Assignment Three

Jeremy Poff

10/20/2017

Cexpr Report

The problem I set out to solve was that of parsing simple C math expressions. We were told to use lex and yacc to solve this problem. We were also given an table showing the precedence and associativity of the operations we needed to implement. We were not allowed to use the built in yacc tools for this and instead had to implement it as a recursive grammar with additional non-terminals.

The lex part was simple and the only additional thing I really added was tokens to recognize assignment operators that weren't just an equal sign and shift operators. Initially I forgot to ignore whitespace, but this was quickly fixed.

The biggest issue I had was that I implemented my grammar initially as right instead of left recursive. Once I solved this problem, the rest was relatively straightforward to implement. I also had some trouble looping back to allow for nested expressions, but this was also fixed. The other problem I dealt without throughout the project was deciding where to put print statements so that nested expressions did not print multiple times and assignment expressions printed the new value that was assigned to the variable.

I tested my solution using a long series of commands piped together and diffed the output from my program with the expected output.

Overall the assignment was not too difficult once I figured out that I was implementing the grammar backwards. Discovering this error did take quite some time though.