Computation Theory (COMP 170), Fall 2020
Recitation 01

---

[1] Induction Practice

We define the reverse of a string recursively as follows:

$$rev(\varepsilon) = \varepsilon,$$
$$rev(w \cdot \mathtt{a}) = \mathtt{a} \cdot rev(w)$$

for all $\mathtt{a} \in \Sigma$ and $w \in \Sigma^*$. Use this definition and induction on the length of v to prove that

$$\text{for all } u, v \in \Sigma^*, rev(u \cdot v) = rev(v) \cdot rev(u).$$

For best results, use the induction proof paradigm resource to help structure your proof.

[2] DFA Construction

Show that each of the following languages are regular by constructing a DFA that accepts it:

a. $A = \{x \in \{a, b\}^\star \mid \text{ the second character of } x \text{ is a } b\}$

b. $B = \{x \in \{a, b\}^\star \mid \text{ the second to last character of } x \text{ is a } b\}$

You do not need to prove the correctness of your construction. However, you should give a brief explanation, and make sure any diagrams are laid out in a clear and logical manner (and labeled, when appropriate.)

[3] Twisted

For a language A, let $twisted(A)$ be the language created by taking the even-length strings in A and exchanging every even-indexed character with the character that follows it. E.g. if

$$A = \{\mathtt{cat}, \mathtt{bear}, \mathtt{monkey}\}$$

then

$$twisted(A) = \{\mathtt{ebra}, \mathtt{omknye}\}.$$

Show that if A is regular, then $twisted(A)$ is also regular. In particular, given a general DFA $M = (Q, \Sigma, \delta, s, F)$ accepting A, show how to construct a DFA $M' = (Q', \Sigma, \delta', s', F')$ accepting $twisted(A)$. You do not need to prove your construction is correct, but should provide an explanation of how your DFA works.