CS 351: Problem Set 5
(assigned 10/31/2002;   due 11/14/2002)

Note that you have *two* weeks for this homework, so it must be hard. I'll wait a week and post additional hints, depending on the volume of questions I receive—the more questions that show you're trying to solve the problems, the more hints you're likely to receive.

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.

(1a) In class, we showed an $O(\log n)$ communication protocol for the equality problem (with error at most $1/n$) using residues modulo a randomly chosen prime. Describe another $O(\log n)$ protocol for the same problem using the following idea. Alice and Bob interpret their inputs as degree $n-1$ polynomials with 0–1 coefficients, and compare the values of the polynomials on a randomly chosen point of a suitably large finite field.

(1b) Describe a randomized *simultaneous protocol* with $o(n)$ communication (and error at most 1/3, say) for the equality problem. A simultaneous protocol is one where Alice and Bob simultaneously write one message each on the blackboard (that is, Alice's message is written without knowledge of Bob's message, and similarly for Bob). (*Hint.* First ask yourself why the protocol of (1a) doesn't work. Then suitably modify the protocol of (1a) to do multiple evaluations so that with high probabilty, Alice and Bob evaluate their polynomials on a common point.)

(2) Show that if NP $\subseteq$ BPP, then NP = RP. (*Hint.* Use the reduction from search to decision to construct satisfying assignments; you'll need some amplification too.)

(3) A *hitting set* is a family of subsets $\{H_{n,s} \mid n, s \in \mathbf{N}, n \leq s\}$ with the following properties:

(a) There exists $a > 0$ such that for all n, s, $H_{n,s} \subseteq \{0,1\}^n$, $|H_n| \leq s^a$;

(b) There exists $b > 0$ and an algorithm A such that for all n, s, A, on input (n, s), produces the set $H_{n,s}$ within time s^b.

(c) For every n and s, for every n-input circuit C of size at most s such that $|\{x \mid C(x) = 1\}| \geq (2/3)2^n$, there exists $x \in H_{n,s}$ such that $C(x) = 1$.

Property (c) is the "hitting" property—the idea is that $H_{n,s}$ "hits," or intersects, $C^{-1}(1)$ for every n-input circuit C of size s that accepts a large fraction of all inputs. Properties (a) and (b) address the efficiency or constructibility of the hitting set.

(3a) The easy part: show that if a hitting set exists, then RP = P.

(3b) The harder part: show that if a hitting set exists, then BPP = P. (*Hint.* Revisit the proof of BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$, and note that the "existence" proofs are really "abundance" proofs. Use this idea to define a set A and build a one-sided error randomized algorithm (RP type) for L that, on input x, tosses a set of coins y, and then needs to decide if (x, y) belongs to A. Then show that the instance (x, y) of A can be solved by a one-sided error randomized algorithm (again, RP type, but note that I am not claiming that $A \in$ RP). Use the hitting set to derandomize each step in turn.

(4) Let #SAT denote the problem of counting the number of assignments that satisfy a given CNF Boolean formula. We know that #SAT is hard for #P (and, by Toda's theorem, is as hard as the entire polynomial-time hierarchy). Nevertheless, there is an approximation algorithm for #SAT (and more generally, for any #P counting problem) that runs in polynomial time using "oracle" access to a language in Σ_2^p. Prove this. That is, define a language L in Σ_2^p and construct an algorithm A that takes a Boolean formula φ as input, runs in polynomial time using polynomially many "subroutine calls" to L, and outputs an integer s such that $N_\varphi / c \leq s \leq c N_\varphi$, where N_φ denotes the number of assignments that satisfy φ, and $c > 1$ is a fixed constant independent of φ. (Note: in estimating the running time of A, we assume that the subroutine calls are free, that is, queries to L are answered in unit time step. In other words, A is a polynomial-time reduction from approximate computation of #SAT to L.)

Hint. The idea is that if $N_\varphi \approx 2^k$, and if h is a pairwise independent hash function from assignments to φ into $\{0,1\}^\ell$, then the element 0^ℓ is likely to have a pre-image under h if k is somewhat larger than ℓ, and is unlikely to have a pre-image under h if k is somewhat smaller than ℓ. To estimate k, try hash functions with range $\{0,1\}^\ell$ for $\ell = 0, 1, \ldots, n$, where n is the number of variables in φ. Then suitably replace randomness by quantifiers, as in the proof of BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$.

(5) Show that SAT has a proof system Π with the following properties:

(a) *Efficient verification.*

The verifier for Π is a randomized logspace algorithm that has one-way access to the proof tape (and one-way access to its random tape);

(b) *Perfect completeness.*

If φ is satisfiable, there is a proof that the verifier accepts with probability 1;

(c) *Probabilistic soundness.*

If φ is not satisfiable, then every alleged proof is rejected by the verifier with probability at least 2/3.