CS 351: Problem Set 3
(assigned 10/17/2002; due 10/24/2002)

On each problem, you may work with at most one other person, but you must write up the solutions independently. Please turn in each problem on a separate sheet of paper that also contains your name, the name of your collaborator (if any), and other sources (eg., papers, books) that you consulted.

(1) Show that the language $\overline{\text{2SAT}} = \{\varphi \mid \varphi$ is a Boolean formula in CNF that has at most two literals per clause, and φ is *not* satisfiable$\}$ is in NL by designing an appropriate proof system.

(*Hint.* Read Theorem 9.1 in Papadimitriou's book, but write your solution in the form of a proof system, that is, describe the format of the proof, and describe a logspace verifier with one-way access to the proof, and argue briefly that the proof system is complete and sound. Additional note: you don't need the Immerman–Szelepscényi theorem.)

(2) *Problems in* PH. In this problem, we will place a few combinatorial problems in the polynomial hierarchy. In each case, place the problem in the lowest possible level of the hierarchy that you can, keeping in mind that it might belong to Σ_k^p and Π_k^p for some k.

(2a) *VC Dimension.* Given an $m \times n$ table A of 0's and 1's, A is said to *shatter* a set C of d columns if the restriction of A to the columns in C contains every possible string in $\{0,1\}^d$. The *Vapnik–Cervonenkis dimension (VC dimension)* of A is the largest d such that A shatters some set of d columns. The decision problem VCD is defined by VCD $= \{(A,d) \mid$ the VC dimension of A is $d\}$.

(2b) *Clique Number.* Given a graph G, define $\alpha(G)$, the *clique number* of G to be the size of the largest clique in G, that is, the number of vertices in the largest complete (vertex-induced) subgraph of G. The decision problem CLIQUE-NUMBER is defined by CLIQUE-NUMBER $= \{(G,k) \mid \alpha(G) = k\}$.

(2c) Problem 17.3.11 in Papadimitriou's book.

(3) We described a PSPACE-complete problem, the SUCCINCT-GAP problem, in class. A natural extension of the ideas used there can be used to show something stronger, and nicer, namely that the version of the Boolean CNF satisfiability problem that admits arbitrary number of alternating quantifiers is PSPACE-complete. Read the proof of Theorem 19.1 in Papadimitriou's book, where it is shown that QSAT is PSPACE-complete, and list the main ideas in the proof.