# Homework 1 – Due Monday, February 3, 2020 before 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) $L_1 = \{w \mid w$ begins with 10 and ends with 11$\}$.

   (b) $L_2 = \{w \mid w$ represents a binary number that is congruent to 2 modulo 3$\}$. In other words, this number minus 2 is divisible by 3. The number is presented starting from the most significant bit and can have leading 0s.

   (c) $L_3 = \{w \mid w$ is a string of the form $x_1y_1x_2y_2 \ldots x_ny_n$ for some natural number $n$ such that if $x$ is the integer with binary representation $x_1x_2 \ldots x_n$ and $y$ is the integer with binary representation $y_1y_2 \ldots y_n$ then $x > y\}$. Both $x$ and $y$ are presented starting from the most significant bit and can have leading 0s.

   (d) $L_4 = \{w \mid w$ represents a possible series of flips in the game in which player 1 wins where H represents heads and T represents tails$\}$ where the game goes like this: Repeatedly flip a coin. On heads, player 1 gets a point. On tails, player 2 gets a point. A player wins (and the game ends) as soon as they are ahead by two points.

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

   (e) $L_5 = \{w \mid$ the third symbol from the end in $w$ is 0$\}$.

   (f) $L_6 = \{w \mid w$ contains substrings 100 and 00 which do not overlap$\}$.

2. In class, we have shown that if $M$ is a DFA that recognizes a language $A$ then we can obtain a DFA $M'$ that recognizes $\overline{A}$ by swapping accept and non-accept states in $M$.

   (a) Show, by giving an example, that if $M$ is an NFA that recognizes a language $B$ then we do not necessarily obtain an NFA recognizing $\overline{B}$ by swapping accept and non-accept states in $M$.

   (b) Is the class of languages recognized by NFAs closed under complement? Explain your answer.

3. (a) Given a string $w$ of 0s and 1s, the *flip* of $w$ is obtained by changing all 0s in $w$ to 1s and all 1s in $w$ to 0s. Given a language $A$, the flip of $A$ is the language $\{w \mid$ the flip of $w$ is in $A\}$. Prove that the class of regular languages is closed under the flip operation.

(b) Given languages $A$ and $B$ over alphabet $\Sigma$, the *interleaving* of $A$ and $B$ is defined as

$$\{w \mid w = a_1 b_1 \ldots a_k b_k, \text{ where } a_1 \ldots a_k \in A \text{ and } b_1 \ldots b_k \in B, \text{ each } a_i, b_i \in \Sigma^*\}.$$

Prove that the class of regular languages is closed under interleaving. Be careful: each $a_i, b_i$ is a (possibly empty) string of symbols.

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

Let's consider a function $f : \Sigma \to \Gamma^*$ 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 $\Sigma$ by defining $f(w) = f(w_1)f(w_2)\ldots f(w_n)$, where $w = w_1 w_2 \ldots w_n$ 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, by giving a formal construction, that the class of regular languages is closed under homomorphism. In other words, given a DFA $M$ that recognizes $B$ and a homomorphism $f$, construct a finite automaton $N$ that recognizes $f(B)$. Consider the machine $N$ that you constructed. Is it a DFA in every case?