Recent advances in Complexity CIS 6930/CIS 4930		November 12, 2002
	Lecture 23	
Lecturer: Dr. Meera Sitharam		Scribe: Srijit Kamath

Recall that we had split the proof of (2)⇒(3) of the main theorem into two subparts (Lecture 22). In this lecture we present proofs of these subparts.

Exercise 1 *Show that subparts 1 and 2 imply (2)⇒(3) of the main theorem.*

1 Proof of Subpart 1

Lemma 1 *Let m, n, l be integers. Let $f : \{0,1\}^m \longrightarrow \{0,1\}$, $H_f(m) \geq n^2$, and let A be a Boolean $n \times l$ matrix which is a $(logn, m)$ design. Then $G : \{0,1\}^l \longrightarrow \{0,1\}^n$ defined from A as $G(x) = f(x_1), \ldots, f(x_n)$ is a pseudo random generator satisfying*

$$|P[C(y) = 1] - P[C(G(x)) = 1]| \leq \frac{1}{n},$$

for all circuits C of size n.

Proof. Before presenting the proof note that in the main theorem we will apply the lemma for $l = m^2$ and $n = s(l^c)$.

The proof involves a series of bias preservations. The idea is to decompose an event into a bunch of events. If the bias holds for the original event, it also holds for atleast one event in the union. Assume that G does not satisfy the given condition, i.e.,

$$|P[C(y) = 1] - P[C(G(x)) = 1]| > \frac{1}{n}.$$

In other words the circuit C is able to distinguish between y and $G(x)$. We will arrive at a contradiction that f could not have assumed the hardness property as G was constructed from f. It is first shown that $\exists i$ such that C can predict $f(x_i)$ from $f(x_1), \ldots, f(x_{i-1})$. Since x_i is uncorrelated to $x_1, x_2, \ldots, x_{i-1}$, it follows that C could not have got information about x_i from those. So $f(x_i)$ has been independently computed from scratch contradicting the fact that f is hard. The details of the proof follow.

Define E_i to be a distribution on $\{0,1\}^n$, such that E_0 is uniform over $\{0,1\}_n$, E_n is the distribution $G(x)$, where x is uniform over $\{0,1\}^l$ and in general E_i is obtained by choosing $f(x_1), \ldots, f(x_i)$, where x is uniform and the remaining bits are uniform over $\{0,1\}^{n-i+1}$. Observe that the definitions of E_0

and E_n are consistent with $E_0 = E_i$, $i = 0$ and $E_n = E_i$, $i = n$. Also define $P_i = P[C(z) = 1]$ where z is chosen from E_i. We have $P_0 - P_n = \sum_i [P_{i-1} - P_i]$. This is a telescoping sum (refer to Henry's notes for more on this) and clearly $\exists i$ such that $|P_{i-1} - P_i| \geq 1/n^2$, i.e., C distinguishes well between two types of strings; strings having $i-1$ bits defined from applying f on the design and the rest random (E_{i-1}) and those having i bits defined from applying f on the design and the rest random (E_i).

Using this we build a circuit D that predicts $f(x_i)$ from $f(x_1), \ldots, f(x_{i-1})$. D takes as inputs the first $i-1$ bits, $z_1, z_2, \ldots, z_{i-1}$, of a string z in E_{i-1}, i.e., $f(x_1), \ldots, f(x_{i-1})$ for some $x \in \{0,1\}^l$. It will output a bit z_i, which is a good approximation of $f(x_i)$. D is a probabilistic circuit (at start) that chooses $n-i+1$ random bits $r_i, \ldots, r_n$, computes $C(z_1, \ldots, z_{i-1}, r_i, \ldots, r_n)$, and if the output is 1 it predicts $z_i = r_i$ and if the output is 0 it predicts $z_i = \bar{r_i}$. Using the same proof as Yao' s XOR lemma, $P[D(z_1, z_2, \ldots, z_{i-1}) = f(x_i)] - 1/2 \geq 1/n^2$. This bias is true for the probability taken over the collection of $r_i, \ldots, r_n$ and the $z_i, \ldots, z_n$. We can now claim that $\exists \{r_i, \ldots, r_n\}$ (fixed values for the random bits) such that the prediction works with the bias. Note that once $r_i, \ldots, r_n$ are fixed the circuit is no longer probabilistic. However it still works with the same bias when the probability is taken over $z_1, \ldots, z_{i-1}$. Also note that D has the same size as C. This completes the first part of what we are trying to do: prediction of $f(x_i)$ from $f(x_1), \ldots, f(x_{i-1})$.

Next we show that the prediction has not used $f(x_1), \ldots, f(x_i)$, since x_i is unrelated to $x_1, \ldots, x_{i-1}$, and so the prediction comes down to computing $f(x_i)$ directly from x. In order to do this we construct a circuit that uses D to compute $f(x_i)$ from x, while still preserving the size. Call this new circuit D'.

In constructing D' we make use of the fact that the restrictions $x_1, \ldots, x_{i-1}$ have a certain relationship to the restriction x_i. Each shares at most $logn$ bits of x with x_i. Without loss assume that $x_i = x^1, x^2, \ldots, x^m$, the first m bits of x. Since z_i does not depend on the other bits of x, we can rewrite the probability that D predicts z_i correctly as $P[D(z_1, z_2, \ldots, z_{i-1}) = z_i]$ (where x is chosen as random) over all possible choices of the bits $x^{m+1}, \ldots, x^l$ of the same probability over the distribution where only $x_1, \ldots, x_m$ are chosen at random i.e., $E_{i-1,x^1,\ldots,x^l} = \bigcup E^{c_{m+1},\ldots,c_l}_{i-1,x^1,\ldots,x^m}$, where the union is over all possible choices $c_{m+1}, \ldots, c_l$ of $x^{m+1}, \ldots, x^l$.

Applying the bias preservation property note that there exists some element of this union, i.e., some choice of $x^{m+1}, \ldots, x^l$, such that the bias on the probability is preserved. Once these $l-m$ bits are fixed, the probability is over the remaining m bits. Now $z_1, \ldots, z_{i-1}$, each depend only on $logn$ of the bits in x_i. This allows D' to incorporate the computation of f on $logn$ bits for each x_j. Recall that f is assumed by the Theorem to be computable in exponential time. There are $2^{logn} = n$ such values of f for each x_j that can actually be tabulated within D'. Each of these sets of values requires $O(n)$ space and there are $O(n)$ of such x_j resulting in a $O(n^2)$ additional size to the circuit D' (besides the size of the circuit C). The $f(x_j)$s obtained can now be fed into a copy of D in D' to give x_i as output. We have constructed the required circuit D', which contradicts hardness of f at m. ■

2 Proof of Subpart 2

Lemma 2 *For all integers $n, m, logn \leq m \leq n$, $\exists(logn, m)$ design that can be constructed by an algorithm in $DPACE(logn)$.*

Proof. The design has n rows of subsets of $\{0, 1, \ldots, l\}$ of size m with intersections of size $logn$. Without loss, assume that m is a prime power. Also let $l = m^2$. If m is not a prime power pick the smallest power of 2 greater than m. Note that this doubles m at most. Consider numbers in $\{0, 1, \ldots, l\}$ as pairs of elements in $GF(m)$, i.e., construct subsets of $\{< a, b > | a, b \in GF(m)\}$. Given a polynomial q on $GF(m)$, define a set $S_q = \{< a, q(a) > | a \in GF(m)\}$. We take sets of this form and q varies over polynomials of degree at most $logn$. Now the following facts can be verified:

(1) The size of each set is exactly m.

(2) Any two sets intersect in at most $logn$ points.

(3) There are atleast n different sets (the number of polynomials over $GF(m)$ of degree at most $logn$ is $m^{logn+1} \geq n$).

Note that the sets can be effectively constructed using simple arithmetic in $GF(m)$. Since m has length of $O(logn)$ bits, everything can be computed in $O(logn)$ space. ■