CS 332: *Theory of Computation* | Prof. Ran Canetti
Boston University | September 8, 2020

Homework 1 – Due Tuesday, September 15, 2020 <u>before</u> 2:00PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises Please practice on exercises and solved problems in Chapters 1. The material they cover may appear on exams. It's also a good idea to practice constructing DFAs and NFAs for specific languages using `http://automatatutor.com/`.

Problems There are 3 mandatory problems and one bonus problem.

1. Give state diagrams of DFAs with as few states as you can recognizing the following languages.

   (a) The alphabet is the set of Nucleic acids $\Sigma = \{A, C, G, T\}$.
   $L_1 = \{w \mid w \text{ is a sequence of correct pairings of Nucleic acids}\}$. (Recall that the correct pairings are $\{A, T\}$ and $\{C, G\}$). In other words, a string is in L_1 if the first and second acids are a corect pairing, the third and fourth acids are a correct pairing, and so on.

   (b) $\Sigma = \{b, e, o\}$.
   $L_2 = \{w \mid w \text{ contains the word “oboe” as a substring}\}$.

   (c) $\Sigma = \{0, 1\}$.
   $L_3 = \{w \mid w \text{ represents a } \textit{valid} \text{ binary number in big endian that is a multiple of } 3\}$. In other words, and the number is presented starting from the most significant bit and *cannot* have leading 0s. For example, 0, 11, 110, 1001 are in L_3, but 00, 1, 011 are not.
   $L_4 = \{w \mid w \text{ represents a binary number in little endian that is congruent to 1 modulo 3}\}$. In other words, the number equals $3n+1$ for some integer $n \geq 0$, and the number is presented starting from the least significant bit and *can* have trailing 0s. For example, 1, 0010, 111 are in L_4.

   Give state diagrams of NFAs with as few states as you can recognizing the following languages:

   (d) $\Sigma = \{1\}$.
   $L_5 = \{w \mid \text{the length of } w \text{ is divisible by 3 or by 5}\}$.

   (e) $\Sigma = \{a, d\}$.
   $L_6 = \{w \mid w \text{ contains substrings “add” and “dad” which do not overlap}\}$.

   You can optionally provide some reasoning for your DFA/NFA, which could be accounted for partial credits if the construction is not completely correct.

2. (a) For any finite set Σ and finite set $L \subseteq \Sigma^*$, prove that L is regular.

   (b) Let $\Sigma = \{0, 1\}$. Prove that for any $k > 0$, the language $L_k := \{w \in \Sigma^* \mid w \text{ ends with } 0^k\}$ can be recognized by an NFA, where 0^k means the 0 character repeated k times.

3. We call string x is a *prefix* of a string y if a string z exists where $xz = y$, and that x is a *proper prefix* of y if in addition $x \neq y$. Show that if a language A is regular, then $NOEXTEND(A) = \{w \in A \mid w \text{ is not the proper prefix of any string in } A\}$ is also a regular language.

4. **Bonus problem, no collaboration is allowed**

   Given a regular language L.

   (a) Let $f_k(L) := \{a \mid \exists b : ab \in L \text{ and } |b| = k\}$. Prove that for every $k > 0$, $f_k(L)$ is regular. (hint: start with f_1)

   (b) Let $double(L) := \{a \mid \exists b : ab \in L \text{ and } |b| = |a|\}$. Prove that $double(L)$ is regular. (hint: notice that counterintuitively the language $f_k(L)$ does not become "harder" when k goes to infinity)

   (c) Let $exp(L) := \{a \mid \exists b : ab \in L \text{ and } |b| = 2^{|a|}\}$. Prove that $exp(L)$ is regular.