SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Compiler Design Lab – CS6612

Programming Assignment-2 - Implementation of Lexical Analyzer for the patterns (identifier, comments, operators, constants)

Date of Assignment: 05.02.16 **Due Date: 09.02.16 & 11.02.16**

Develop a Lexical analyzer to recognize the patterns namely, identifiers, constants, comments and operators using the following regular expressions.

Regular Expression for Identifier	**Regular Expression for Constant**
letter → [a-zA-Z] digit → [0-9] id→letter(letter|digit|ϵ)*	digit → [0-9] digits →digit digits optFrac →.digits optExp → E(+|-|ϵ) digits numberconst →digits optFrac optExp charconst → '|letter|' stringconst → "|(letter)*|" constant → numberconst | charconst | stringconst
Regular Expression for Comments	**Regular Expression for Operators**
start1→ * end1 → */ multi → start (letter)* end start2 → // single → start (letter)*	relop → < | <= | == | != | > | >= arithop → + | - | * | / | % logicalop → && | || | ! operator → relop | arithop | logicalop

Convert the regular expressions into cumulative transition diagram as shown in Figure 1. Each state represents a condition that could occur during the process of scanning the input looking for a lexeme that matches one of the several patterns. Convert each state into a piece of code.

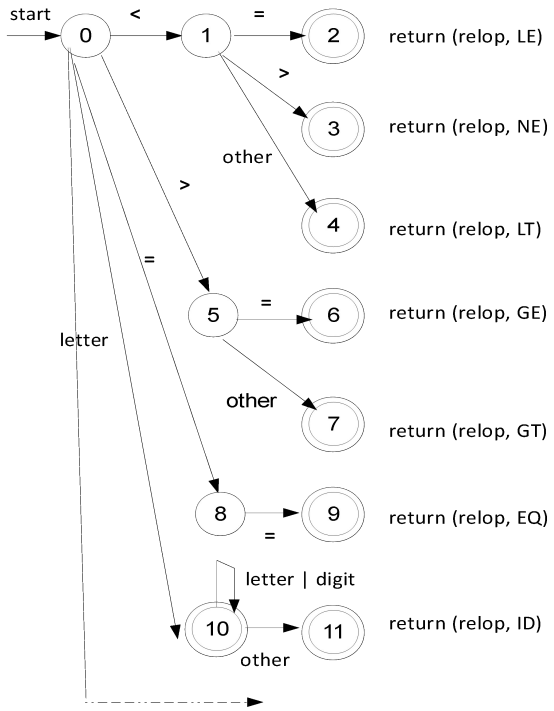

Figure 1. Cumulative Transition diagram