

# Programming Assignment 1 – CS 301 - Theory of Automata – Spring 2019

**Total Marks: 100**

**Due: Saturday, February 9<sup>th</sup> 2019 (11:00 PM)**

**Note: Late submissions are not allowed for this assignment.**

## The Language L

In this assignment, you will have to make a deterministic finite automata for a restricted version of C++ programming language in which the variable names have the following restrictions:

1. A variable name can only use English letters (a, b, c, ..., z, A, B, C, ..., Z) or numbers (0, 1, 2, ..., 9). Special characters, such as underscore, are not allowed.
2. A variable name must start with a letter.
3. The length of the variable name must not be more than 10 characters.
4. A variable name cannot be a keyword (e.g. if, else, while etc.). A finite list of keywords is provided below.

## DFA for L

You will have to construct a DFA  $M = (Q, \Sigma, \delta, q_1, F)$  that recognizes L. The DFA must satisfy each of the following properties:

- Assume  $\Sigma$  to be all the characters on the keyboard. You will have to include trap states (if needed) in your DFA. You can divide your  $\Sigma$  into three sets. E.g.  $\Sigma_1$  can be set of English letters,  $\Sigma_2$  can be set of numbers and  $\Sigma_3$  can be rest of the characters on the keyboard.
- The states in the DFA must be labeled  $q_1, q_2, q_3, \dots, q_n$  where  $q_1$  is the start state and  $n$  is the number of states in the DFA.

Please note that if the DFA is overly complicated, you may lose points. Include “trap state” if needed and clearly identify that state. To simplify your drawing, it is better to use  $\Sigma_1, \Sigma_2$  etc.

## Programming Specifications

You will have to do this assignment in C++ only.

1. Your program should first print out your name and FAST ID ( e.g. kamran.lodhi ), and explain to the user what the program does.
2. Your program next asks the user if they want to enter a string. The user enters “y” for “yes” and “n” for “no”.
  - a. If “n” is entered, the program terminates
  - b. Else, the user is prompted to enter a string
3. After the string is inputted, the program prints and processes the entire string on your DFA, **one character at a time**, in the following manner.
  - a. Your program must begin the start state of the DFA and print out the name of the state ( $q_1$  or  $q_0$ )
  - b. After each character from the string is processed, your program prints out the character and the current state of the DFA. Your program should also mention TRAP state if your DFA is in TRAP state and must process all characters until the string reaches its end.
4. After processing the entire string on the DFA, your program must indicate if the string is a valid variable name, a keyword or an invalid variable name based on the state in which the DFA ended. Your program then returns to step 2.

### **What to Hand In:**

- Your code, **including a user documentation file that includes your clean drawing of the DFA.**
- All files should be zipped together into a single file, whose name is comprised of your FAST Roll#, followed by the letters "progassignment1", followed by the assignment number. ( E.g. L1XXXXXprogassignment.zip ).
- If there is anything special I should know about your program, be sure to document it in the documentation file. Anything special I need to do to run your program and understand the results should also be documented.
- If there are problems that you know your program cannot handle, it is best to document them as well.

### **List of Keywords**

int, char, bool, double, if, else, while, for, and, or, static.