CS 332: *Theory of Computation* Professor Mark Bun
Boston University February 5, 2020

Homework 2 – Due Monday, February 10, 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 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.

1. (**Non-regular languages**) Prove that the following languages are not regular. You may use the pumping lemma and the closure of the class of regular languages under union, intersection, complement, and reverse.

   (a) $L_1 = \{0^n1^m \mid n, m \geq 0 \text{ and } n = m^2\}$.

   (b) $L_2 = \big\{w \mid w \in \{0,1\}^* \text{ is not a palindrome}\big\}$. A *palindrome* is a string that reads the same forward and backward.

   (c) $L_3 = \big\{1^k y \mid y \in \big\{0,1\big\}^* \text{ and } |y| = k\big\}$.

   (d) $L_4 = \big\{a^i b^j c^k \mid i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\big\}$.

   (e) L_4 satisfies conditions of the pumping lemma. Explain why this does not contradict the fact that L_4 is not regular.

2. **(Number of states)** Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. For each $k \geq 1$, let C_k be the language consisting of all strings that contain an $\mathtt{a}$ exactly k places from the right-hand end. Thus $C_k = \Sigma^*\mathtt{a}\Sigma^{k-1}$.

   (a) Describe an NFA with $k+1$ states that recognizes C_k in terms of both a state diagram and a formal description.

   (b) If a DFA enters different states after reading two different input strings xz and yz with the same suffix z then the DFA must enter different states after reading input strings x and y. Explain why.

(c) Find 2^k strings on which every DFA recognizing C_k must enter different states. (Hint: Start by finding 2 such strings).

(d) Prove that for every k, no DFA with fewer than 2^k states can recognize C_k. (Hint: Combine parts (b) and (c).)

3. (**DFAs, NFAs, regular expressions and converting between them**)

(a) (**Description to NFA**) Let $\Sigma = \{\texttt{A}, \texttt{O}, \texttt{L}\}$. Give an NFA with 4 states recognizing the language $\{w \in \Sigma^* \mid w \text{ ends in } \texttt{LOL} \text{ or in } \texttt{LL}\}$.

(b) (**NFA to DFA**) Convert your NFA from part (a) to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

(c) (**Rex to NFA**) Use the procedure described in class (also in Sipser, Lemma 1.55) to convert $(\mathsf{T}(\mathsf{GGA})^* \cup \mathsf{C})^*$ to an equivalent NFA. Simplify your NFA.

(d) (**Description to rex**) Give a regular expression that generates the following language:

$$\{w \mid w \text{ is a binary string and the number of 0s in } w \text{ is divisible by } 3\}.$$