Computational Complexity Theory, Fall 2010 November 26

Lecture 22: Nisan-Wigderson PRG

Lecturer: Kristoffer Arnsfelt Hansen Scribe: Kristoffer Arnsfelt Hansen

1 Unpredictability implies pseudo randomness

Last lecture we stated the proof showing how to convert a circuit that can distinguish the output of a generator from uniformly random bits into a next bit predictor circuit. We now present the proof. It uses a very useful technique known as the "hybrid method".

Lemma 1. *Let $G : \{0,1\}^{l(m)} \to \{0,1\}^m$ be any function, and let C be a circuit of size S such that*

$$|\Pr[C(G(U_{l(m)})) = 1] - \Pr[C(U_m) = 1]| > \epsilon$$

Then there is $i \in \{1, \ldots, m\}$ and a circuit P of size S such that

$$\Pr[P(w_1 \ldots w_{i-1}) = w_i] > \frac{1}{2} + \frac{\epsilon}{m} \ ,$$

where the probability is over $w_1 \ldots w_n = G(y)$ for uniform random $y \in \{0,1\}^{l(m)}$.

Proof. Let $z_1 \ldots z_m$ be uniform random bits, and let y and w be at random chosen as above.

Our assumption is thus

$$|\Pr[C(w) = 1] - \Pr[C(z) = 1]| > \epsilon$$

Next consider *hybrid* distributions (between z and w) defined as follows:

$$\begin{aligned} D_0 &: z_1 z_2 z_3 \ldots z_m \\ D_1 &: w_1 z_2 z_3 \ldots z_m \\ D_2 &: w_1 w_2 z_3 \ldots z_m \\ &\vdots \\ D_{m-1} &: w_1 w_2 \ldots w_{m-1} z_m \\ D_m &: w_1 \ldots w_{m-1} w_m \end{aligned}$$

□

We can now write the above inequality as a telescoping sum, and use the triangle inequality.

$$\begin{aligned} \epsilon &< \left| \Pr_{x \sim D_m}[C(x) = 1] - \Pr_{x \sim D_0}[C(x) = 1] \right| \\ &= \left| \sum_{i=1}^{m} \Pr_{x \sim D_i}[C(x) = 1] - \Pr_{x \sim D_{i-1}}[C(x) = 1] \right| \\ &\leq \sum_{i=1}^{m} \left| \Pr_{x \sim D_i}[C(x) = 1] - \Pr_{x \sim D_{i-1}}[C(x) = 1] \right| \end{aligned}$$

Thus there is fixed $i \in \{1, \ldots, m\}$ such that

$$\frac{\epsilon}{m} < \left| \Pr_{x \sim D_i}[C(x) = 1] - \Pr_{x \sim D_{i-1}}[C(x) = 1] \right|$$

or using the definition of the distributions,

$$\frac{\epsilon}{m} < |\Pr[C(w_1 \ldots w_i z_{i+1} \ldots z_m) = 1] - \Pr[C(w_1 \ldots w_{i-1} z_i \ldots z_m) = 1]|$$

Now to get rid of the absolute value, let $a \in \{0, 1\}$ be such that

$$\frac{\epsilon}{m} < \Pr[C(w_1 \ldots w_i z_{i+1} \ldots z_m) = a] - \Pr[C(w_1 \ldots w_{i-1} z_i \ldots z_m) = a]$$

Now, by an averaging argument, we can fix the values of $z_{i+1}, \ldots, z_m$ to constants $\tilde{z}_{i+1}, \ldots, \tilde{z}_m$ preserving the advantage. Let $\tilde{z} = \tilde{z}_{i+1}, \ldots, \tilde{z}_m$. Then we have

$$\frac{\epsilon}{m} < \Pr[C(w_1 \ldots w_i \tilde{z}) = a] - \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a]$$

In other words, C is now a circuit with i inputs, that is more likely to output a when it after having seen $i-1$ pseudorandom bits, sees another pseudorandom bits, than when seeing a uniform random bit. This suggests the following *randomized* predictor P:

On input $w_1, \ldots, w_{i-1}$, choose $z_i \in \{0, 1\}$ uniformly at random. Then if $C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a$ output z_i, otherwise output $1 - z_i$.

Now, our randomized predictor is correct if $z_i = w_i$ and in fact $C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a$, or when $z_i \neq w_i$ and $C(w_1 \ldots w_{i-1} z_i \tilde{z}) \neq a$. Thus

$$\begin{aligned}
&\Pr[P(w_1 \ldots w_{i-1}) = w_i] \\
=&\frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i = w_i] + \frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) \neq a \mid z_i \neq w_i] \\
=&\frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i = w_i] + \frac{1}{2}(1 - \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i \neq w_i]) \\
=&\frac{1}{2} + \frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i = w_i] - \frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i \neq w_i] \\
=&\frac{1}{2} + \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i = w_i] - \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a] \\
=&\frac{1}{2} + \Pr[C(w_1 \ldots w_{i-1} w_i \tilde{z}) = a] - \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a] \\
>&\frac{1}{2} + \frac{\epsilon}{m}
\end{aligned}$$

where we use between lines 4 and 5 that

$$\begin{aligned}
&\Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a] = \\
&\frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i = w_i] + \frac{1}{2} \Pr[C(w_1 \ldots w_{i-1} z_i \tilde{z}) = a \mid z_i \neq w_i]
\end{aligned}$$

Now finally by a last avering argument we can fix z_i to a constant, thereby obtaining a deterministic predictor. Finally observe that is can be computed by a circuit of size S, since it consists of fixing inputs to constants in C, and finally possibly negating the output.

2 The Nisan-Wigderson construction

Before we present the Nisan-Wigderson PRG we need the notion of a combinatorial design.

Definition 2 (Combinatorial Design)**.** A (l, n, d) design is a family of sets $I_1, \ldots, I_m \subseteq \{1, \ldots, l\}$ that satisfies the following properties:

- $|I_j| = n$ for all j.
- $|I_j \cap I_k| \leq d$ for all $j \neq k$.

The sets of the design correspond to the "almost independent" subsets of the seed on which we will evaluate the hard function.

Recall also the definition of the *hardness* of a language L at input length n $H_L(n)$ defined by $H_L(n) = \max S$ such that no circuit of size S on n inputs can compute L on a $\frac{1}{2} + \frac{1}{S}$ fraction of the inputs of length n.

Definition 3 (Nisan-Wigderson Generator)**.** Given a function $f : \{0,1\}^n \to \{0,1\}$ and a (l, n, d) design $I_1, \ldots, I_m$ the Nisan-Wigderson generator $G^f_{NW} : \{0,1\}^l \to \{0,1\}^m$ is defined by

$$G^f_{NW}(z) = f(z_{I_1}) f(z_{I_2}) \ldots f(z_{I_m}) \ ,$$

where here z_{I_j} is the string of n bits of z indexed by the set I_j.

Theorem 4. *Let $L \in \mathbf{E}$ be a language with characteristic function f such that $H_f(n) > m^2$. Let $I_1, \ldots, I_m \subseteq \{1, \ldots, l\}$ be a (l, n, d) design that can be computed in time $2^{O(n)}$ and where $2^d \leq m$. Let $G = G^f_{NW} : \{0,1\}^l \to \{0,1\}^m$ be defined as above. Then G is a quick $\frac{1}{m}$-PRG for circuits of size m with m inputs.*

(Note that we can use this generator on circuits of size m with fewer than m inputs, by only using a prefix of the output of the generator (corresponding to adding dummy input gates to the circuits).)

Proof. Assume G is not a PRG as stated above. Thus there is a circuit C of size m with m inputs such that

$$|\Pr[C(U_m) = 1] - \Pr[C(G(U_l)) = 1]| > \frac{1}{m}.$$

This means there is a next bit predictor for G as well. Thus let $i \in \{1, \ldots, m\}$ and P be a circuit of size m such that

$$\Pr_z[P(G(z)_1 \ldots G(z)_{i-1}) = G(z)_i] > \frac{1}{2} + \frac{1}{m^2} \ .$$

Note that $G(z)_i$ only depends on the bits of z indexed by I_i. Now by an averaging argument we can fix the bits of z outside of I_i to constants, and maintain the prediction advantage of P. For $x \in \{0,1\}^n$ let $z(x) \in \{0,1\}^l$ be the string with the bits of x in positions indexed by I_i and the fixed constants in the rest of the coordinates. Thus we have

$$\Pr_x[P(G(z(x))_1 \ldots G(z(x))_{i-1}) = G(z(x))_i] > \frac{1}{2} + \frac{1}{m^2} \ .$$

We plug in the definition of G. Thus

$$\Pr_x[P(f(z(x)_{I_1}) \ldots f(z(x)_{I_{i-1}}) = f(x)] > \frac{1}{2} + \frac{1}{m^2} \ .$$

Now here is the crucial observation: For $j < i$ the function $f(z(x)_{I_j}$ depends on *at most* d bits of x. Thus these functions can be computed by circuits $C_1, \ldots, C_{i-1}$, each of size at most $2^d \leq m$ (*every* function on d bits can be computed by a circuit of size 2^d).

Substituting these circuits in the inequality above we have

$$\Pr_x[P(C_1(x) \ldots C_{i-1}(x)) = f(x)] > \frac{1}{2} + \frac{1}{m^2} \ .$$

But now note that $P(C_1, \ldots, C_{i-1})$ is a circuit of size at most $m + (m-1)m = m^2$ computing f on a fraction more than $\frac{1}{2} + \frac{1}{m^2}$ of inputs of length n. This contradicts the hardness assumption of f, and therby completes the proof.

Finally, for the running time of the generator, note that each evaluation of f takes time $2^{O(n)}$, thereby giving a total running time of $m2^{O(n)} = 2^{O(l)}$. □

2.1 Constructions of designs

The first design we present is a very explicit construction. The drawback is that the parameters are not good enough for the so-called high-end setting.

Proposition 5 (Design #1)**.** *For all m, n with $\log m \leq n \leq m$ there exists an efficiently computable (l, n, d) design with $d = \log m$ and $l = O(n^2)$ of size m.*

Proof. Assume without loss of generality that n is a prime-power (otherwise take for instance the smallest power of 2 larger than n). Let now $l = n^2$. Identify $\{1, \ldots, l\} \simeq GF(n) \times GF(n)$. Let $p_1, p_2, \ldots$ be an enumeration of univariate polynomials over $GF(n)$ of degree $\leq d$. Define sets

$$I_j = \{(a, p_j(a)) \mid a \in GF(n)\} \ .$$

In other words, I_j is just the graph of p_j. This gives $\geq n^{d+1} = n^{\log m + 1} \geq m$ sets.

Furthermore we must have $|I_j \cap I_k \leq d$, whenever $j \neq k$, because the polynomial p_j and p_k can intersect in at most d points (or: the polynomial $p_j(x) - p_k(x)$ has at most d roots). □

We can for instance use this design in the "low-end" setting. That is, if we have $L \in \mathbf{E}$ with $H_L(n) = n^{\omega(1)}$ we get a quick generator $G : \{0,1\}^{m^\epsilon} \to \{0,1\}^m$ for all $\epsilon > 0$, and this leads to the conclusion that $\mathbf{BPP} \subseteq \bigcap_{\epsilon > 0} \mathrm{DTIME}(2^{n^\epsilon})$.

This second design is also usable in the "high-end" setting, but the proof is non-constructive. With this, if we have $L \in \mathbf{E}$ with $H_L(n) = 2^{\Omega(n)}$ we get a quick generator $G : \{0,1\}^{O(\log m)} \to \{0,1\}^m$, and this leads to the conclusion that $\mathbf{BPP} = \mathbf{P}$.

Proposition 6 (Design #2)**.** *For all m and n, where $n = c \log m$, for some constant $c > 0$, there exists efficiently computable (l, n, d)-design of size m where $l = O(c^2 \log m)$ and $d = \log m$.*

Proof. Just construct the sets greedily: each time search for a new set with intersection $\log m$ with all already found sets.

Let $I_1, \ldots, I_k \subseteq \{1, \ldots, l\}$, with $|I_j| = n$ be the sets already picked. We will show that another set I exits by the probabilistic method, assuming $k < m$.

Choose a set I as follows: Include each $a \in \{1, \ldots, l\}$ in I independently with probability $2n/l$. Then $E[|I|] = l2n/l = 2n$ and $E[|I \cap I_j|] = n2n/l = 2n^2/l$.

Using Chernoff bounds we get

$$\Pr[|I| < n] = \Pr[|I| < (1 - 1/2)E[|I|]] < \exp(-\frac{(1/2)^2}{2}2n) < 2^{-n/4} \leq 1/2$$

for $n \geq 4$, and

$$\begin{aligned}\Pr[|I \cap I_j| > d] &\leq \Pr[|I \cap I_j| > 4n^2/l] < \Pr[|I \cap I_j| > (1+1)E[|I \cap I_j|]] \\ &< \exp(-1/3 \cdot 2n^2/l) = \exp(-2/3c^2 \log^2 m/l) < 1/2m\end{aligned}$$

for $l = O(c^2 \log m)$, such that $4n^2/l \leq d$.

Taking the union bound over all the k sets already chosen, we find that with strictly positive probability we have $|I| \geq n$ and $|I \cap I_j| \leq d$ for all j. We can then reduce excess elements of I to get a set of size n with the required intersection bounds. □