CS 520: Theory of Computation	Homework 2
Out: 02/19/08	**Due: 03/04/08**

Ground Rules

- **(Grading)** You will be graded on the correctness as well as clarity of your solutions. Please state and prove any assumptions or claims that you make.
- **(Collaboration)** You are allowed to discuss questions with other people in the class. However, **you must solve and write your answers yourself without any help**. You must also give explicit citations to any sources besides the textbook and class notes, including discussions with classmates.
- **(Lateness)** Late submissions do not get any credit.
- Start working on your homework early. Plan your work in such a way that you have the opportunity to put some problems on the back burner for a while and revisit them later. Good luck!

Problems

1. **(10 pts)** Consider the following generalization of languages we discussed in class: For any two strings c and d, let $L_{c,d}$ denote the language of all binary strings x such that c and d occur an equal number of times as a substring of x.

   Determine whether the following languages are regular: (a) $L_{001,011}$, and, (b) $L_{001,100}$. How do your answers change when you consider the alphabet $\{0, 1, 2\}$ instead of $\{0, 1\}$?

   (Prove all your claims.)

2. **(6 pts)** Problem 1.49 in the book (pg. 90).

3. **(12 pts)** Recall that in class we defined the indistinguishability relation $\sim$ between pairs of states in a DFA. We will now extend this relation to strings. A pair of strings x and y, are said to be *distinguishable by a language* L if there exists a string $z \in \Sigma^*$ such that *exactly one* of the strings xz and yz is in L. Alternately, if for every string z, we have $xz \in L$ if and only if $yz \in L$, then x and y are called indistinguishable by L, written $x \sim_L y$.

   Verify for yourself that indistinguishability over strings is an equivalence relation. Let $n(L)$ denote the number of equivalence classes of $\sim_L$.

   (a) Prove that any DFA for L must contain at least $n(L)$ states.
   (b) Prove that for every regular language L, there exists a DFA for L with $n(L)$ states.
   (c) Conclude that a language L is regular if and only if $n(L)$ is finite. (You don't need to do anything here; just give this a thought and convince yourself that it holds.)
   (d) Use part (c) to prove that the language $\{0^k1^k | k \geq 0\}$ is not regular.

4. **(10 pts)** Show how to decide the language $L = \{0^k1^k | k \geq 0\}$ on a single-tape Turing machine in $O(n \log n)$ steps, where n represents the length of the input. (Note: Doing this in $O(n^2)$ steps should be easy; the challenge here is in doing this in $O(n \log n)$ steps.)

   First describe your strategy in words, and then inplement it on a Turing machine. Specify the meaning of all states of the Turing machine, and argue correctness.

5. **(12 pts)** Problem 3.20 in the book (pg. 162).