

The following problems ask you to prove some “obvious” claims about recursively-defined string functions. In each case, we want a self-contained, step-by-step induction proof that builds on formal definitions and prior results, *not* on intuition. In particular, your proofs must refer to the formal recursive definitions of string length and string concatenation:

$$|w| := \begin{cases} 0 & \text{if } w = \varepsilon \\ 1 + |x| & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

$$w \cdot z := \begin{cases} z & \text{if } w = \varepsilon \\ a \cdot (x \cdot z) & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

You may freely use the following results, which are proved in the lecture notes:

Lemma 1: $w \cdot \varepsilon = w$ for all strings w .

Lemma 2: $|w \cdot x| = |w| + |x|$ for all strings w and x .

Lemma 3: $(w \cdot x) \cdot y = w \cdot (x \cdot y)$ for all strings w , x , and y .

The *reversal* w^R of a string w is defined recursively as follows:

$$w^R := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ x^R \cdot a & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

For example, **STRESSED^R = DESSERTS** and **WTF374^R = 473FTW**.

1. Prove that $|w| = |w^R|$ for every string w .
2. Prove that $(w \cdot z)^R = z^R \cdot w^R$ for all strings w and z .
3. Prove that $(w^R)^R = w$ for every string w .

[Hint: You need #2 to prove #3, but you may find it easier to solve #3 first.]

To think about later: Let $\#(a, w)$ denote the number of times symbol a appears in string w . For example, $\#(X, \text{WTF374}) = 0$ and $\#(0, \text{000010101010010100}) = 12$.

4. Give a formal recursive definition of $\#(a, w)$.
5. Prove that $\#(a, w \cdot z) = \#(a, w) + \#(a, z)$ for all symbols a and all strings w and z .
6. Prove that $\#(a, w^R) = \#(a, w)$ for all symbols a and all strings w .