**Problem Set 3**  
**CS 373: Theory of Computation**  
Assigned: September 13, 2012  
Due on: September 20, 2012

**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 3,4,5 and 6.

**Problem 1.** [Category: Design+Proof] Let $\Sigma = \{0, 1\}$. Let $W_k = \{w_1w_2 \mid w_1, w_2 \in \Sigma^k$ and $w_1 \neq w_2\}$

1. Prove that, for every $k$, any DFA recognizing $W_k$ must have at least $2^k$ states.
2. Construct an NFA that recognizes $W_k$ with $O(k^2)$ states. You need not prove that your construction is correct, but you should clearly explain the intuition behind your design.

**Problem 2.** [Category: Comprehension+Design]

1. Describe the language of the following regular expressions. A clear, crisp one-level interpretable English description is acceptable, like “This is the set of all binary strings with at least three 0s and at most hundred 1s”, or like “$\{0^n(10)^m \mid n$ and $m$ are integers\}”. A vague, recursive or multi-level-interpretable description is not, like “This is a set of binary strings that starts and ends in 1, and the rest of the string starts and ends in 0, and the remainder of the string is a smaller string of the same form!” or “This is a set of strings like 010, 00100, 0001000, and so on”. You need not prove the correctness of your answer.

   (a) $(0^* \cup 0 \cup 1^*)^*$  
   (b) $1(00^*1)^*0^*$  
   (c) $1^*(0 \cup 111^*)^*1^*$  

2. Give regular expressions that accurately describe the following languages. You need not prove the correctness of your answer.

   (a) All binary strings such that if it starts with 0 it has odd length and if it starts with 1 it has even length.  
   (b) All binary strings such that in every prefix, the number of 0s and 1s differ by at most 1.  
   (c) All binary strings such that every pair of consecutive 0s appears before any pair of consecutive 1s.

**Problem 3.** [Category: Design] Let $\text{SUFFIX}(L) = \{y \in \Sigma^* \mid \exists x \in \Sigma^*, xy \in L\}$. Let $r, s, r_T, s_T$ be the regular expressions for languages $R, S, \text{SUFFIX}(R)$, and $\text{SUFFIX}(S)$, respectively. Using only these regular expressions and the operations $\cup$, concatenation, and $*$, give the regular expressions for the following languages:
(a) SUFFIX\(R \cup S\)  
(b) SUFFIX\(RS\)  
(c) SUFFIX\(R^*\)  

Using the solutions to (a), (b), and (c), give an inductive algorithm that constructs the regular expression for SUFFIX\(L\) from the regular expression for \(L\). You need not prove that your algorithm is correct.