CS 332: *Theory of Computation* — Prof. Ran Canetti
Boston University — September 15, 2020

Homework 2 – Due Tuesday, September 22, 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 Chapter 1 and on the exercise below. The material they cover may appear on exams.

1. (**Conversion procedures**) Use asymptotic (big-O) notation to answer the following questions. Provide brief explanations.

   (a) Let N be an NFA that has n states. If we convert N to an equivalent DFA M using the procedure we described, how many states would M have?

   (b) Let R be a regular expression that has n symbols (each constant/operation counts as one symbol). If we convert R to an equivalent NFA N using the procedure described in class, how many states would N have in the worst case?

   (c) Let M be a DFA that has n states. If we convert M to an equivalent regular expression R using the procedure we described, how many symbols would R have in the worst case?

Problems There are 3 mandatory problems and one bonus problem. You may use anything we covered in the lectures, including the pumping lemma and the closure of the class of regular languages under union, intersection, complement, and reverse.
Hint: Note that sometimes to prove that a language is not regular, you will not be able to use the pumping lemma directly on the language in question – instead you will need to use it on a related language (as was done in class).

1. (**Regular languages**) For every language over the binary alphabet $\Sigma = \{0, 1\}$ given below, either prove that it is regular or prove that it is not regular. To prove that a language is regular, prov that it is recognized by an NFA or DFA. To prove that a language is not regular you may use either the pumping lemma or any other method you wish.

   (a) (**6 points**) $L_1 = \{0^n x 1^n \mid x \in \Sigma^*, n \geq 1\}$.

   (b) (**8 points**) $L_2 = \{xyyx \mid x, y \in \Sigma^*\}$.

   (c) (**8 points**) $L_3 = \big\{xy \mid x, y \in \Sigma^*$ and x represents a valid binary number in little endian and $|y| = x\big\}$.

   (d) (**10 points**) $L_4 = \big\{a_1b_1a_2b_2...a_nb_n \mid n \geq 0$ and $a_1a_2\cdots a_n, b_1b_2\cdots b_n$ represents two binary numbers x, y in big endian and $x = y^2\big\}$.

   (e) (**10 points**) $L_5 = \big\{0^i1^j0^k \mid i, j, k \geq 0$ and if $i = 1$ then $j = k\big\}$.

2. (**Closure of regular languages**) In each of the following parts, we define an operation on a language A. Prove or disprove that the class of regular languages is closed under that operation.

(a) (**8 points**) $UNARY(A) = \{1^n \mid w \in A \text{ and } w \text{ represents a valid binary number } n \text{ in little endian}\}$. For example, $UNARY(\{010, 1, 011\}) = \{1, 111111\}$, since 010 has trailing zeroes, and 1, 011 represent 1, 6 respectively.

(b) (**12 points**) $NOPREFIX(A) = \{w \in A \mid \text{no proper prefix of } w \text{ is a member of } A\}$. A string x is a *prefix* of string y if a string z exists where $xz = y$, and that x is a *proper prefix* of y if in addition $x \neq y$.

(c) (**16 points**) $ADDONE(A, B) = \{xz \in A \mid \exists y : xyz \in B \text{ and } |y| = 1\}$.
Hint: Start by considering A to be the language of all strings.

3. **(Shortest string)** Consider two DFAs M_1 and M_2 with k_1 and k_2 states, respectively, and languages $A_1 = L(M_1)$ and $A_2 = L(M_2)$.

   (a) (**10 points**) Show that if $A_1 \neq \emptyset$, it contains a string of length less than k_1.

   (b) (**10 points**) Let $U = A_1 \cup A_2$. Show that if $U \neq \emptyset$, it contains a string of length less than $\max(k_1, k_2)$.

   (c) (**12 points**) Show that if $U \neq \Sigma^*$, then U excludes some string of length less than $k_1 k_2$.
   Think, but do not hand in: Give an example of M_1 and M_2 where the shortest excluded string is as close as possible to $k_1 k_2$.

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

   Let's consider a function $f : \Sigma \rightarrow \Gamma^*$ from one alphabet to strings over another alphabet. A fuction of this type is called a *homomorphism*.

   We can extend the domain of this function to all strings over alphabet Σ by defining $f(w) = f(w_1)f(w_2)\dots f(w_n)$, where $w = w_1 w_2 \dots w_n$ and $w_i \in \Sigma$ if w is not empty, and $f(\varepsilon) = \varepsilon$ in case it is. We also can further extend f to operate on languages by defining $f(A) = \{f(w) \mid w \in A\}$, for any language A.

   Show that the class of regular languages is closed under extended homomorphisms. In other words, given a DFA M that recognizes a language $B \subseteq \Sigma^*$ and an extended homomorphism $f : \Sigma^* \rightarrow \Gamma^*$, construct a finite automaton N that recognizes the $f(B) \subseteq \Gamma^*$.