## Computation Theory (COMP 170), Fall 2020
## Assignment 02

---

Answer each problem below to the best of your ability. Submit all parts by 9:00 AM on Monday, September 28. List your collaborators. Late homework is accepted within 24 hours for half credit. After 24 hours no credit is given. The first late assignment (up to 24 hours) per student incurs no penalty. **Make sure that your submission follows the formatting guidelines given at the end of this document.**

---

**Reading:** Sipser Chapters 1.2, 1.4

[ 1 ] *(10 pts.)*     **Close**

We'll say two strings are *close* if they are the same length and differ in exactly one character. That is, if $x, y \in \Sigma^*$, $a, b \in \Sigma$, and $a \neq b$, then $xay$ and $xby$ are close.

Given a language $A$ over the alphabet $\Sigma$, define

$$Close(A) = \{x \mid x \text{ and } y \text{ are close}, y \in A\}.$$

a. Give an example of a simple, infinite language $A$ over some alphabet $\Sigma$ for which $A \cap Close(A) = \emptyset$.

b. Give an example of a simple, infinite language $A$ over some alphabet $\Sigma$ for which $A = Close(A)$.

c. Show that if $A$ is regular, then $Close(A)$ is regular too. To do this, you should provide a precise procedure for constructing a finite automaton, and an explanation of how it works. In particular, you should make sure it's clear what the states in your new machine represent.

[ 2 ] *(5 pts.)* **Close But Not Quite**

Given a language $A$ over the alphabet $\Sigma$, define

$$CloseButNot(A) = \{x \mid x \text{ and } y \text{ are close}, y \in A, x \notin A\}.$$

Prove that if $A$ is regular, than so is $CloseButNot(A)$. You can (and probably should) make use of the conclusion of problem 1.

[ 3 ] (*5 pts.*) **All-NFA** (Sipser, problem 1.38)

An *all*-NFA $M$ is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if *every* possible state that $M$ could be in after reading input $x$ is a state from $F$. Note, in contrast, that an ordinary NFA accepts a string if *some* states among these possible states is an accept state. Prove that all-NFAs recognize the class of regular languages.

**Hint:** This proof has two directions (every regular language is recognized by some all-NFA; and every all-NFA recognizes a regular language).

**Format requirements**: work for COMP 170 should correspond to the following guidelines:

- Work must be in type-written format, with any diagrams rendered using software to produce professional-looking results. No hand-written or hand-drawn work will be graded.
- Work must be submitted in PDF format to Gradescope.
- Each answer should start on a new page of the document. When possible, try to limit answers to a single page each. (Thus, the answers to this homework must be no less than three pages, and preferably no more.)

You can find links to information about using LaTeX to produce type-written mathematical work,[1] and to a handy web-based tool for drawing finite-state diagrams, on the Piazza class site:

https://piazza.com/tufts/fall2020/comp170/resources

[1] LaTeX was used to produce this document.