assembly*
- Implementation Language: *Python*
- Tool Used : PLY (Python Lex and Yacc)

Running Instruction

1. Run the makefile `make`
2. To run the parser, pass the path of filename as argument. `bin/parser test/test[\d+].py` The parser will call the converter and then call dot to finally output the png file of parse tree. The output will be saved in base directory.
3. To clean the executables and other helper files, run make clean. `make clean`
4. To generate tree on a png file manually, use *dot* command. `dot -Tpng dotfilename.dot -o pngfilename.png` for png format output or, `dot -Tps dotfilename.dot -o psfilename.ps` for ps format output.

Directory Structure

- bin:
  - converter.py [Python source file to convert the dump of parser into dot file]
  - lex.py [Python source file from PLY for lexing]
  - lexer.py [Python source file to specify language lexemes]
  - parser [Python dependent bytecode for parsing]
  - parser.py [Python source file to specify grammar]
  - yacc.py [Python source file from PLY for parsing]
- src:
  - converter.py [Python source file to convert the dump of parser into dot file]
  - lex.py [Python source file from PLY for lexing]
  - lexer.py [Python source file to specify language lexemes]
  - parser.py [Python source file to specify grammar]
  - yacc.py [Python source file from PLY for parsing]
- test:
  - test[\d+].py [Test files]
- .gitignore
- makefile [To move the source files to bin directory and compile bytecode for lexer and making it executable]
- readme.md