CS 172: Computability and Complexity, Spring 2010	**S. A. Seshia & O. Etesami**

HW 1: Introduction to DFAs and NFAs

Assigned: January 25, 2010 — *Due in class: February 1, 2010*

Note: Take time to write clear and concise solutions. Confused and long-winded answers may be penalized. Consult the course webpage for course policies on collaboration.

1. (5 points) In basic arithmetic, any formula X is *valid* if and only if it takes one of the following forms:

   - X is an integer
   - X is of the form $Y + Z$, where Y and Z are valid formulas
   - X is of the form $Y * Z$, where Y and Z are valid formulas
   - X is of the form (Z), where Z is a valid formula

   For example, the formulas 2, $3 + (2 + 9)$, $((14))$, and $(38 + (3) + 16 + (0 + 9))$ are valid, but the formulas $+2$, $(2(3))$, $2) + 3)$, and $(4 + 4$ are not

   Prove that any valid formula must have an equal number of left parentheses and right parentheses using a proof by induction.

   Hint: Show that the property holds on the simplest possible formula, then show that it is preserved during any possible expansions of this formula. Consider each possible expansion separately.

2. (6 points)

   (a) Design and draw a DFA which reads in a non-negative integer written in binary, least significant bit first, and accepts exactly when the input is divisible by four. (There is no sign bit in the representation.) Specify what Q, Σ, δ, q_0, and F are for your DFA.

   (b) Now suppose that before the least significant bit, there is a bit indicating the sign of the integer: 0 for a negative number, and 1 for a positive number. How would you modify your DFA to handle these signed integers?

3. (5 points) Consider the set of all binary strings where the difference between the number of 0s and the number of 1s is even.

   (a) Is this set a valid *alphabet* for a DFA? Why or why not?

   (b) Design and draw a DFA with as few states as possible recognizing this set.

4. (8 points) Let L be the language of all strings over the alphabet $\Sigma = \{a, b\}$ which does not contain the string aa and does not contain the string bb and has length at least 1.

   (a) Design and draw an NFA which recognizes L and uses only three states.

   (b) Explain each step of the accepting computation taken by your NFA on the string $baba$.

   (c) Can any DFA recognize this language with only three states? Give such a DFA or show why one cannot exist.

5. (6 points) Let L denote a language. The reversal of L, denoted L^R, is the language $\{x^R : x \in L\}$; i.e., the set of all strings whose reversal is in L. For example, if L is the set of all strings of length at least two whose first two bits are zero, then L^R is the set of all strings of length at least two whose last two bits are zero.

   (a) Suppose you are given a DFA M with alphabet Σ that recognizes L. How would you construct an NFA M^R that recognizes L^R? (Your NFA can have ϵ-transitions.) Prove that your construction is correct.

   (b) Does your construction work if M is non-deterministic?