

Programming Assignment 1

Be sure to read this entire document before starting the assignment. Also, you should complete problem 6 of Homework 3 before starting.

1 Overview

Define the language L to consist of strings that represent numbers in a modified version of Java defined below. For this assignment you are to write a program that implements a DFA that recognizes the language L . You must also turn in a drawing of your DFA for L .

2 The Language L

To precisely specify L , first define the set of *digits* to be $\Sigma_1 = \{0, 1, 2, \dots, 9\}$, and define the set of *signs* to be $\Sigma_2 = \{+, -\}$. Also, define the following sets:

- L_1 is the set of all strings that are decimal integer numbers. Specifically, L_1 consists of strings that start with an optional sign, followed by one or more digits. Examples of strings in L_1 are “02”, “+9”, and “-24167”.
- L_2 is the set of all strings that are floating-point numbers that are not in exponential notation. Specifically, L_2 consists of strings that start with an optional sign, followed by zero or more digits, followed by a decimal point, and end with zero or more digits, where there must be at least one digit in the string. Examples of strings in L_2 are “13.231”, “-28.” and “.124”. All strings in L_2 have exactly one decimal point.
- L_3 is the set of all strings that are floating-point numbers in exponential notation. Specifically, L_3 consists of strings that start with a string from L_1 or L_2 , followed by “E” or “e”, and end with a string from L_1 . Examples of strings in L_3 are “-8086.1E-043”, “+2.E5” and “1e+31”.

Then we define L as $L = L_1 \cup L_2 \cup L_3$. There is no limit on the number of digits in a string in L . Also, we do not allow for the suffixes L, l, F, f, D, d, at the end of numbers to denote types (long integers, floats, and doubles).

3 DFA for L

First construct a DFA $M = (Q, \Sigma, \delta, q_1, F)$ that recognizes L , where Σ is the alphabet of all printable characters on a standard American computer keyboard (i.e., decimal ASCII codes 33 to 126). The DFA must satisfy each of the following properties:

- The DFA must be defined with the above alphabet Σ . In other words the DFA must be able to handle any string of printable characters.
- The DFA must have exactly three accept states: one for L_1 , one for L_2 , and one for L_3 . In other words, all strings in L_1 must end in one accept state, all strings in L_2 must end in another accept state, and all strings in L_3 must end in a third accept state. Any string over Σ not in L must end in one of the other states.
- 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. (It is also acceptable for the states to be labeled q_0, q_1, \dots, q_{n-1} , with q_0 the start state.)

4 Program Specifications

You must write your program in either C, C++, or Java. All input/output must be through standard input/output, and your program is to work as follows:

1. Your program asks the user if s/he wants to enter a string. The user then enters “y” for “yes”, or “n” for “no”.
 - If the user enters “n”, then the program terminates.
 - If the user enters “y”, then the user is prompted to enter a string over Σ .
2. If the user chooses to input a string, your program then reads in the string. After reading in the string, your program prints it. Then your program processes the string on the DFA, one character at a time, in the following manner.
 - Your program must begin in the start state of the DFA and print out the name of that state (q_1 or q_0).
 - After each character from the string is processed on the DFA, your program must print out the character and the name of the current state of the DFA.

To simplify your program, you should check the ASCII code of each character of the string and process on the DFA accordingly.

3. After processing the entire string on the DFA, your program must indicate if the string is accepted or rejected. If the string is accepted, then your program indicates if the string is in L_1 , L_2 or L_3 , based on the accept state in which the DFA ended. Your program then should return to step 1.

5 Test Cases

Test your program on each of the following input strings:

-193.	+000	-0.E+	4	+57.317
.Ab123C	-.7e0	CS341	-5.E-0	3.6#
+	-903.15e+12	.6	5e2.8	7E2

6 Deliverables

You must complete the following by the **beginning of class** on the due date given in the syllabus (for eLearning students, your envelope containing the hardcopies must be postmarked by the due date):

1. turn in a drawing of the DFA for L that your program implements,
2. turn in a hardcopy listing (printout) of your program,
3. turn in a hardcopy listing (printout) of the output from running your program on each of the test cases,
4. e-mail a copy of the source code to marvin@njit.edu with “CS 341 program 1” as the subject line.

You will not receive any credit if you do not complete all of the above. **Late assignments will not be accepted.** Your grade will mainly be determined by the hardcopies that you turn in; the softcopy of the source code will only be tested if there are questions about your code.

7 Grading Criteria

The criteria for grading are as follows:

- correctness of your DFA for L ,
- your program matches your DFA for L ,
- the program works according to the specifications given in Section 4,
- your program is properly documented (i.e., comments),
- your output is correct for the test cases.

To receive any credit for this assignment, you must turn in a drawing of your DFA for L and a *minimally working* program. For a program to be minimally working, it must

- compile without syntax errors;

- properly accept strings in L_1 ; and
- implement the drawing of your DFA for L .

If you do not hand in a minimally working program by the due date/time, then you will receive a 0 for the assignment *and* your grade in the course will be lowered by one step, e.g., from B to C+, or from C to D.

You should do this assignment in steps. Start by developing a DFA to recognize L_1 . Once you have this figured out, then enhance your DFA to also accept the strings in L_2 . Once you have this figured out, then enhance your DFA to also accept the strings in L_3 .

8 Honor Code

Any student caught cheating on this assignment will be reported to the dean of students. Cheating includes, but is not limited to, copying code from someone else, having someone else write your program, turning in code that you did not write, or giving your program to another student.