

CPSC 323 Project 2 Documentation

1. Problem Statement

For assignment 2, the syntax analyzer requires us to display the output of tokens, lexemes, and production rules that was used. The goal of the assignment is to utilize the lexical analyzer from our previous assignment to create a syntax analyzer for our lexemes and tokens.

2. How to use your program

The program can be run in Windows 10 by using the terminal, navigate to the directory containing all the files. Then type in “py syntax2.py” or “python syntax2.py” to compile the code. The output will then be displayed in the terminal.

3. Design of your program

The design of the program starts of by importing the code from our previous assignment lexical analyzer. The lexical analyzer from the previous assignment will help us identify the tokens and lexemes we need to analyze. The input text file will first be analyzed by our lexical analyzer, then our syntax analyzer will continue the process.

The lexemes and tokens will be passed into our helper function statements() containing two other functions, assignment() and declarative(). The assignment function is used to identify each token and lexeme in the code. It will check the token and lexeme one by one to see if it follows the syntax

grammar rule that was created. First it will check the first token to see if it is an Identifier and increment the identifier counter. Next it will check for the Assignment operator (=) after the Identifier. Once these conditions are met, it will break down the expression following the Assignment operator and using the Factorization rules created. The program will continue to analyze the rest of the code with the same analyzing process.

4. Any Limitation

A minor limitation for the code is that the lexemes of the text file must be formatted in a specific way. In this case, the lexemes must be spaced out with at least one blank space. For example, the semicolon must be separated from the lexeme.

5. Any Shortcomings

The program is currently unable to identify the If, while tokens.

Factorization Rules.

$A = A + C ;$

$A = A - D ;$

$B = G * N ;$

$\langle \text{Expression} \rangle \rightarrow \langle \text{Expression} \rangle + \langle \text{Term} \rangle$

$\langle \text{Expression} \rangle \rightarrow \langle \text{Expression} \rangle - \langle \text{Term} \rangle$

$\langle \text{Expression} \rangle \rightarrow \langle \text{Expression} \rangle \langle \text{Expression Prime} \rangle$

$\langle \text{Expression Prime} \rangle \rightarrow + \langle \text{Term} \rangle \mid - \langle \text{Term} \rangle$

$\langle \text{Expression} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Term} \rangle$