CS 332: *Theory of Computation*
Boston University

Prof. Ran Canetti
September 22, 2020

Homework 3 – Due Tuesday, September 29, 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 procedure, cont'd**) In an *extended* regular expression, we may use the complement operation ($\neg$) in addition to the three regular operations ($\cup, \circ, \star$). For example,

$$\neg(\Sigma^*001\Sigma^*) \cup \neg(\Sigma^*100\Sigma^*)$$

is an extended regular expression that describes the collection of all strings that either do not contain the substring 001 or do not contain the substring 100.

   (a) Describe how to modify the conversion procedure from regular expressions to NFAs so that it becomes a conversion procedure from extended regular expressions to NFAs.

   (b) Let R be an *extended* 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 you described above, asymptotically how many states would N have in the worst case? Provide brief explanations.

Problems There are 3 mandatory problems and one bonus problem.

1. (**Regular expressions**) For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, show a regular expression that recognizes the language. You are again encouraged to write out explanations for your regular expression as (1) if your construction is incorrect we can assign you partial credits for the explanations and (2) it could also help yourself find bugs (see also `https://rubberduckdebugging.com`).

   (a) (**10 points**) $L_a = \{w \mid |w| \equiv 1 \pmod 3\}$.

   (b) (**10 points**) $L_b = \{w \mid w \text{ contains exactly four 0's}\}$.

   (c) (**10 points**) $L_c = \{w \mid w \text{ contains at least two 0's or exactly two 1's}\}$.

   (d) (**10 points**) $L_d = \{w \mid w \text{ contains at least two 0's and exactly two 1's}\}$.

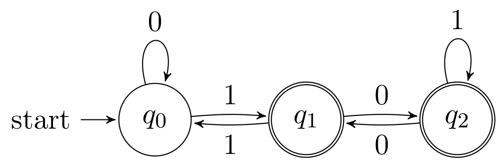
   (e) (**10 points**) The complement of the language defined by $(\varepsilon \cup 1 \cup 11)(0 \cup 01 \cup 011)^*$.

2. (**Rex to DFA**) Let $\Sigma = \{0, 1\}$.

   (a) (**20 points**) Use the procedure described in class (also in Sipser, Lemma 1.55), give an NFA for the following regular expression $\Sigma^*(11|101)$. Simplify your NFA.

   (b) (**20 points**) Convert your NFA from part (b) to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

3. (**DFA to rex, 20 points**) Use the procedure described in Lemma 1.60 to convert the following DFA (which recognizes binary number in big endian that is not a multiple of 3) to a regular expression.



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

   (a) Let $L \subseteq 0^*$ be an arbitrary unary language. Prove that L^* is regular.

   (b) Prove that there exists a binary language $L \subseteq \{0,1\}^*$ such that L^* is not regular.