Complexity Theory

Homework Sheet 5

(Turn in before the lecure of Wednesday 7 May.)

28 April, 2014

Exercise 1. Show that given any set $A \in \mathrm{NP}^X$, we can, in $\mathrm{P}^{\mathrm{NP}^X}$ (Hint: not necessarily in P^A), find the lexicographically least string of a given size that belongs to A. More precisely, show that the function

$$g(m) = \min A \cap \{0,1\}^m$$

can be computed in time polynomial in m when given oracle access to a set in NP^X.

Exercise 2. Let $k > 0$ be a fixed arbitrary natural number. Let $f : \{1, \ldots, n^{k+1}\} \to \{0,1\}$ denote boolean functions, and say that a circuit of size s computes f if, when given as input a number i from 1 to n^{k+1} encoded using $\log n^{k+1}$ bits, it outputs the bit $f(i)$.

(a) Prove that there is such a function f which doesn't have circuits of size n^k. (Hint: use counting, also see Theorem 6.21 in the book)

(b) Show that, given as input the n^{k+1} bits $f(1), \ldots, f(n^{k+1})$ defining some such function f, we may decide in coNP (and hence in Σ_2^p) whether f is **not** computable by any circuit of size n^k.

(c) Show that there is a set in $\mathrm{P}^{\Sigma_2^p}$ which does not have circuits of fixed-polynomial size n^k (Hint: Use Exercise 1). Why is this not the same as showing a separation from P/poly?

(d) Use line (c) together with the Karp-Lipton theorem to show that there is actually a set in $\Sigma_2^p \cap \Pi_2^p$ which does not have circuits of fixed-polynomial size n^k. (Hint: if SAT doesn't have polynomial-sized circuits, then you're done, why?)

Definition 1. Let $\mathbb{F}$ be some field. A function $f : \{0,1\}^n \to \{0,1\}$, is *represented* by a multivariate polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$ if

$$p(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$$

for every $x_1 \ldots x_n \in \{0,1\}^n$. (For the left side of the equation we take 0 and 1 to be the additive and multiplicative identities of the field $\mathbb{F}$, respectively.)

The *degree* of f over $\mathbb{F}$, written $\deg_{\mathbb{F}}(f)$, is the smallest degree of any polynomial representing f.

Exercise 3. Show that every boolean function f can be represented by a multilinear polynomial p, and that such a p is unique.

Exercise 4. Show, using polynomials, that the NAND function can not be computed by a circuit consisting only of XOR and NOT gates.

(Hint: What is the degree of XOR over fields you could consider?)