Introduction to the Theory of Computation
Homework #2

Note: Working with others on this homework is *allowed* and carries no penalty. However, you must do your own writeup, and you must state on your homework who you worked with. Due on Thursday, February 20th.

1. (Exercise 1.4 a, b, i and l) Draw diagrams of DFAs that recognize the following languages. In all cases the input alphabet is $\Sigma = \{0, 1\}$.

- $\{w \mid w \text{ begins with a 1 and ends with a 0}\}$
- $\{w \mid w \text{ contains at least three 1s}\}$
- $\{w \mid \text{every character at an odd position of } w \text{ is a 1}\}$
- $\{w \mid w \text{ contains an even number of 0s, or exactly two 1s}\}$

2. (Exercise 1.5c) Give an NFA with six states that recognizes the last language in the previous exercise.

3. (Exercise 1.13) Give regular expressions for all four languages in Exercise 1.4 above.

4. (Exercise 1.10b) We saw in class that switching the accepting and non-accepting states of a DFA that recognizes the language L yields a DFA that recognizes the language $\overline{L}$. Show by example that this is not true for NFAs; that is, show an NFA where switching the accepting and non-accepting states does not complement the language it recognizes.

5. Let's formalize the idea of strings that are equivalent if they can be followed by the same suffixes. For a given language L, say $u \sim v$ if, for all w, $uw \in L$ if and only if $vw \in L$. This equivalence divides the set of strings Σ^* up into equivalence classes.

Suppose L is recognized by a DFA M. Now using the definition of δ^* we gave in class, prove formally that if $\delta^*(q_0, u) = \delta^*(q_0, v)$ — that is, if u and v lead to the same state of M from the start state — then $u \sim v$. (This proof only takes a few lines in our notation.)

Conclude from this that the number of states of M must be at least as large as the number of equivalence classes defined by $\sim$.

Now prove formally that the set of palindromes is not a regular language.

6. (Exercise 1.17c plus a little more) Give two proofs, one based on the *pumping lemma* (which will will talk about this week) and one based on the previous exercise, that the language $\{a^{2^n} \mid n \in \mathbb{N}\}$ is not regular. Here $\Sigma = \{a\}$ and a^k means a string of k a's.

7. (Problem 1.31) Consider a new kind of automaton called an *All-Paths NFA*. This is just like an NFA, except we say it recognizes a word if *all* computation paths accept, unlike an NFA which accepts if at least one path does. (Note that what we have done logically is replace $\exists$, "there exists a path", with $\forall$, "for all paths".)

Show that All-Paths NFAs recognize exactly the regular languages, by describing how to convert an All-Paths NFA into an DFA that recognizes the same language.

8. (Problem 1.39) Prove that for all $k > 1$, DFAs with k states are more powerful than DFAs with $k - 1$ states. That is, come up with a family of languages L_k such that L_k can be recognized by a DFA with k states but not by a DFA with $k - 1$ states. Thus the DFAs form a strict hierarchy based on the number of states, which classify the regular languages according to their complexity.

9. (Problem 1.44 — a little harder) Now prove that NFAs can be exponentially more compact than DFAs, or, to put it differently, that the exponential blowup in the number of states we get when converting an NFA to a DFA is sometimes necessary. That is, come up with a family of languages L_k such that L_k can be recognized by an NFA with k states, but that the smallest DFA that recognizes it has $\Theta(2^k)$ states. Hint: Look at Example 1.14 in Sipser.

10. (Problem 1.42 — Tricky!) For a language L, define $L_{1/2}$ as

$$L_{1/2} = \{x \mid xy \in L \text{ for some word } y \text{ such that } |y| = |x|\}$$

That is, $L_{1/2}$ is the set of "first halves" of L, namely the set of words x that can be followed by words y of the same length giving a word xy in L. Prove that if L is regular, so is $L_{1/2}$.

11. (Problem 1.41) As we will discuss in class, the language

$$L = \{w \mid w \text{ has an equal number of 0s and 1s}\}$$

is not regular. However, prove that the language

$$L = \{w \mid w \text{ has an equal number of 01s and 10s}\}$$

is regular.