CS301P Compiler Design Laboratory Exercises Week-2

Date: Aug 12 2024

Objectives

- To develop *Lexical Analyzer* for a few languages

Exercise Problems

1. Construct a lexical analyzer to accept the valid strings of the following language. $\{w \in \{x.x^R \mid x \in \{0,1\}^*\}$ and the number of 0's in w is divisible by $4\}$

   Sample Input:

   001100 → Accepted

   001010 → Rejected

   110011 → Rejected

2. Construct a lexical analyzer to identify valid C assignment statements while defining the variables. Assume only *int* and *float* types.

   Sample Input:

   int a = 3; → Accepted

   float b == 7 → Rejected

   int c = 9, d = 0; → Accepted

3. Build a lexical analyzer that takes a C-program as an input (through CLI), removes the comments, extra white spaces/tabs/newlines etc., tokenizes all valid C program constructs, such as integer constants, floating point constants, character/string constants, identifiers, keywords, operators, and stores the necessary information about the identifiers into symbol table datastructure.

   You may display each token and the corresponding lexeme on the screen. Before exit, store the symbol table entries in a separate file named SymbolTable.txt.

References

1. Flex Manual https://westes.github.io/flex/manual/
2. GNU Make Manual https://www.gnu.org/software/make/manual/html_node/index.html
3. C Language Reference Manual https://www.bell-labs.com/usr/dmr/www/cman.pdf

Note: Submission guidelines, Evaluation guidelines, and Declaration of academic honesty are same as Lab#1