Aidan Wolter

Programming Assignment #2

Program Review Report

Dr. Baas

COSC4503

Program Translation

Feb 26, 2015

**1. Usage**

To use this program ensure that the following files are in the same directory:

* lex\_table.csv – defines what constitutes a token for the language
* parse\_table.csv – defines proper syntax for the language

Typing *./syn input\_file* into a command prompt will then run the program with the file named *input\_file*.

**2. Code Structure**

I have separated my code into four modules: main controller, lexical analyzer, syntax analyzer, and symbol table. The main controller requires three files: two to define the language and one for input. First, the controller initialized the symbol table and constructs both the lex\_table and the parse\_table. It then runs a function to parse the input file. This function calls the lexical analyzer in order to get a stream of tokens then checks the order of the tokens with the defined parse\_table. If an error is found in the lexical analyzer, the lexical analyzer will report an error and jump to the next line. If an error is found in the syntax analyzer, the syntax analyzer will report an error and jump to the next line. In this way, multiple errors can be found and reported in a single input file. As the syntax analyzer receives ID tokens, it adds these tokens to the symbol table for future use.

**3. Symbol Table Structure**

I am using a simple C struct to hold the ID symbols. The struct contains an array of symbols and an integer value holding the number of symbols added. Each symbol is defined in a separate struct that holds the symbol name, token, and type.

**4. Syntax Error Reporting**

As the program is being parsed, both unrecognized tokens and unexpected tokens are reported. In the case of an unrecognized token, the output will indicate which line the error was found on, the token, and it will point to where the token is located in the line. Similarly, if an unexpected token is found, the output will indicate the line number, the token, where the token is located, and also what are valid token options for that location.

**5. Conclusion**

I spent a lot of time on this project mostly because I enjoyed working on it and wanted it to not only look good but also be structured well. Writing the code to jump through the parse\_table was the most enjoyable part. The most difficult parts were working on the pretty error output and parsing the lex table and parse table CSVs.

Time spent: 20-25hrs