This lab is about strings and regular expressions. Recall the definition and properties of the concatenation operator between strings.

Lemma 1: Concatenating nothing does nothing: For every string w, we have $w \bullet \varepsilon = w$.

Lemma 2: Concatenation adds length: $|w \bullet x| = |w| + |x|$ for all strings w and x.

Lemma 3: Concatenation is associative: $(w \bullet x) \bullet y = w \bullet (x \bullet y)$ for all strings w, x, and y.

---

1. Strings over the alphabet $\{0, 1\}$ are called boolean strings. For a boolean string w, define the bitwise complement $c(w)$ inductively as follows: $c(\epsilon) = \epsilon$, $c(0) = 1$, $c(1) = 0$, and $c(au) = c(a)c(u)$. Reversal is defined as always: $r(au) = r(u)a$ with base case $r(\epsilon) = \epsilon$.

   Prove that $r(c(w)) = c(r(w))$ for all strings w. You can assume a lemma that says for all u, v, $c(uv) = c(u)c(v)$.

2. Give regular expressions that describe each of the following languages over the alphabet {`0`,`1`}. We won't get to all of these in section.

   (a) All strings containing at least three `0`s.

   (b) All strings containing at least two `0`s and at least one `1`.

   (c) All strings containing the substring `000`.

   (d) All strings *not* containing the substring `000`.

   (e) All strings in which every run of `0`s has length at least 3.

   (f) Every string except `000`. *[Hint: Don't try to be clever.]*

   (g) All strings w such that *in every prefix of* w, the number of `0`s and `1`s differ by at most 1.

   ⋆(h) All strings w such that *in every prefix of* w, the number of `0`s and `1`s differ by at most 2.

   ★(i) All strings in which the substring `000` appears an even number of times.
   (For example, `0001000` and `0000` are in this language, but `00000` is not.)
