Assignment 3

Ethan Poole
LING 185A: Comp. Ling. I
Due: 21 April 2019

Instructions: Download `RegEx.hs` and `Assignment03.hs` from the course website into the same directory on your computer. The **`import`** line near the top of `Assignment03.hs` imports all of the definitions from `RegEx.hs`, which you may then use in your assignment. Please submit your assignment as two files: (i) a modified version of `Assignment03.hs` for part one, making sure not to change the file name, and (ii) a PDF for parts two and three.

1

10 points

Please code up the following functions that extend the utility of our regular-expression implementation in Haskell.

(1) a. `occurrences :: Int -> RegEx -> RegEx` such that `occurrences n r` returns a `RegEx` that would match exactly *n* occurrences of *r*.

b. `optional :: RegEx -> RegEx` such that `optional r` returns a `RegEx` that would match either zero occurrences or one occurrence of *r*.

$\boxed{2}$ *10 points*

Please write a regular expression and draw a finite-state automaton with the fewest states possible that recognize the language L in (2). Assume that a space represents concatenation.

(2) a. $\Sigma = \{\text{the}, \text{very}, \text{hungry}, \text{caterpillar}\}$

b. $$L = \left\{\begin{array}{l} \text{the caterpillar,} \\ \text{the hungry caterpillar,} \\ \text{the very hungry caterpillar,} \\ \text{the very very hungry caterpillar,} \\ \text{the very very very hungry caterpillar,} \\ \text{the very very very very hungry caterpillar,} \\ \ldots \end{array}\right\}$$

$\boxed{3}$ *15 points*

Please draw a finite-state automaton with the fewest states possible that recognizes the set of strings described in (3). Please also give the set of transitions as a table.

(3) a. Your password may be any length.

b. Your password may contain alphabetic characters (a–z, A–Z), integers (0–9), punctuation (. ? !), and nothing else.

c. Your password must contain at least one alphabetic character, one integer, *and* one punctuation.

In your finite-state automaton, you may (and should) use A, I, and P as shorthand to represent any alphabetic character, integer, and punctuation respectively.