COMS 4236: Introduction to Computational Complexity, Spring 2012

Problem Set 5, due Wednesday April 18, in class

Problem 1. Consider the following problem, ALGEBRAIC IDENTITY:
Input: Two algebraic expressions E_1, E_2 involving a set of variables $x_1, x_2, ..., x_m$ and integer constants, with operations of addition, subtraction and multiplication.
Question: Is $E_1(x_1, \ldots, x_m) = E_2(x_1, \ldots, x_m)$ for all tuples of real values $x_1, x_2, ..., x_m$?
For example, the answer is Yes, for $E_1 = (x_1 - x_2) \cdot (x_1 - x_2) + 4 \cdot x_1 \cdot x_2$ and $E_2 = (x_1 + x_2) \cdot (x_1 + x_2)$.

1. One way to solve the problem is the standard school method: take each of the two expressions, eliminate all the parentheses multiplying the terms as needed, then collect the like terms, and compare the two expressions to see if they are identical. Show that this is not a polynomial time algorithm. In particular, give a family of examples on which this algorithm takes exponential time.

2. Give a randomized polynomial time algorithm for this problem. Does your algorithm place this problem in RP, coRP, or BPP?

Problem 2. Show that the following are equivalent for a language L.

1. L $\in$ ZPP=RP$\cap$coRP.

2. There is a probabilistic Turing Machine M that runs in polynomial time and which returns either "Yes" or "No" or "?", with the property that M never makes a mistake and always answers Yes or No with probability at least 1/2. That is, for every input string *x*:
$\forall$x $\in L \Rightarrow$ Pr(Yes) $\geq$ ½ , Pr(No)=0, Pr(?) $\leq$ ½
$\forall$x $\notin L \Rightarrow$ Pr(Yes) =0, Pr(No) $\geq$½ , Pr(?) $\leq$ ½

3. There is a probabilistic Turing Machine M which runs in polynomial time and returns either "Yes" or "No" or "?" such that
$\forall$x $\in L \Rightarrow$ Pr(Yes) $\geq 1\text{-}2^{-n}$, Pr(No)=0, Pr(?) $\leq 2^{-n}$
$\forall$x $\notin L \Rightarrow$ Pr(Yes) =0, Pr(No) $\geq 1\text{-}2^{-n}$, Pr(?) $\leq 2^{-n}$

Problem 3. Prove that a language *L* is in ZPP if and only if it is decided by a probabilistic Turing Machine *N* that runs in *expected* polynomial time. That is, for every input string *x*, (1) every computation of *N* terminates with the correct answer Yes or No (there is no "?"

answer), and (2) the expected running time of N on input *x*, $\overline{T}_N(x)$, is bounded by a polynomial *p*(|*x*|). The expected running time of N on input *x* is by definition $\overline{T}_N(x) = \sum \{\Pr(comp) \cdot t(comp) \mid comp \text{ a computation of } N \text{ on input } x\}$, where Pr(*comp*) is the probability of computation *comp*, *t*(*comp*) is the running time of *comp*, and the summation is over all possible computations *comp* of the PTM *N* on input *x*.
(*Hint:* Use the equivalent conditions 2 and 3 of Problem 2 for ZPP. For the one direction, repeat enough times the algorithm of a PTM M if necessary to reduce sufficiently the probability of "?", and then switch to an exhaustive algorithm to guarantee termination. For the other direction, truncate the computation of N after a suitable amount of time.)

Problem 4. Recall that a *Cook reduction* (or *polynomial-time Turing reduction*) from a language *A* to a language *B* is a polynomial-time deterministic oracle machine M^B that decides *A* using an oracle for *B*, i.e. $A = L(M^B)$. We use the notation $A \le_p^T B$ to denote that there is a Cook reduction from *A* to *B*.

1. Show that Cook reductions compose, i.e. if $A \le_p^T B$ and $B \le_p^T C$ then $A \le_p^T C$.

2. We say that a complexity class $\mathfrak{C}$ is *closed* under Cook reductions if, whenever *A* reduces to *B* and *B* is in $\mathfrak{C}$, then also *A* is in $\mathfrak{C}$. Assume that all conjectured inequalities between complexity classes are true (eg. P≠NP, NP≠coNP etc.)
Which of the classes, P, NP, Δ_2, PH, PSPACE are closed under Cook reductions?
Give *very brief* justifications.

Problem 5. Show that $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{coNP}} = \mathrm{NP}$.