CS 262: Computational Complexity **Due: February 19, 2010**

Homework 4

Instructor: Sandy Irani

Do two of the three problems below.

1. An input to the problem ST-NON-CONN is a graph G along with vertices s and t in G. The language ST-NON-CONN consists of all triplets (G, s, t) such that there is not a path from s to t in G. Prove that ST-NON-CONN is complete for the class **co-NL**.

2. In the Arora-Boak text gives an alternative definition of the class **NL** which makes use of a Turing Machine with a special read-once tape. The head on a read-once tape starts at the left-most end of the non-blank symbols written on the tape and can only move to the right or stay in the same place (i.e. it can never move left). The alternative definition says that a language L is in **NL** if there is a deterministic Turing Machine M (called a *verifier*) with a special read-once tape and a polynomial p such that for every $x \in \Sigma^*$,

$$x \in L \Leftrightarrow \exists u \in \Sigma^{p(|x|)} \text{ such that } M(x, u) = 1,$$

   where $M(x, u)$ is the output of M when x is placed on the input tape and u is placed on its special read-once tape and M uses $O(\log n)$ space on its work tape for every input x.

   Prove that this definition is equivalent to the definition using non-deterministic Turing Machines discussed in class.

3. Prove that if in the above definition, the read-once tape is replaced with a read-only tape (which could be read many times), then the resulting class is **NP**.