Project 1: Lexer

Austin Johns

Grand Canyon University

CST-405: Principles of Compiler Design

**Introduction**

In this project, a we focus on lexical analysis. We started this process in CST-301, and we are building on it now with the goal of ending up with a full code analyzer and compiler. The first step to this process is creating a Lexical Analyzer. A Lexical Analyzer, or Lexer, follows an input to gather information on what types of input the characters are. The input can result in classifications such as: Identifier, operator, constant, end symbol, etc. From here, a syntax analyer can help further identify the meaning of code input.

**Defined Regular Expressions**

Some regular expressions that will be defined by the lexer include:

* Integer
* Character
* Return
* Keyword
* Operator
* Unary operator
* Identifier

**Main Purpose**

The main purpose of the lexical analyzer is to be the first interpretation for code analysis. The Lexical analyzer produces an output of tokens for the parser to use for syntax analysis. From here, the parser will send the syntax analysis to be interpreted into actions for the compiler to take.

**Detailed Description**

In this project, for the Lexer to function properly, first the input file is read in. The program reads all characters of the input file and assigns them to a token. From here, a table is produced to display the types of tokens and the string on the same line to be read by the user. In the future, these tokens will be passed on to be interpreted by the parser.

**Code Skeleton**

The code skeleton for this project is long and detailed, but fairly simple. The code skeleton follows the structure:

1. Defining tables for storage of strings
2. Loop to sort through each table to find every type of character input for tokens.
3. Creating a list of multiple lists to be sorted through a large for loop.
4. Iterating through the rows of tables
5. Iterating through the main list
   1. Defining tokens
6. Printing the strings, tokens, and line numbers into a legible table for the user.

**GCU-PL Input File**

The GCU-PL input file includes a few terms for the lexical analyzer to read through. Below is a screenshot of the GCU-PL file.

A screenshot of a cell phone

Description automatically generated

**Screenshots**

Below are some screenshots of code and execution of the lexical analyzer.

A screenshot of a computer

Description automatically generated

**A screenshot of a cell phone

Description automatically generated**