COMS 4236 – Test 3 Practice Problems

(1) Show that for every $L \subseteq \{0,1\}^*$, if $L \in \mathsf{RP}$, then $L \in \mathsf{P/poly}$ (that is, L has polysize circuits) in three steps.

(a) Since $L \in \mathsf{RP}$, there is a deterministic polynomial-time machine $M(x,r)$ and a polynomial p such that for every $x \in \{0,1\}^n$,

$$x \in L \implies \Pr_{r \in \{0,1\}^{p(n)}}[\, M(x,r) = 1 \,] \geq \tfrac{1}{2},$$
$$x \notin L \implies \Pr_{r \in \{0,1\}^{p(n)}}[\, M(x,r) = 1 \,] = 0.$$

Choose t strings $r_1, \ldots, r_t \in \{0,1\}^{p(n)}$ independently and uniformly at random, and accept x if and only if $M(x, r_i) = 1$ for at least one i. Show that an input $x \notin L$ is never accepted, and that an input $x \in L$ is rejected with probability at most 2^{-t}.

(b) Set $t = n+1$. Show that there exist *fixed* strings $r_1, \ldots, r_t$ such that for every $x \in \{0,1\}^n$,

$$x \in L \iff M(x, r_i) = 1 \text{ for some } i \in \{1, \ldots, t\}.$$

In fact, show that at least half of all choices of $(r_1, \ldots, r_t)$ have this property.

Hint: Which inputs can a choice of $(r_1, \ldots, r_t)$ get wrong? How many such inputs are there at most?

(c) Now using (b), show that $L \in \mathsf{P/poly}$. That is, describe a polysized circuit family C_n, $n > 0$ that decides L on inputs of length n. What is hardwired into it, and what is its size in terms of the running time of M?

(d) Now show how you can adapt this argument to prove the stronger statement that for every $L \in \mathsf{BPP}$, $L \in \mathsf{P/poly}$.

Hint: Two things change: the rule for combining the t runs, and the bound on the probability that the combined answer is wrong. You may use the following Chernoff bound: if $X_1, \ldots, X_t$ are independent 0/1 random variables, each with mean at least 2/3, then $\Pr[\sum_i X_i \leq t/2] \leq e^{-t/18}$.

(2) Prove that if SAT $\in \mathsf{BPP}$, then SAT $\in \mathsf{RP}$. Your algorithm should use the BPP decider to construct an assignment and should never accept an unsatisfiable formula. Bound the error and running time, including the cost of amplification.

(3) You may assume the following consequence of the PCP theorem: for some fixed constant $0 < \gamma < 1$, there is a polynomial-time map from a SAT instance x to a 3-CNF ψ_x with $m \geq 1$ clauses such that

$$x \in \text{SAT} \quad : \text{OPT}(\psi_x) = m,$$
$$x \notin \text{SAT} \quad : \text{OPT}(\psi_x) \leq (1-\gamma)m.$$

(a) Show that a deterministic polynomial-time algorithm always outputting an assignment of value at least $(1 - \gamma/2)\text{OPT}$ would imply $\mathsf{P} = \mathsf{NP}$.

(b) Suppose the approximation algorithm is randomized and produces such an assignment with probability at least 2/3 on every input. It always outputs an assignment, but may perform badly on unsuccessful runs. Prove that the existence of such an algorithm would imply $\mathsf{NP} = \mathsf{RP}$, not merely $\mathsf{NP} \subseteq \mathsf{BPP}$.

(4) Recall the standard reduction from 3SAT to Independent Set: given a 3-CNF ψ with m clauses, the graph G_ψ has one vertex for each occurrence of a literal in a clause, with an edge between two vertices if they belong to the same clause, or if one is an occurrence of a variable and the other an occurrence of its negation.

(a) Show that the size of a maximum independent set in G_ψ is exactly $\mathrm{OPT}(\psi)$, the maximum number of clauses of ψ that can be satisfied simultaneously.

(b) Using the gap from Problem (3), show that a polynomial-time algorithm that on every graph G outputs an independent set of size at least $(1-\gamma/2)$ times the maximum would imply $\mathsf{P} = \mathsf{NP}$.

(5) In the standard definition of PCP, $\mathsf{PCP}(r, q)$, the verifier uses r random coins, and makes q queries to the prover's string Π. What if instead the verifier is allowed to make q adaptive queries, meaning that the verifier first queries the first position of Π, and then based on the value returned, they then can decide on the second position to query, etc. Prove that any language L that has a PCP verifier using r random coins, and q adaptive queries also has a standard (i.e., non-adaptive) verifier using r coins and 2^q queries.

(6) Let y and m be integers with $\gcd(y, m) = 1$. We say that y is a quadratic residue mod m if there exists an integer x such that $y \equiv x^2 \pmod{m}$. Give an IP (interactive protocol) for the following language:

$$\text{NonResidue} = \{(y, m) \mid \gcd(y, m) = 1 \text{ and } y \text{ is not a quadratic residue mod } m\}.$$

(7) Explain in one or two sentences why in the proof that $\#\mathrm{SAT} \in \mathsf{IP}$, we need the prime p used in the protocol to be larger than 2^n.