Computation Theory (COMP 170), Fall 2020
Test 1

- Answer each problem below to the best of your ability.
- Submit all parts with 72 hours of downloading the exam, and before the start of class at 9am on Monday, October 19th.
- You may use your notes, the textbook, and materials from the Resources section of the course piazza page. No other resources may be used. Your work must be entirely your own.
- Make sure that your submission follows the formatting guidelines given at the end of this document.

---

[1] (*8 pts.*) **Shorties**

a. If A is non-regular, must the complement of A (i.e. $\overline{A}$) be non-regular too? Explain briefly.

b. If A and B are non-regular languages, must their union $(A \cup B)$ be non-regular too? Explain briefly.

c. If $N = (Q, \Sigma, \Delta, S, F)$ is an NFA and we define $N' = (Q, \Sigma, \Delta, S, Q \setminus F)$, is $L(N')$ the complement of $L(N)$? Explain briefly. (Note $\setminus$ means set difference, so $Q \setminus F$ is all states that are in Q but not in F.)

d. Let A denote the set strings over $\Sigma = \{a, b, c\}$, that do not include all character of Σ. E.g. A includes the strings *bb* and *cac*, but not *abac*. Draw a 3-state NFA that accepts A.

[2] (*8 pts.*) **Finite Automata**

Given two strings v and w over an alphabet Σ such that $|v| = |w|$, the *mix* of v and w is the string formed by adding characters alternately from v and w, in starting with the first character in v. So $mix(\texttt{cat}, \texttt{dog}) = \texttt{cdaotg}$. We can now define a *mix* operation on languages A, B as follows:

$$mix(A, B) = \{mix(v, w) \mid v \in A, w \in B, |v| = |w|\}$$

Show that if A and B are regular languages over some alphabet Σ, then $mix(A, B)$ is regular too. Be sure to give a precise, mathematical specification for the DFA or NFA you create. No proof is required, but be sure to explain how your machine works.

While not required, you will probably find it useful to have your solution start with the following assumptions and notation:

Suppose A and B are regular. There must exist DFA's M_A and M_B where

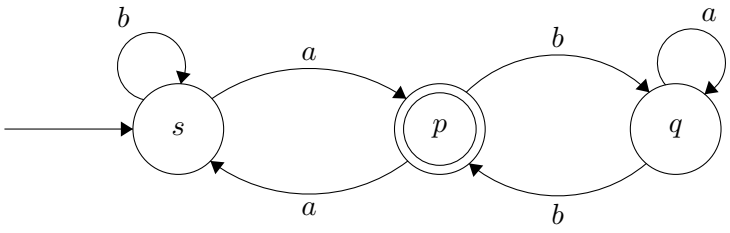
$$M_A = (Q_A, \Sigma, \delta_A, s_A, F_A),$$

$$M_B = (Q_B, \Sigma, \delta_B, s_B, F_B)$$

such that $L(M_A) = A$ and $L(M_B) = B$.

[3] (*8 pts.*) **Machines to Expressions**

Below is a 3-state DFA M. Use the procedure from class and the text to create a regular expression R such that $L(R) = L(M)$.



[4] (*8 pts.*) **Non-Regular**

Let A be the set of all odd-length strings over $\{a, b\}$ whose middle character is a. Use the pumping lemma to prove that A is not regular.

[5] (*8 pts.*) **CFGs**

Give a context free grammar that generates the language of non-palindromes over $\Sigma = \{a, b\}$, and provide a brief explanation. No proof required.

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.

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.