York University | EECS 2001 | June 8, 2015

Homework Assignment #4
Due: June 17, 2015 at 7:00 p.m.

1. If $L \subseteq \Sigma^*$ is any language, we define $DROP(L)$ to be the set of all strings that can be obtained by removing a single character from a string in L. More formally,

$$DROP(L) = \{xy : x, y \in \Sigma^* \text{ and } \exists a \in \Sigma \text{ such that } xay \in L\}.$$

(a) Let $L_a = \{\mathtt{010}, \varepsilon, \mathtt{011}, \mathtt{0}, \mathtt{0010}\}$. List all the elements of $DROP(L_a)$.

(b) Let L_b be the language described by the regular expression $\mathtt{a(bc)}^*$. Write a regular expression for the language $DROP(L_b)$. You do not have to prove your answer is correct.

(c) Your goal is to define a function D that maps regular expressions to regular expressions. Give a recursive definition of $D(R)$ so that if R is a regular expression for a language L, then $D(R)$ is a regular expression for the language $DROP(L)$.

You can write your answer by filling in the blanks in the following. (I have given you the answer for two of the six cases needed for the recursive definition.)

$$\begin{aligned} D(\emptyset) &= \emptyset \\ D(\varepsilon) &= \underline{\qquad\qquad\qquad} \\ D(a) &= \underline{\qquad\qquad\qquad}, \text{ for any } a \in \Sigma \end{aligned}$$

If R_1 and R_2 are regular expressions then,

$$\begin{aligned} D(R_1 \cup R_2) &= D(R_1) \cup D(R_2) \\ D(R_1 \circ R_2) &= \underline{\qquad\qquad\qquad} \\ D(R_1^*) &= \underline{\qquad\qquad\qquad} \end{aligned}$$

(d) Prove that for all regular expressions R, if L is the language represented by R, then the language represented by $D(R)$ is $DROP(L)$.

In other words, explain why your answer to part (c) is correct for all R.

Hint: use mathematical induction on the number of operators $(\cup, ^*, \circ)$ that appear in the regular expression R.

(e) Prove that if L is any regular language then $DROP(L)$ is also a regular language.