David Silva
862042571

CS-152 Phase 2 Write Up

Motivation

Bison is a parser generator. The parser generated is a bottom-up parser that parses tokens generated from phase 1 (lex). Creating a working bison file that generates a working parser is a necessary step in creating a compiler for the MINI-L language.

Describing your grammar

The grammar follows the railroad diagrams as closely as possible. The only grammar deviation is how many additional productions the terminal symbol UMINUS appears in for the “term” non-terminal rule, as instructed by Professor Payne. I also made a few additional non-terminals so that the grammar was a bit easier to generate/write.

Workload Division (if in a group)

N/A; Done solo.

Specific problem you encounter and how you resolved

The biggest problems I encountered were trying to compile the c files generated by bison/lex using g++. I gave up on that and got the c files to compile using gcc quite quickly/smoothly. Then, Professor Payne showed us how to compile the files using g++

in his Piazza post. I was able to quickly get my files to work with g++ after following the instructions with the Piazza post.

I ran into a few shift-reduce conflicts while debugging my grammar, but they were quickly resolved upon closer inspection of the syntax diagrams and my grammar. Interpreting the syntax diagrams into context free grammars is straightforward.