Recent advances in Complexity CIS 6930/CIS 4930 October 29, 2002

Lecture 19

Lecturer: Dr. Meera Sitharam Scribe: Hongyu Guo

Erwin's Talk on Natural Proofs (Part II)

In the last lecture, we introduced the main theorem. We are going to prove it in this lecture.

Theorem 1 *There is no lower bound proof which is P/poly-natural against P/poly, unless* $H(G_k) \leq 2^{k^{O(1)}}$ *for every pseudo-random generator* $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ *in P/poly.*

Proof Idea: We will use proof by contradiction. We assume that if there is a lower bound proof which is P/poly-natural against P/poly, then we show that we can construct a polynomial algorithm to discriminate the output of a pseudo random generator $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ against the true random $2k$ bit strings.

We do this in two steps. First we construct a mechanism that can distinguish a pseudo random *function* (i.e, a particular collection of strings of length 2^n) from a true random function (a random string of length 2^n), and in step 2 we label the pseudo random functions with $2k$ bit pseudo random *strings* and we label true random functions with $2k$ bit true random strings and using the mechanism in step 1, we can distinguish the $2k$ bit pseudo random strings from the $2k$ bit true random strings in polynomial time. This contradicts the assumption, which is not proved but is common belief, that the hardness of all the polynomial time pseudo random generators is at least exponential.

Proof: For the sake of contradiction, supose that such a lower bound proof exists and C_n is the accociated *P/poly*-natural combinatorial property. Let $C_n^* \subseteq C_n$ satisfy the constructivity and largeness conditions. W.l.o.g. we may assume from the very beginning that $C_n^* = C_n$.

Let $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ be a polynomial time computable pseudo random generator, and $\epsilon > 0$ be an arbitrary constant. Set $n = \lceil k^\epsilon \rceil$.

Step 1. We use $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ for constructing a pseudo random function generator $f : \{0,1\}^k \longrightarrow F_n$ in the following way: Let $G_0, G_1 : \{0,1\}^k \longrightarrow \{0,1\}^k$ be the maps that take the first and the last k bits of G_k, respectively. For each $x \in \{0,1\}^k$, it is mapped by f to $\phi = f(x) \in F_n$. And each $y \in \{0,1\}^n$ is mapped by $\phi = f(x)$ to $\phi(y) = f(x)(y) \in \{0,1\}$. We define the map f by giving a rule of assigning a function $\phi \in F_n$ for each $x \in \{0,1\}^k$ and we define ϕ by giving a rule of assigning a value in $\{0,1\}$ to each $y \in \{0,1\}^n$. In order to define ϕ, which is $f(x)$, we first define an auxiliary map $GA : \{0,1\}^n \times \{0,1\}^k \longrightarrow \{0,1\}^k$

such that $GA(y,x) = G_{y_n} \circ G_{y_{n-1}} \circ \cdots \circ G_{y_1}(x)$. Or we can write the indexed or parameterized version of definition as, for each $y \in \{0,1\}^n$ (y is an index or parameter), $G_y : \{0,1\}^k \longrightarrow \{0,1\}^k$ by $G_y = G_{y_n} \circ G_{y_{n-1}} \circ \cdots \circ G_{y_1}$, where $y_i =$ 0 or 1 is the $i-th$ bit of y from the left. Finally we define $\phi(y) = f(x)(y)$ to be the first bit of $GA(y,x) = G_y(x)$.

Note that $f(x)(y)$ is computable by poly-size circuits, hence (from the definition of a natural proof against P/poly) the function $f(x) \in F_n$ is not in C_n for any fixed $x \in \{0,1\}^k$ and any suffciently large k. In other words, C_n has empty intersection with $\{f(x)|x \in \{0,1\}^k\}$ and this disjointness implies that C_n provides a statistical test for $f(x)$, with

$$|P[C_n(f_n) = 1] - P[C_n(f(x)) = 1]| \geq 2^{-O(n)}$$

This is because $P[C_n(f(x)) = 1] = 0$ and the largeness condition $P[C_n(f_n) = 1] \geq 2^{-O(n)}$. Note that this test is computable by circuits of size $2^{O(n)}$.

So far you have a way of distinguishing pseudorandom functions from true random functions. Now convert this to strings.

Step 2. We use the result in step 1 to construct a statistical test for strings. We construct a full binary tree T of height n. Each left edge is labled 0 and each right edge is labled 1. Each leaf represents an n-bit string in $\{0,1\}^n$ with the sequence of lables on the path from the root of the tree to the leaf. $G_y = G_{y_n} \circ G_{y_{n-1}} \circ \cdots \circ G_{y_1}$ is also represented by the path from the root to the leaf y. We lable the nodes with $1, 2, 3, ..., 2^{n-1}$ in such a way that if v_i is a son of v_j then $i < j$. Let T_i be the union of subtrees of T made by $\{v_1, ..., v_i\}$ along with all leaves. For a leaf y of T let $v_i(y)$ be the root of the subtree in T_i containing y. Let $G_{i,y} = G_{y_n} \circ \cdots \circ G_{y_n - h(i,y)+1}$, where $h(i,y)$ is the distance between $v_i(y)$ and y. $G_{i,y}$ is also represented by the path from $v_i(y)$ to y.

Finally define the random collection $f_{i,n}$ by letting $f_{i,n}(y)$ be the fist bit of $G_{i,y}(x_{v_i(y)})$, where x_v are taken from $\{0,1\}^k$ uniformly and independently for all roots v of trees from T_i.

Notice that basically when i increases, the size of the random collection decreases. The size of the random collection of $f_{i,n}$ is $2^{(2^n - i)k}$. The size of the random collection at the root $f_{2^n-1,n}$ is the smallest, which is 2^k.

We denote $f_{0,n}$ as any random function f_n and we know $f_{2^n-1,n}$ is $f(x)$. Because all the intermedium terms cancel each other, except the first and the last term, and using the result in step 1, we have

$$\begin{aligned}
&\textstyle\sum_{i=0}^{i=n} |P[C_n(f_{i,n}) = 1] - P[C_n(f_{i+1,n}) = 1]| \\
&\geq |\textstyle\sum_{i=0}^{i=n} P[C_n(f_{i,n}) = 1] - P[C_n(f_{i+1,n}) = 1]| \\
&= |P[C_n(f_n) = 1] - P[C_n(f(x)) = 1]| \\
&\geq 2^{-O(n)}
\end{aligned}$$

There must exist a term $|P[C_n(f_{i,n}) = 1] - P[C_n(f_{i+1,n}) = 1]| \geq \frac{1}{n} 2^{-O(n)}$, which is the same as $|P[C_n(f_{i,n}) = 1] - P[C_n(f_{i+1,n}) = 1]| \geq 2^{-O(n)}$.

T_{i+1} is the union of subtrees. Suppose the roots of these subtrees are v_{i+1} and $r_1, r_2, ..., r_m$. Denote these subtrees by their roots, $R_{v_{i+1}}, R_{r_1}, R_{r_2}, ..., R_{r_m}$.

$T_{i+1} = R_{v_{i+1}} \bigcup \{\bigcup_{q \in M} R_{r_q}\}$,

where $M = \{1, 2, ..., m\}$. Let v', v'' be the two sons of v_{i+1}. Then we know T_i is the union of R'_v, R''_v, and $R_{r_1}, R_{r_2}, ..., R_{r_m}$.

$T_i = R_{v'} \bigcup R_{v''} \bigcup \{\bigcup_{q \in M} R_{r_q}\}$.

In T_{i+1} and T_i, all other subtrees are the same except T_{i+1} has $R_{v_{i+1}}$ while T_i has R'_v and R''_v.

The collection f_{i+1} can be expressed as the union

$f_{i+1} = \bigcup_{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}} f_{i+1}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}}$,

where $x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}$ are a set of fixed assignment of k-bit strings placed at the roots $v_{i+1}, r_1, r_2, ..., r_m$.

Similarly, the collection f_i can be expressed as the union

$f_i = \bigcup_{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}} f_i^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}}$.

We obtained from the previous argument that

$|P[C_n(f_{i,n}) = 1] - P[C_n(f_{i+1,n}) = 1]| \geq 2^{-O(n)}$.

The probability space for $P[C_n(f_{i+1,n}) = 1]$ is $\{0,1\}^{(m+1)k}$ and the probability space for $P[C_n(f_{i,n}) = 1]$ is $\{0,1\}^{(m+2)k}$. The above inequality translates to

$\frac{1}{2^{(m+2)k}} \sum_{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}} C_n(f_{i,n}^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}})$

$-\frac{1}{2^{(m+1)k}} \sum_{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}} C_n(f_{i+1,n}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}})$

$\geq 2^{-O(n)}$.

This is equivalent to

$\frac{1}{2^{mk}} \sum_{x_{r_1}, x_{r_2}, ..., x_{r_m}} \{\frac{1}{2^{2k}} \sum_{x_{v'}, x_{v''}} C_n(f_{i,n}^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) - \frac{1}{2^k} \sum_{x_{i+1}} C_n(f_{i+1,n}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}})\}$

$\geq 2^{-O(n)}$.

If the mean of 2^{mk} terms is $\geq 2^{-O(n)}$, then there must exit one term that is $\geq 2^{-O(n)}$. Namely there exists a fixed set of assignment of $x_{r_1}, x_{r_2}, ..., x_{r_m}$ such that

$\frac{1}{2^{2k}} \sum_{x_{v'}, x_{v''}} C_n(f_{i,n}^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) - \frac{1}{2^k} \sum_{x_{i+1}} C_n(f_{i+1,n}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}})$

$\geq 2^{-O(n)}$.

This is a probability argument that we can carefully choose a fixed set x_v for all roots $r_1, r_2, ..., r_m$ of subtrees in T_{i+1} other than v_{i+1} so that the bias $2^{-O(n)}$ is preserved.

$|P_{x_{v'}, x_{v''}}[C_n(f_{i,n}^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) = 1] - P_{x_{v_{i+1}}}[C_n(f_{i+1,n}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) = 1]|$

$\geq 2^{-O(n)}$.

This is the bias preservation lemma and we have just proved it.

The probability space for $P_{x_{v'}, x_{v''}}[C_n(f_{i,n}^{x_{v'}, x_{v''}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) = 1]$ is 2 k-bit strings placed at $x_{v'}$ and $x_{v''}$. The probability space for $P_{x_{v_{i+1}}}[C_n(f_{i+1,n}^{x_{v_{i+1}}, x_{r_1}, x_{r_2}, ..., x_{r_m}}) = 1]$ is k-bit strings placed at $x_{v_{i+1}}$.

Given all other x_v fixed, the collection of functions in f_i is produced by placing random strings x of length k at v' and v'' while the collection of functions in f_{i+1} is produced by placing random strings x of length k at v_{i+1}. The colleciton of functions in f_{i+1} can be also viewed as procuded by placing $G_0(x)$ at v' and placing $G_1(x)$ at v''. So f_i is produced by the concatenation of two k bit random strings while f_{i+1} is produced by the output of the pseudo random generator G_k, which expands k bit random strings to $2k$ bit pseudo randoms strings. From the inequality we just proved, we can distinguish between $G_{x_{v_i+1}}$ and (x'_v, x''_v). Thus, $H(G_k) \leq 2^{O(n)} \leq 2^{O(k^\epsilon)}$. As ϵ is arbitrary, the result follows. ■