Homework 2 – Due Friday, September 21, 2018 before noon

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?

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.

(d) 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.

(e) 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, how many states would $N$ have in the worst case?

Problems This homework contains 1 automatically graded problem. Your solution to all other problems should be submitted on Gradescope. There are 3 mandatory problems, worth 10 points each.

1. (DFAs, NFAs, regular expressions and converting between them) Your solution to this problem will be automatically graded by [automatatutor.com](http://automatatutor.com) (We will rescale the points assigned by Automata Tutor, so that each part is worth 2 points.)

(a) (Description to NFA) Let $\Sigma$ contain all letters and punctuation marks used in English. Give an NFA with 4 states recognizing the language \{w ∈ $\Sigma^*$ | w contains 10 or 1H0\}.
(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 \((T(GGA)^* \cup C)^*\) to an equivalent NFA. Simplify your NFA.

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

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

(e) **NFA to rex** Use the procedure described in Lemma 1.60 to convert the following finite automaton to a regular expression.

---

2. **(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) **(3 points)** Show that if \(A_1 \neq \emptyset\), it contains a string of length less than \(k_1\).

(b) **(3 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) **(4 points)** Show that if \(U \neq \Sigma^*\), then \(U\) excludes some string of length less than \(k_1k_2\).

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 \(k_1k_2\).

3. **(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) **(2 points)** \(L_1 = \{0^n1^m \mid n,m \geq 0 \text{ and } n = m^2\}\).

(b) **(2 points)** \(L_2 = \{w \mid w \in \{0,1\}^* \text{ is not a palindrome}\}\). A palindrome is a string that reads the same forward and backward.

(c) **(3 points)** \(L_3 = \{1^k y \mid y \in \{0,1\}^* \text{ and } |y| = k\}\).

(d) **(3 points)** \(L_4 = \{a^i b^j c^k \mid i,j,k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}\).

Hint: Be careful! \(L_4\) satisfies the conditions of the pumping lemma.