CSI4104 Compiler Design
2nd Semester, 2019

Homework 1

DUE DATE for Exercise 7: Friday, November 1, 2019, 23:00 (to be handed in on the provided YSCEC report section, as a single pdf file).

Exercises 1–6 need not be handed in because you'll be given the solutions for your preparations for the midterm exam. Solutions will be posted on YSCEC two days ahead of the exam.

1. Given the alphabet $\Sigma = \{-, +, x, y, z, 0, 1\}$ and the below regular expression, which strings are valid? (Mark one or more.)

$$R : (-|+|\epsilon)\ (x|y|(0|1)^*)z(0|1)^*$$

   (a) $-xz1$
   (b) $-xz$
   (c) yxz
   (d) ϵ
   (e) $01z-1$
   (f) $+011z1$
   (g) $+x$
   (h) zx

2. Given the regular expression below, which strings are valid? (Mark one or more.) Note: "$[a-z]$" denotes lower-case characters, "$[A-Z]$" denotes upper-case characters, and "$[0-9]$" denotes the digits from 0–9.

$$R : [a-z]([a-z]|[A-Z]|[0-9]|\_)^*$$

   (a) $x\_$
   (b) $\_x$
   (c) $yx0z$
   (d) ϵ
   (e) $ada1$
   (f) $a\_d\_a\_1$

3. Write regular expressions for the following languages whose alphabet is $\Sigma = \{0, 1\}$.
   (a) All possible strings, including the empty string.
   (b) The empty string.
   (c) The string 1011
   (d) The strings 1 and 101.
   (e) All strings beginning with 01.
   (f) All strings that contain exactly two 1's.
   (g) All strings beginning with a 0 and ending with a 1.

4. Design deterministic finite automata (DFAs) to recognize the following languages over the alphabet $\Sigma = \{x, y\}$.
   (a) Every occurrence of the substring yy is followed by an x.
   (b) Every third symbol is an x.
   (c) All strings with an even number of x and an even number of y.

   When you build an automaton by hand, it is a good idea to add a description to each state: the description should specify which strings can possibly reach that state.

5. Use Thompson's construction algorithm to construct non-deterministic finite automata (NFAs) from the following regular expressions:
   (a) $(a|b)^*$
   (b) $a^*(b|c)$
   (c) $a(a|b)^*a$

6. Use the subset construction algorithm to convert the above NFA for $a^*(b|c)$ to a DFA.

7. Real-world regular expressions (not for midterm):

   (a) Use grep to find out the number of concerts in *a minor* by Antonio Vivaldi that are mentioned in file `/opt/ccugrad/Homework1/music.txt` on the elc1 server. Note: you can find out about grep in the Linux man pages (`man grep`). You have to look for strings containing "Vivaldi" and "a minor".

   (b) How would you search with the vi editor for all music pieces written in the following minor keys: c,d,e,f,g,a,h? You have to specify a regular expression that describes the following strings: one of the minor keys, followed by a blank, followed by the word `minor`.

   (c) Write a Python program named "replace.py" to replace all occurrences of a given string in a file with another string. The two strings (source and target string) as well as the file-name should be read from the command-line.

   For example, the following invocation would replace all occurrences of the string "foo" by the string "bar" in a file named "testfile":

   ```
   python replace.py foo bar testfile
   ```

   Your program should replace a string even if it occurs as a substring. In the above example, if file "testfile" contains the line

   "int foo, foo1, foofoobar;",

   it should be re-written to

   "int bar, bar1, barbarbar;".