Theory of Computation

Final Examination on June 19, 2008
Spring Semester, 2008

Problem 1 (20 points)**.** Show that if SAT $\in$ P, then FSAT has a polynomial-time algorithm. (Hint: You may want to use the self-reducibility of SAT.)

Proof. Assume SAT $\in$ P. We describe below how to find a truth assignment to an input Boolean expression ϕ in time polynomial in $|\phi|$. If $\phi \notin$ SAT then it does not have a satisfying truth assignment. So we assume otherwise. Denote the variables of ϕ by $x_1, \ldots, x_n$. Let t be the empty truth assignment to $x_1, \ldots, x_n$. For $i = 1$ up to n, we expand t to include the assignment $x_i = \text{true}$ if $\phi[\,t \cup \{x_i = \text{true}\}\,] \in \text{SAT}$ and $x_i = \text{false}$ otherwise. Clearly, after n iterations, the final t will be a satisfying assignment of ϕ. It is also clear that the above procedure runs in time polynomial in $|\phi|$. $\square$

Problem 2 (20 points)**.** Let $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$ and $G = (U, V, E)$ be a bipartite graph with a perfect matching. Consider the $n \times n$ matrix $A^G(x_{11}, \ldots, x_{nn})$ whose (i,j)-th entry is a variable x_{ij} if $(u_i, v_j) \in E$ and zero otherwise. Does there exist an integer assignment $i_{11}, \ldots, i_{nn}$ to $x_{11}, \ldots, x_{nn}$ such that $\det(A^G(i_{11}, \ldots, i_{nn})) \neq 0$?

Proof. Let $\{(u_i, v_{\pi(i)}) \mid 1 \leq i \leq n\}$ be a perfect matching where π is a permutation on $\{1, \ldots, n\}$. Then the monomial $\prod_{i=1}^{n} x_{i\,\pi(i)}$ has coefficient 1 or -1 in $\det(A^G(x_{11}, \ldots, x_{nn}))$ and no other monomials contain all those variables $x_{i\,\pi(i)}$ for $1 \leq i \leq n$. Hence, by setting $x_{i\,\pi(i)}$ to 1 for $1 \leq i \leq n$ and all other variables to zero, the determinant will be ± 1. $\square$

Problem 3 (20 points)**.** For $c \in [\,0, 1\,]$, let $P(c)$ be the following statement:

> There exists a randomized polynomial-time algorithm outputting "Hamiltonian" with probability at least c when its input is a Hamiltonian graph, and "Not Hamiltonian" with probability 1 otherwise.

Show that $P(3/5)$ implies $P(3/4)$.

Proof. Assume the truth of $P(3/5)$. Consider the algorithm M which determines whether a given graph G is Hamiltonian by repeating the algorithm witnessing the truth of $P(3/5)$ for 100 times using independent random coin tosses and outputting "Hamiltonian" (resp., "Not Hamiltonian") if any (resp., none) of the 100 executions outputs "Hamiltonian." Given any Hamiltonian graph G, the probability that M outputs "Hamiltonian" is at least $1 - (1 - 3/5)^{100} > 3/4$. □

Problem 4 (20 points)**.** Let M be a polynomial-time Turing machine that, given as input an odd prime p, a primitive root g of p and $-g^x \bmod p$ for an unknown x, finds $x \bmod (p-1)$. Show how to break the discrete logarithm in polynomial time. That is, given an odd prime p, a primitive root g of p and $g^x \bmod p$ for an unknown x, show how to find $x \bmod (p-1)$ in time polynomial in the length of the inputs. (Hint: You may want to consider $g^{(p-1)/2} \bmod p$.)

Proof. Compute $-g^x \bmod p$ and feed M with p, g and $-g^x \bmod p$ to obtain $x \bmod p - 1$. □

Problem 5 (20 points)**.** Does PRIMES belong to IP? Briefly justify your answer.

Proof. We have BPP $\subseteq$ IP because a verifier can neglect all messages of the prover. We have also shown PRIMES $\in$ BPP in class. Therefore, PRIMES $\in$ IP. □

Problem 6 (20 points)**.** Prove that INDEPENDENT SET is NP-hard. You may assume the NP-completeness of CLIQUE or any other problem shown to be NP-complete in class.

Proof. We describe a reduction from CLIQUE to INDEPENDENT SET. Given a graph G and a number k as input, the reduction outputs the complement of G and k. Clearly, G has a clique of size k if and only if its complement has an independent set of size k. It is also clear that the reduction runs in logarithmic space. □