Computation Theory (COMP 170), Fall 2020
Assignment 01

Answer each problem below to the best of your ability. Submit all parts by 9:00 AM on Monday, September 21. 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 0, 1.1

[1] (*8 pts.*) **Delta Hat**

In class, we've seen how a δ function takes in a state and a character, and yields a state. We've somewhat cavalierly been pretending that this function works not only on single characters, but arbitrary strings. Indeed, this is a pretty natural generalization of δ — given a string, we repeatedly apply δ to each character, updating the state as we go. Here we'll make sure we're above board by formally capturing this idea by defining the $\hat{\delta}$ (pronounced "delta hat") function.

For any DFA transition function $\delta : Q \times \Sigma \to Q$, we define

$$\hat{\delta} : Q \times \Sigma^{\star} \to Q$$

inductively on the length of the input string as follows:

$$\hat{\delta}(q, \varepsilon) = q \text{ for all } q \in Q$$
$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a) \text{ for all } q \in Q, x \in \Sigma^{\star}, a \in \Sigma$$

The first part of the definition is a base case, and simply states that $\hat{\delta}$ doesn't change states on an empty string. The second part is the inductive step: to process a string, inductively process all but the last character, and from the resulting state, transition according to δ on the last character. Take a minute to convince yourself that this definition lines up with your understanding.

Having this definition allows us to more concisely and precisely define the language of a DFA. In particular, we can now say that $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(s, x) \in F\}$.

a. (*2 pts.*) Use the definition of $\hat{\delta}$ to prove that $\hat{\delta}$ is the same as δ when applied to single characters. That is, prove that for any state $p \in Q$ and character $a \in \Sigma$,

$$\delta(p, a) = \hat{\delta}(p, a).$$

b. (*6 pts.*) Let δ_C be the transition function used in class for the product construction. That is,

$$\delta_C((p, q), a) = (\delta_A(p, a), \delta_B(q, a))$$

Intuitively, C simulates A on the first coordinate, and B on the second. To formally argue this, we need to prove something about the behavior of $\hat{\delta}_C$. Namely, we need to show that for all states $p \in Q_A$, $q \in Q_B$, and strings $x \in \Sigma^{\star}$,

$$\hat{\delta}_C((p,q),x) = (\hat{\delta}_A(p,x), \hat{\delta}_B(q,x))$$

Prove this using induction on the length of x. For best results, use the induction proof paradigm resource to help structure your proof.

[2] (*6 pts.*) **DFA Constructions**

Show that the following languages are regular by drawing a DFA that accepts each of them.

a. $A = \{x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^\star \mid$ exactly one character in $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ appears zero times in $x\}$

b. $B = \{x \in \{a, b\}^\star \mid x = yzy$ for some $y, z \in \{a, b\}^\star, |y| = 2\}$

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

[3] (*6 pts.*) **Doubled**

For a language A, let $doubled(A)$ be the language created by taking each string x in A and replacing each character of x by that character twice. E.g. if

$$A = \{\texttt{cat}, \texttt{monkey}\}$$

then

$$doubled(A) = \{\texttt{ccaatt}, \texttt{mmoonnkkeeyy}\}.$$

Show that if A is regular, then $doubled(A)$ 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 $doubled(A)$. You do not need to prove your construction is correct, but should provide an explanation of how your DFA works.

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.