ITCS:CCT09 : Computational Complexity Theory May 26, 2009

Problem Set # 2

Topic: Lectures 9-$ *Due on:* **Jun 10, 2009**

Problem 1

(Slice Functions)

A function is called a *slice function* if for some positive integer k,

$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains more than } k \text{ 1's} \\ 0 & \text{if } x \text{ contains less than } k \text{ 1's} \end{cases}$$

and there are no restrictions when the number of 1s is exactly k.

1. Notice that slice functions are monotone. Show that a general circuit computing a slice function can be converted into a monotone circuit at the expense of only a polynomial blow up in the size of the circuit.

2. Consider the following problem : Given a graph G with $2n$ vertices, test which of the following is true (1) G has a clique of size n or (2) G has at least $\lceil \frac{n(2n-1)}{2} \rceil + 1$ edges. This can be shown[1] to be NP-coomplete. Define a sequence of slice functions which represents this NP-complete problem.

Problem 2

(Super-linear Circuit Lower Bounds)

As we saw in the class, it is unknown (and is unlikely because it implies a collapse of the PH) whether NP has polynomial sized circuits. That whether or not $\mathsf{NP} \subseteq \mathsf{P/poly}$. On the lower bounds front, in fact, it is unknown whether there is an explicit function in NP that does not have $O(n)$ sized circuits. In this exercise, we will show something weaker.

- Show that there is a function in the PH which does not have $O(n)$ sized circuits. (Hint: how does one describe a boolean function which can be computed by a circuit of size k, but cannot be computed by any circuit of size k' where $k' < k$? think of k as $O(n^2)$ and k' as $O(n)$.)

- Improve this to show that there is a function in $\Sigma_2^p \cap \Pi_2^p$ which does not have $O(n)$ sized circuits.

[1] You need not prove it in your solutions.

Problem 3

(Structural results about randomization)

Here we will show some basic structural results about the randomized classes, which we skipped in class.

1. Recall the definition of ZPP as the languages accepted by randomized Turing machines (that runs in expected polynomial time) which never makes an error (in terms of acceptance), but it can answer "I dont know". Show that $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.
2. A class A is said to be *low* for a class B if it is true that $A^B = A$. Show that BPP is *low* for itself.
3. Consider a randomized logspace machine M. Show that we can compute in polynomial time the probability that the machine reaches the accepting configuration (Use matrix multplication !). Thus show that $\mathsf{BPL} \subseteq \mathsf{P}$.

Problem 4

(Query Length and Randomness)

A polynomial-time oracle DTM is called a *log-query machine* if for any input x of length and for any oracle A, it makes at most $c \log n$ queries, where c is a constant. Let Θ_2^{P} denote the class of sets computable by polynomial-time log-query oracle machines with an oracle in NP. Prove that $\Theta_2^{\mathsf{P}} \subseteq \mathsf{PP}$.

Problem 5

(Can it be the case that $\mathsf{3SAT} \leq_r \overline{\mathsf{3SAT}}$)?

A non-deterministic circuit has two inputs x and y (size measured in terms of $|x|$). We say that it accepts a string x if and only if there exists a y such that $C(x, y) = 1$. Let $\mathsf{NP/poly}$ be the languages that are decided by polynomial sized non-deterministic circuits.

1. We defined the BP operator in class. Show that $\mathsf{BP.NP} \subseteq \mathsf{NP/poly}$.
2. Show that if $\overline{\mathsf{3SAT}} \in \mathsf{BP.NP}$ then PH collapses to Σ_3^p (Use ideas similar to Karp-Lipton theorem that we already saw). Thus conclude that it is unlikely that $\mathsf{3SAT} \leq_r \overline{\mathsf{3SAT}}$.