Samprayash Dahal

Cpsc 323-04

Project 1: Lexer

Professor Anthony Le

CPSC 323-04 Documentation

1. Problem Statement:
   This assignment consists of a lexer function that returns a token when it is needed. The lexer function returns a record, one field for the token and another field for the actual value of the token. This program reads a file containing the source code given from class to generate tokens and write out the results to an output file.
2. How to use program
   This program can be run on any IDE for c++ and consists of no errors.
3. Design of the program
   In this program I have used functions like:
   bool check_if_keyword(char* string); to check if the lexemes obtained from the input file are keyword or not. In this function I compare the strings and if they are of type keyword then the function will return true if not it will return false.

   bool check_if_delimiter(char c); to check if the lexemes obtained from the input file are delimiter or not. In this function I compare the strings and if they are of type delimiter then the function will return true if not it will return false.

   bool check_if_identifier(char* string); to check if the lexemes obtained from the input file are identifier or not. In this function I check if the strings are of any numbers or delimiter. If they are of type number or delimiter then the function will return that the it is not an identifier, it will return true if its an identifier.

   bool check_if_int(char* string); to check if the lexemes obtained from the input file are an integer or not. In this function I compare the strings and if they are of type other than an integer then the function will return false if not it will return true stating that it is an integer.

   bool check_if_real_num(char* string); to check if the lexemes obtained from the input file are of real numbers or not. In this function I compare the strings and if they are of type other than realnumbers then the function will return false if not it will return true stating that it is a real numbers.

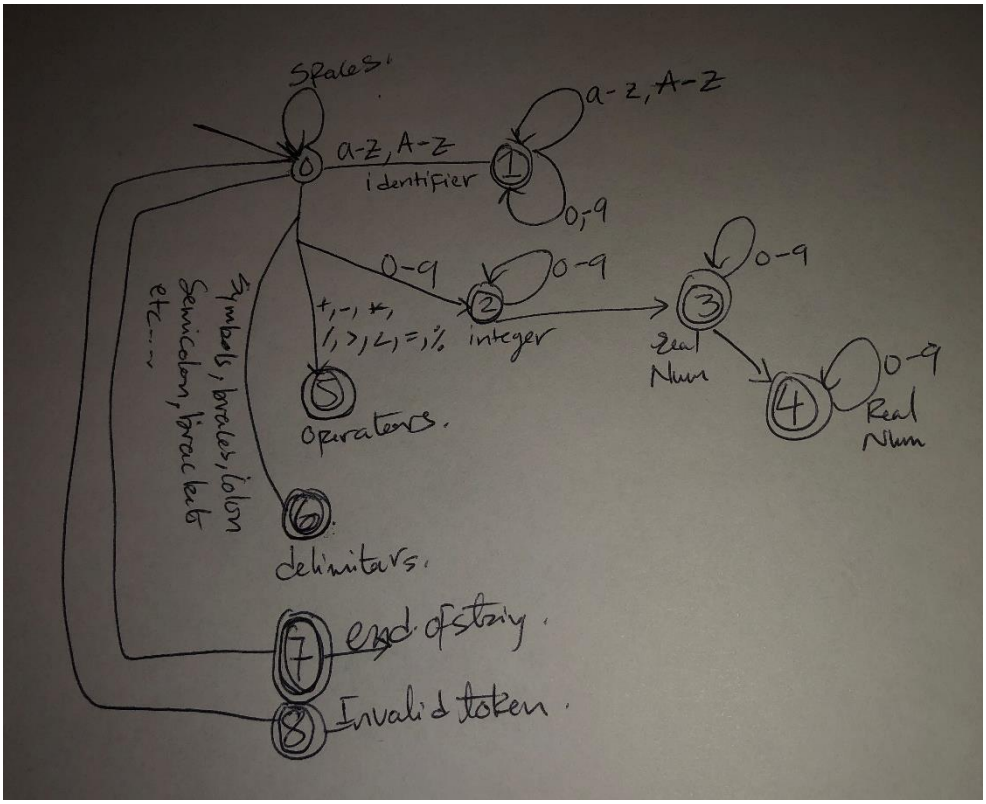
   bool check_if_operator(char c); to check if the lexemes obtained from the input file are an operators or not. In this function I compare the strings and check to se if the are

from one of these category “+, -, *, /, <, >, =, %” and if it passes then it will return true.

char* subString(char* string, int left, int right); in this function it extract the string and return the substring. Returns a newly constructed string object with its value initialized to a copy of a substring of this object. The substring is the portion of the object that starts at character position *pos* and spans len characters (or until the end of the string, whichever comes first).

void parse(char* string); this function compares the string obtained from the input file and checks to see which token it belongs to and writes results to an output file

void readFile(string filename); this function reads a file from and stores in a string, later that string is stored in a char array and the function parse is called to parse through the string.



RE for the program

Lexer: consists of keywords, identifiers, real number, integers, operator and separator. The extra spaces is ignored when the result is written to a output file.

Identifier: a sequence of letters or numbers. First number must be a letter and last char must be either $ or letter. It accepts both the upper and lower case.

Integer: numbers without decimal digits.

Real Numbers: its an integer followed by decimal point

4. Any Limitation
   None
5. Any shortcomings
   None