Assignment 4

Advanced Computing concepts for Engineering

Due 2018 Feb 26

Note that the work that you turn in for this assignment must represent your individual effort. You are welcome to help your fellow students to understand the material of the course and the meaning of the assignment questions, however, the answer that you submit must be created by you alone.

Q0. Make a regular expression that represents valid numbers according to the following rules

- A number consists one or more of characters in the following set

  {'.','-','E','+','0','1','2','3','4','5'.'6','7','8','9'}

- A number consists of 3 parts. Each is optional.
- The first part consists of 1 or more digits.
- The second part consists of a dot followed by 0 or more digits.
- However, if the first part is omitted, the second part must not be omitted and must have at least one digit.
- The third part consists of the letter E, followed by either a - or a + or nothing, followed by one or more digits.

(a) Write the regular expression without using any of the abbreviations given in the notes.

(b) Write the regular expression as compactly as you can using the abbreviations in the notes.

Q1. Comments.

A C comment is a string that follows the following rules. C comments have at least 4 characters. The first two are "/*" and the last two are "*/". The sequence "*/" does not occur in between the first two and last two characters. Examples include "/**/", "/*/*/", and "/*o*o/*/" . Counterexamples include "o/**/", "/**/o", and "/**/*/". Suppose x is a regular expression whose language is the set of all strings of length 1 other than "/" and "*". Devise a regular expression whose language is the set of all C comments.

Q2. Reversal

Show that if a language M is represented by a regular expression x, then there is a regular expression that represents the reversal of M, i.e., the language N that consists of all the strings in M written backwards.

Q3. Addition

Represent an addition such as $2 + 3 = 5$ by “235” and $12 + 35 = 47$ by “134257”. Pad the operands with 0s so that both operands and the result are the same number of digits; e.g., represent $23 + 777 = 800$ by “078270370”. Thus strings in the language are always of lengths that are multiples of 3.Create a regular expression that represents correct additions in binary. Hint: it might be easier to first create a regular expression for the reversed language. Examples

- $11 + 101 = 1000$ is represented by a string 001 010 100 110 which is in the language.
- $11{+}11 = 110$ is represented by a string 001 111 110 which is in the language
- $1 + 1 \neq 1$ and so 111 is not in the language.
- 00000 is not in the language, as its length is not a multiple of 3.