

Lexical Analyzer Report

Name: Ameer Ayman Ahmed

ID: 135058

Lexical analyzer has main purpose which is generate tokens to use these tokens as an input in the parser. There are 6 types of the tokens that we split the input or the source code according to them, data type, reserved words, symbols, operators, literals and identifiers.

**The method that use for tokenization process:**

The program is divided into some functions to tokenize the source code. Firstly, the program read the file that contains the source code line by line, remove the single comments which is between these symbols ‘{}’ and the spaces between the words if it is more than one space then it saves the line into array list.

The second function remove multiple comments which is between these symbols ‘{\*\*}’.

The third function is putting spaces between words and symbols or special characters, this function is working by checking the special characters from an array to put space before and after this symbol then updating the array list of the strings.

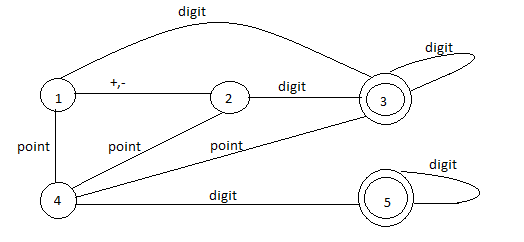
After finishing the previous three functions the array list is now ready to split into tokens. The fourth function is tokenization which split the words and symbols by whitespace and store each token in new array list and the line number for each token in another array list.

The first part of lexical is finished and we need to specify type for each token from the 6 types that mentioned above. The 6 functions are:

1. isDataType ()
2. isOperator ()
3. isSymbol ()
4. isReservedWords ()
5. isLiterals ()
6. isIdentifier ()

all of them return Boolean and takes argument which is string. isLiteral and isIdentifier are implemented using DFA machines.

**Literals DFA:**



**Identifiers DFA:**

