CPSC 4660 PL Scanner

Steven Deutekom
Ricky Bueckert
University of Lethbridge

March 11, 2020

1 Contributions

The entire class contributed equally on this project. We discussed the design and then split up the classes. Once we started the implementation there was a lot of communication to decide on how each peice of the Scanner should work. There was also a lot of collaboration deciding on what to put in our test files and debugging when the tests exposed flaws.

2 Files Included

2.1 Header Files

- Symbol.h
- SymbolTable.h
- Token.h
- Scanner.h
- Administration.h
- Parser.h
- Grammar.h
- Types.h
- BlockTable.h

2.2 Source Files

- plc.cc
- SymbolTable.cc
- Token.cc
- Scanner.cc
- Administration.cc
- Parser.cc
- BlockTable.cc

2.3 Test Files

- search.pl - A full test program from the Brinch Hansen textbook.
- defs.pl - Test all of the defenition rules.
- simpleExps.pl - Tests all of the expression rules.
- errorTest.pl - Tests a number of different errors. Tries to ensure that the parser catches all errors and recovers to a stable state.
- fibDP.pl - A program to calculate fibonacci numbers from 0-10. Tests recursion and procedure parameters.
- typeErrors.pl - Check a few of the possible type errors.
- testNew.pl - Sets up some records and procedures with parameters.

3 Compiling and Running

1. Navigate into the project folder and type the following:

   ```
   \$ make compiler
   ```

2. This will create a binary called compiler that cna be run with the following command:

   ```
   \$ compiler <testfile-path> -o <output-file>
   ```

3. If you omit the `-o <output-file>` flag the tokens will be output to a file called `pl.out`
4. If you add the `-v` flag some debugging info will be printed. This will include each function that is called and matched tokens. This will not affect the output of tokens.

4 Bugs

None. That we know about.

5 Time Spent

Hours spent on this project combine all group member contributions. Scanner 30 hrs. Parser 40 hours. Rule additions and Scope and Type checking 30 hours.