Problem Set 4
CS 373: Theory of Computation

Assigned: February 7, 2013 Due on: February 14, 2013

Instructions: This homework has 3 problems that can be solved in groups of size at most 3. Please follow the homework guidelines given on the class website; submitions not following these guidelines will not be graded.

Recommended Reading: Lectures 7 and 8.

Problem 1. [Category: Comprehension+Design] Consider the language $L = \overline{\mathbf{L}((abb^*)^*)}$.

1. Construct a DFA recognizing L. You need not prove that your construction is correct. If you construct it using the algorithms described in class then you should show all your steps. If you construct the automaton directly then you should explain the intuition behind your construction clearly. **[5 points]**
2. Construct a regular expression for the language L. Again you don't need to prove your regular expression to be correct, but you should show all the steps in the construction. **[5 points]**

Problem 2. [Category: Comprehension+Design+Proof] For any string $w = w_1 w_2 \cdots w_n \in \Sigma^*$ (with $w_i \in \Sigma$), the reverse of w, denoted as w^R, is the string w in reverse order, i.e., $w^R = w_n w_{n-1} \cdots w_1$. For a language $A \subseteq \Sigma^*$, let $A^R = \{w^R \mid w \in A\}$.

1. Given a DFA M, construct an NFA that recognizes $(\mathbf{L}(M))^R$. **[5 points]**
2. Prove that the NFA constructed in the previous part is correct. **[5 points]**

Problem 3. [Category: Comprehension+Design+Proof] For languages $A \subseteq \Sigma^*$ and $B \subseteq \Sigma^*$, define the *avoids* operation as follows.

$$A \text{ avoids } B = \{w \in A \mid w \text{ doesn't contain any string in } B \text{ as substring}\}$$

1. Let $A = \{1001, 1111\}$ and $B = \{01, 10\}$. What is A avoids B? **[1 points]**
2. Let $A = \mathbf{L}((0 \cup 1)^*)$ and $B = \mathbf{L}(1^*)$. What is A avoids B? **[1 points]**
3. Prove that if A and B are regular then Aavoids$B$ is regular. You can either construct a DFA/NFA/regular expression for A avoids B (and then you don't have to prove that your construction is correct) or use previously established closure properties to prove this result. **[8 points]**