CS-621 Theory Gems December 21, 2012

Problem Set 5

Lecturer: Aleksander Mądry *Due: January 21, 2013*

Problem 1.

(a) Prove that if one-way functions exist then there exists a one-way function f such that $|f(x)| = |x|$ for all x, i.e., the length of the output of f is always equal to the length of its input, where $|x|$ denotes the length (number of bits) of the string x.

(b) Prove that if one-way functions exist then there exists a one-way function f and a constant n_0 such that $f(x)$ can be computed in $|x|^2$ time for all x of length at least n_0.

Problem 2. In this problem, we want to show that the assumption that one-way functions exist is stronger than the one that $P \neq NP$.

(a) Prove that if $P = NP$ then one-way *permutations* do not exist.

Hint: Recall that if $P = NP$ then for any polynomial-time computable Boolean function $g : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ with $m = O(n^c)$, for some constant c, there exists a polynomial-time algorithm A_g that on any input $x \in \{0,1\}^n$, decides if there exists a $y \in \{0,1\}^m$ such that $g(x, y) = 1$.

(b) Extend the ideas form (a) to prove that if $P = NP$ then one-way *functions* do not exist.

Note: If you get stuck on this problem, email the lecturer to get a hint.

Problem 3. For a given function f and a constant $c \in \mathbb{N}$, let $g_{f,c}$ be a function defined as $g_{f,c}(x) := f^{|x|^c}(x)$. (Here, $f^k(x)$ denotes $f(f(\ldots f(x)))$, where f is applied k times and $|x|$, again, denotes the length of the string x.)

(a) Let f be a one-way permutation, show that for any constant $c \in \mathbb{N}$ the function $g_{f,c}$ is also a one-way permutation.

(b) Assuming that one-way functions exist, show that there exists a one-way function f' and a constant $c \in \mathbb{N}$ such that function $g_{f',c'}$ is *not* a one-way function.

Note: If you get stuck on this problem, email the lecturer to get a hint.

Note 2: Recall that in the lecture we mentioned that if f is a one-way permutation then a function that maps some $x, r \in \{0,1\}^n$ to $r, f^l(x) \odot r, f^{l-1}(x) \odot r, \ldots, f(x) \odot r$ with $l = n^c$, for some constant c, is a pseudo-random generator with stretch $\ell(2n) = n + n^c$.[1]
So, in this problem we discovered one (of many) reasons why this simple construction does not work when f is only a one-way function and not a one-way permutation.

Problem 4 (Extra credit). Recall the *Levin's one-way function* f_L that was defined in the lecture as follows. Given input $x \in \{0,1\}^n$ of length n, x is broken into $\log n$ pieces $x_1, \ldots, x_{\log n}$ of length $\frac{n}{\log n}$ each[2] (i.e., $|x_i| = \frac{n}{\log n}$ for all i, and $x = x_1, \ldots, x_{\log n}$) and

$$f_L(x) := M_1^{n^2}(x_1), \ldots, M_{\log n}^{n^2}(x_{\log n}).$$

(Here, M_i denotes the i-th Turing machine according to some canonical (and efficient) enumeration of all Turing machines[3] and $M_i^t(x)$ is: the output of the Turing machine M_i on input x, if M_i stops after at most t steps on input x; 0, otherwise.)

Prove universality of f_L, i.e., show that if one-way functions exist then f_L is a one-way function.

[1] Here, $\odot$ denotes inner product modulo 2.

[2] For simplicity, we assume that both $\log n$ and $n/\log n$ are integer numbers.

[3] Alternatively, just consider the lexicographical ordering $0 \preceq 1 \preceq 00 \preceq 01 \preceq \ldots$ of binary strings (of positive length) and think of M_i as a computer program obtained by running, say C compiler, on lexicographically i-th string. (Of course, most of the time, this string will not correspond to a valid C program and thus the compiler will fail, but in such cases we can define M_i to be just a dummy program that simply prints 0 and terminates.)