Electronic Colloquium on Computational Complexity, Report No. 13 (2003)

An ε-Biased Generator in NC^0

Luca Trevisan*

Abstract

Cryan and Miltersen [CM01] recently considered the question of whether there can be a pseudorandom generator in NC^0, that is, a pseudorandom generator such that every bit of the output depends on a constant number k of bits of the seed. They show that for $k = 3$ there is always a distinguisher; in fact, they show that it is always possible to break the generator with a *linear test*, that is, there is a subset of bits of the output whose XOR has a noticeable bias. They leave the question open for $k \geq 4$, and conjecture that every NC^0 generator can be broken by a statistical test that simply XORs some bits of the input. Equivalently, they conjecture that no NC^0 generator can sample an ε-biased space with negligible ε.

We refute the conjecture for $k \geq 5$, and we give a generator that maps n bits into cn bits, so that every bit of the output depends on 5 bits of the seed, and the XOR of every subset of the bits of the output has bias $2^{-\Omega(n/c^4)}$.

We also present a polynomial-time distinguisher for the case $k = 4$, having constant distinguishing probability. We observe that constant distinguishing probability is not achievable via linear tests.

It remains open whether noticeable distinguishing probability can be achieved with linear tests for the case $k = 4$, and whether there is a polynomial time test that breaks every generator (or even just our proposal) for $k \geq 5$.

1 Introduction

A pseudorandom generator is an efficient deterministic procedure that maps a shorter random input into a longer output that is indistinguishable from the uniform distribution by resource-bounded observers.

A standard formalization of the above informal definition is to consider polynomial-time procedures G mapping n bits into $l(n) > n$ bits such that for every property P computable by a family of polynomial-size circuits we have that the quantity

$$\left| \mathop{\mathbf{Pr}}_{z \in \{0,1\}^{l(n)}} [P(z) = 1] - \mathop{\mathbf{Pr}}_{x \in \{0,1\}^n} [P(G(x))] \right|$$

goes to zero faster than any inverse polynomial in n. The existence of such a procedure G is equivalent to the existence of one-way functions [HILL99], pseudorandom functions [GGM86] and pseudorandom permutations [LR88].

What are the minimal computational requirements needed to compute a pseudorandom generator? Linial et al. [LMN93] prove that pseudorandom functions cannot be computed in AC^0 (constant-depth circuits with NOT gates and unbounded fan-in AND and OR

*luca@cs.berkeley.edu. Computer Science Division, U.C. Berkeley.

ISSN 1433-8092

gates),[1] but their result does not rule out the possibility that pseudorandom generators could be computed in AC^0, since the transformation of pseudorandom generators into pseudorandom functions does not preserve bounded-depth. Impagliazzo and Naor [IN96], in fact, present a candidate pseudorandom generator in AC^0. Goldreich [Gol00] suggests a candidate one-way function in NC^0. Recall that NC^0 is the class of functions computed by bounded-depth circuits with NOT gates and bounded fan-in AND and OR gates. In an NC^0 function, every bit of the output depends on a constant number of bits of the inputs. While it is easy to see that there can be no one-way function such that every bit of the output depends on only two bits of the input,[2] it still remains open whether there can be a one-way function such that every bit of the output depends on only three bits of the input.

Cryan and Miltersen [CM01] consider the question of whether there can be pseudorandom generators in NC^0, that is, whether there can be a pseudorandom generator such that every bit of the output depends only on some a constant k number of bits of the input.

They present a distinguisher for $k = 3$, and they observe that their distinguisher is a *linear* distinguisher, that is, it simply XORs a subset of the bits of the output. Cryan and Miltersen formulate a conjecture that implies that there is no pseudorandom generator in NC^0. Specifically, they conjecture that for every constant k and for every generator such that every bit of the output depends on k bits of the input, a linear distinguisher always exist. In order to formulate an equivalent version of the stronger conjecture, let us introduce the notion of a ε-*biased* distribution. For $\varepsilon > 0$, we say that a random variable $X = (X_1, \ldots, X_m)$ ranging over $\{0,1\}^m$ is ε-biased if for every subset $S \subseteq [m]$ we have $1/2 - \varepsilon \leq \mathbf{Pr}[\bigoplus_{i \in S} X_i = 0] \leq 1/2 + \varepsilon$. It is known [NN93, AGHP92] that an ε-biased distribution can be sampled by using only $O(\log(m/\varepsilon))$ random bits, which is tight up to the constant in the big-Oh. So the conjecture of [CM01] can be formulated as stating that there is no ε-biased generator in NC^0 that samples an m-bit ε-biased distribution starting from, say, $o(m)$ random bits and with a negligible ε.

Our Result

We first extend the result of Cryan and Miltersen by giving a (non linear) distinguisher for the case $k = 4$. Our distinguisher has a constant distinguishing probability, which we show to be impossible to achieve with linear distinguishers. Our distinguisher uses semidefinite programming and uses an idea similar to the "correlation attacks" used in practice against block cyphers.

Then we present an ε-biased generator mapping n bits into cn bits such that $\varepsilon = 1/2^{\Omega(n/c^4)}$ and every bit of the output depends only on $k = 5$ bits of the seed. The parameter c can be chosen arbitrarily, and may depend on n. The constant in the $\Omega()$ notation does not depend on c. The construction refutes the conjecture of Cryan and Miltersen.

The main idea in the construction is to develop a generator with $k = 3$ that handles well linear tests that XOR a *small* number of bits, and then develop a generator with $k = 2$ that handles well linear tests that XOR a *large* number of bits. The final generator

[1] To be precise, the results in [LMN93] only rule out security against adversaries running in time $O(n^{(\log n)^{O(1)}})$.

[2] Finding an inverse can be formulated as a 2SAT problem.

outputs the bitwise XOR of the outputs of the two generators, on two independent seeds.

The generator uses a kind of unique-neighbor expander graphs that are shown to exist using the probabilistic method, but that are not known to be efficiently constructable, so the generator is in NC^0 but not in *uniform* NC^0.

2 Preliminaries

Unless otherwise noted, when we give an expression for a function $g : \{0,1\}^n \to \{0,1\}$, additions and multiplications are done in $GF(2)$. We use boldface letters to denote vectors, as in $\mathbf{x} = (x_1, \ldots, x_n)$.

We say that a function $g : \{0,1\}^n \to \{0,1\}$ is *balanced* if $\mathop{\mathbf{Pr}}_{\mathbf{x}}[g(\mathbf{x}) = 1] = 1/2$. We say that a function $g : \{0,1\}^n \to \{0,1\}$ is *unbiased* towards a function $f : \{0,1\}^n \to \{0,1\}$ if $\mathop{\mathbf{Pr}}_{\mathbf{x}}[g(\mathbf{x}) = f(\mathbf{x})] = 1/2$.

Definition 1 (Affine function) *A function $g : \{0,1\}^n \to \{0,1\}$ is* affine *if there are values $a_0, \ldots, a_n \in \{0,1\}$ such that $g(x_1, \ldots, x_n) = a_0 + a_1 x_1 + \ldots + a_n x_n$.*

The following result was proved by case analysis for $k = 3$ in [CM01], and the case $k = 4$ could also be derived from a case analysis appearing in [CM01] (but it is not explicitly stated). The proof for the general case is due to Mossell.

Lemma 2 (Mossell) *Let $g : \{0,1\}^n \to \{0,1\}$ be a non-affine function that depends on only k variables and let l be the affine function that is biased towards g and that depends on a minimal number of variables. That is, for some d, l depends on d variables, $\mathop{\mathbf{Pr}}_{\mathbf{x}}[g(\mathbf{x}) = l(\mathbf{x})] > 1/2$, and g is unbiased towards affine functions that depend on less than d variables.*

Then $\mathop{\mathbf{Pr}}_{\mathbf{x}}[g(\mathbf{x}) = l(\mathbf{x})] \geq 1/2 + 2^{d-k}$.

The Lemma can be stated in somewhat more natural terms in terms of the Fourier spectrum of g, as follows. For a subset $\alpha \subseteq [n]$ define the Fourier coefficient $\hat{g}_\alpha$ as the bias of g towards the linear function that XORs the bits in α, that is, define

$$\hat{g}_\alpha = \mathop{\mathbf{Pr}}_{\mathbf{x}}\left[g(\mathbf{x}) = \bigoplus_{i \in \alpha} x_i\right] - \mathop{\mathbf{Pr}}_{\mathbf{x}}\left[g(\mathbf{x}) \neq \bigoplus_{i \in \alpha} x_i\right]$$

Then the Lemma states that if g depends on k inputs and is not affine, and α is a smallest set such that $\hat{g}_\alpha \neq 0$, then $|\hat{g}_\alpha| \geq 2^{|\alpha|+1-k}$.

For example, for $k = 3$, a non-affine function g is either unbalanced, or it is biased towards one of its inputs; in the latter case it agrees with an input bit (or with its complement) with probability at least 3/4.

For $k = 4$, a function g either is affine, or it is unbalanced, or it has agreement at least 5/8 with an affine function that depends on only one input bit, or it has agreement at least 3/4 with an affine functions that depends on only two input bits.

3 Review of the Case $k = 3$

In this section we summarize the main result of [CM01].

Let $G : \{0,1\}^n \to \{0,1\}^m$ be a generator and let $g_i : \{0,1\}^n \to \{0,1\}$ be the i-th bit of the output of the generator. Suppose each g_i depends on only three bits of the input.

Suppose that one of the g_i is not a balanced function. Then we immediately have a distinguisher.

Suppose that more than n of the g_i are affine. Then one of them is linearly dependent of the others, and we also have a distinguisher.

It remains to consider the case where at least $m - n$ of the functions g_i are balanced and not affine. Let I be the set of i for which g_i is as above. Then, by Lemma 2, for each such g_i there is a linear function l_i that depends on only *one* bit, such that g_i agrees with l_i on a 3/4 fraction of the inputs. In other words, each such g_i has high correlation with one of the bits of its input. By the pigeonhole principle, there is a bit x_j of the seed, and a set C, $|C| \geq 1 + (m - n - 1)/n$, such that the output of $g_i(x_1, \ldots, x_n)$ is correlated to x_j for every $i \in C$. Let $c = |C|$. We see that the average over x of $\max\{\#i \in C : g_i(x) = 0, \#i \in C : g_i(x) = 1\}$ is at least $3c/4$. If c is a sufficiently large constant, then the restriction of the generator to C has constant statistical distance from the uniform distribution over c bits, for which that average value is $c/2 + O(\sqrt{c})$. By the Vazirani XOR Lemma [Vaz86], it also follows that the XOR of some subset of the bits of C has constant bias.[3]

While the above analysis uses the same ideas as in [CM01], it is slightly better because we achieve constant bias instead of inverse polynomial bias.

We state for future reference the following result that follows from the above analysis.

Lemma 3 *For every $\delta > 0$ there are constant $c_\delta = O(1/\delta^2)$ and $\varepsilon_\delta = 2^{-O(1/\delta^2)}$ such that the following holds. Let $G : \{0,1\}^n \to \{0,1\}^m$, and let $G(x) = (g_1(x), \ldots, g_m(x))$. Suppose that each function $g_i(x)$ agrees with a bit x_j or with its complement with probability at least $1/2 + \delta$, and that $m \geq 1 + c_\delta n$; then there is a set $C \subseteq [m]$ such that $\sum_{i \in C} g_i(x) \pmod 2$ has bias at least ε_δ.*

In particular, we can compute that when we flip 4 random coins, the average of the maximum between the number of zeroes and ones is $2.75 < \frac{3}{4} \cdot 4$, so we can set $c_{1/4} = 3$. Also, when we flip 10 random coins, the average of the maximum between the number of zeroes and ones is $6.23 < \frac{5}{8} \cdot 10$, so we can set $c_{1/8} = 9$.

4 Distinguisher for the Case $k = 4$

4.1 Preliminaries

Let $G : \{0,1\}^n \to \{0,1\}^m$ be a generator and let $g_i : \{0,1\}^n \to \{0,1\}$ be the i-th bit of the output of the generator. Suppose each g_i depends on only four bits of the input.

Again, it is easy to construct a distinguisher if any of the g_i is unbalanced, or if more than n of the g_i are linear.

If one of the g_i is biased towards one of the bits of its input, then it follows from Lemma 2 that it must agree with that bit or its complement with probability at least 5/8. Then, if more than $c_{1/8} n = 9n$ of the functions g_i have bias towards one bit, then we can obtain a distinguisher from Lemma 3.

[3] The Vazirani XOR Lemma is the fact that if $X_1, \ldots, X_t$ are 0/1 random variables, then they are uniform and mutually independent if and only if for every non-empty $S \subseteq [t]$ we have $\mathbf{Pr}[\bigoplus_i X_i = 1] = 1/2$.

It remains to consider the case where at least $m - 10n$ of the functions are balanced, non-linear, and unbiased towards single bits. Following [CM01], we call such functions *problematic*. It follows from Lemma 2 that for each problematic g there is an affine function l of two variables that agrees with g on a $3/4$ fraction of the inputs.

Let P be the set of i such that g_i is problematic. For each such i we denote by l_i the affine function of two inputs that agrees with g_i on a $3/4$ fraction of the inputs.

4.2 The Distinguisher Based on Semidefinite Programming

Given a string $r_1, \ldots, r_m \in \{0,1\}^m$, consider the following linear system over $GF(2)$ with two variables per equation.

$$\forall i \in P. l_i(x) = r_i \tag{1}$$

We will argue that the largest fraction of satisfying assignments in the system (1) is distributed differently if $r_1, \ldots, r_m$ is uniform or if it is the output of G.

Lemma 4 *If $r_1, \ldots, r_m$ is the output of G, then, for every $\varepsilon > 0$, there is a probability at least ε that at least at $3/4 - \varepsilon$ fraction of the equations in (1) are satisfiable.*

PROOF: Pick a random $z \in \{0,1\}^n$ and consider the agreement between $G(z)_{|P}$ and $l_i(z)$ for $i \in P$. This agreement is the sum of $|P|$ random variables each of whom has average at least $3/4$. therefore the average agreement is at least $3|P|/4$. By Markov inequality, there is a probability at least ε that the agreement is at least $(3/4 - \varepsilon)|P|$. Whenever this happens, z is a witness of the fact that at least a $3/4 - \varepsilon$ fraction of the equations can be satisfied. □

Lemma 5 *If $r_1, \ldots, r_m$ is chosen uniformly at random from $\{0,1\}^m$, and $|P| > (1/2\delta^2)(\ln 2)(n + c)$, then the probability that there is an assignment that satisfies more than a $1/2 + \delta$ fraction of the equations of (1) is at most 2^{-c}.*

PROOF: Fix an assignment z; then the probability that a fraction at least $1/2 + \delta$ of the r_i agree with $l_i(z)$ is at most $e^{-2\delta^2 m} \leq 2^{-c-n}$. By a union bound, there is at most a probability 2^{-c} that such a z exists. □

Given a system of linear equations over $GF(2)$ with two variables per equation, it is NP-hard to determine the largest number of equations that can be satisfied, but the problem can be approximated to within a .878 factor using semidefinite programming [GW95].

We can then fix ε and δ small enough so that $.878(3/4 - \varepsilon) > 1/2 + \delta$, and we get a polynomial time algorithm that distinguishes between the two cases. For example, we can fix $\delta = .158$ and $\varepsilon = 10^{-4}$; then, if there are at least $14n$ problematic functions in the output of G and n is large enough, the above procedure has constant distinguishing probability.

This means that if $m > 24n$ we always have a distinguisher.

4.3 Correlation Attacks

In this section we discuss how our distinguisher for the case $k = 4$ can be seen as a "correlation attack."

Correlation attacks are a class of attacks that are often attempted in practice against candidate pseudorandom generators,[4] see e.g. the introduction of [JJ99] for an overview.

The basic idea is as follows. Given a candidate generator $G : \{0,1\}^n \to \{0,1\}^m$, where $G(x) = g_1(x), \ldots, g_m(x)$, we first try and find linear relations between input bits and output bits that are satisfied with non-trivial probability. For example, suppose we find coefficients $a_{i,j}$, $b_{i,j}$ and c_j such that each of the equations

$$\begin{array}{l} \sum_{i=1}^n a_{i,1}x_i + \sum_{i=1}^m b_{i,1}g_i(x) = c_1 \pmod 2 \\ \sum_{i=1}^n a_{i,2}x_i + \sum_{i=1}^m b_{i,2}g_i(x) = c_2 \pmod 2 \\ \ldots \\ \sum_{i=1}^n a_{i,t}x_i + \sum_{i=1}^m b_{i,t}g_i(x) = c_t \pmod 2 \end{array} \tag{2}$$

is satisfied with probability bounded away from 1/2.

Now we want to use this system of equations in order to build a distinguisher. The distinguisher is given a sample $\mathbf{z} = (z_1, \ldots, z_m)$ and has to decide whether $\mathbf{z}$ is uniform or is the output of G. The distinguisher substitutes z_i in place of $g_i(x)$ in (2) and then tries to find an $\mathbf{x}$ that maximizes the number of satisfied equations. The hope is that, if $\mathbf{z} = G(\mathbf{x})$, then we will find $\mathbf{x}$ as a solution of the optimization problem.

Unfortunately, maximizing the number of satisfied equations in a linear system over $GF(2)$ is an NP-hard problem, and, in fact, it is NP-hard to achieve an approximation factor better than 1/2 [Hås97]. In practice, one uses belief-propagation algorithms that often work, although the method is typically not amenable to a formal analysis.

In the previous section, we were able to derive a formal analysis of a related method because we ended up with a system of equations having only two variables per equation, a class of instances for which good approximation algorithms are known. Furthermore, we did not try to argue that, when the method is applied to the output of the generator, we are likely to recover the seed; instead, we argued that just being able to approximate the largest fraction of satisfiable equations gives a way to distinguish samples of the generators from random strings.

5 Constructions for the Case $k = 5$

5.1 Preliminaries

We will construct a generator mapping $2n$ bits into cn bits; we think of c as an arbitrarily large constant (for every c, the construction is possible for every large enough n), although super-constant c is also achievable.

In fact, we will construct two generators: one will be good against linear tests that involve a small number of output bits (we call them *small tests*), and another is good against linear tests that involve a large number of output bits (we call them *large tests*). The final generator will be obtained by computing the two generators on independent seeds, and then XOR-ing their output bit by bit. In this way, we fool every possible test.

The generator that is good against large tests is such that every bit of the output is just the product of two bits of the seed. We argue that the sum (modulo 2) of t output bits of the generator has bias exponentially small in t/c^2, where c, as above, is the stretch of the generator.

[4] Pseudorandom generators are called "block ciphers" in the applied cryptography literature.

Then we describe a generator that completely fools linear tests of size up to about n/c^2, and such that every bit of the output is the sum of three bits of the seed. Combined with the generator for large tests, we get a generator in $\mathrm{NC}^0{}_5$ such that every linear test has bias $2^{-O(n/c^4)}$.

5.2 The Generator for Large Tests

Let us call the bits of the seed $y_1, \ldots, y_n$.

Let K be an undirected graph formed by $n/(2c+1)$ disjoint cliques each with $2c+1$ vertices. Then K has n vertices, that we identify with the elements of $[n]$, and $cn = m$ edges. Fix some ordering of the edges of K, and let (a_j, b_j) be the j-th edge of K. Define the functions $q_1, \ldots, q_m$ as $q_j(y_1, \ldots, y_n) = y_{a_j} y_{b_j}$.

Claim 1 *For every subset $S \leq [m]$, the function $q_S(\mathbf{y}) = \sum_{j \in S} q_j(\mathbf{y})$ is such that*

$$\frac{1}{2} - \left(\frac{1}{2}\right)^{1+|S|/(2c^2+c)} \leq \mathop{\mathbf{Pr}}_{\mathbf{y}}[q_S(\mathbf{y}) = 0] \leq \frac{1}{2} + \left(\frac{1}{2}\right)^{1+|S|/(2c^2+c)}$$

The proof relies on the following two lemmas. The first one is from [CM01], and it is easy to prove it by induction on the number of variables, and the second one is standard and it is easy to prove it by replacing $\{0,1\}$ with $\{-1,1\}$ and $\oplus$ with multiplication.

Lemma 6 ([CM01]) *Let p be a non-constant degree-2 multilinear polynomial over $GF(2)$. Then $1/4 \leq \mathbf{Pr}[p(x) = 0] \leq 3/4$.*

Lemma 7 *Let $X_1, \ldots, X_t$ be independent 0/1 random variables, and suppose that for every i we have $\delta \leq \mathbf{Pr}[X_i = 0] \leq 1 - \delta$. Then*

$$\frac{1}{2} + \frac{1}{2}(1-2\delta)^t \leq \mathbf{Pr}\left[\bigoplus_i X_i = 0\right] \leq \frac{1}{2} + \frac{1}{2}(1-2\delta)^t$$

We can now prove Claim 1.

PROOF: [Of Claim 1] We can see S as a subset of the edges of K. Each connected component of K has $2c^2 + c$ edges, so S contains edges coming from at least $|S|/(2c^2+c)$ different connected components, let us call t this number. If we decompose the summation $\sum_{j \in S} q_j(y_1, \ldots, y_n)$ into terms depending on each of the connected components, then each term is a non-trivial degree-2 polynomial, and the t terms are independent random variables when $y_1, \ldots, y_n$ are picked at random. We can then apply Lemma 7, where the X_i are the values taken by each of the t terms in the summation, $\delta = 1/4$, and $t = |S|/(2c^2+c)$. □

5.3 The Generator for Small Tests

Let $A \in \{0,1\}^{n \times m}$ be a matrix such that every row is a vector in $\{0,1\}^n$ with exactly three non-zero entries, and let also A be such that every subset of σ rows are linearly independent. Let $A_1, \ldots, A_m$ be the rows of A.

We define the linear functions $l_1, \ldots, l_m$ as $l_i(\mathbf{x}) = A_i \cdot \mathbf{x}$. Note that each of these linear functions depends on only three bits of the input.

Claim 2 *For every subset $S \leq [m]$, $|S| < \sigma$, the function $l_S(\mathbf{x}) = \sum_{j \in S} l_j(\mathbf{x})$ is balanced.*

PROOF: We have $l_S(\mathbf{x}) = (\sum_{j \in S} A_j) \cdot \mathbf{x}$, and since $\sum_{j \in S} A_j$ is a non-zero element of $\{0,1\}^n$, it follows that $l_S()$ is a non-trivial linear function, and therefore it is balanced. □

There are matrices with linear σ.

Lemma 8 *For every $c = c(n) = o(\sqrt{n}/(\log n)^{3/4})$ and for sufficiently large n there is a 0/1 matrix A with cn rows and n columns such that every row has exactly three non-zero entries and such that every subset of $n/(4e^2c^2(n))$ rows are linearly independent.*

This is a standard probabilistic construction Similar calculations have been done several times, for example in [BKPS98, BSW01, BOT02]. We give the calculation in the Appendix for the sake of self-containment.

5.4 Putting Everything Together

Theorem 9 *For every c and sufficiently large n, there is a generator in ${NC^0}_5$ mapping n bits into cn bits and sampling an ε-biased distribution, where $\varepsilon = 2^{-n/O(c^4)}$.*

6 Generator for the case $k = 4$

In this section we give a construction (that is essentially from [CM01]) of a generator for small tests and with $k = 2$. Together with results from the previous section, this will give an ε-biased generator with inverse-polynomial ε in ncz_4.

Let H be an undirected graph with n vertices, that we identify with $[n]$, having cn edges and girth γ. Fix some ordering of the edges of H, and let (a_j, b_j) be the j-th edge of H. We define the linear functions $l_1, \ldots, l_m$ as $l_i(x_1, \ldots, x_n) = x_{a_j} + x_{b_j}$.

Claim 3 *For every subset $S \leq [m]$, $|S| < \gamma$, the function $l_S(\mathbf{x}) = \sum_{j \in S} l_j(\mathbf{x})$ is balanced.*

PROOF: We can see S as a set of edges in H, and l_S as the function that sums x_i for each vertex i that is incident on an odd number of edges in S. Since $|S| < g$, the subgraph of H induced by the edges of S is a forest, and so some vertex must have odd degree (in fact, some vertex must have degree one). It follows that l_S is the sum of a non-empty subset of its inputs, and so it is balanced.[5] □

We can let γ be as large as about $\log_c n$.

Lemma 10 ([LPS88]) *For every c and for sufficiently large n there are explicitely constructible graphs H with n vertices, cn edges, and girth $\Omega((\log n)/(\log c))$.*

Theorem 11 *For every c and sufficiently large n, there is a generator in uniform ${NC^0}_4$ mapping n bits into cn bits and sampling an ε-biased distribution, where $\varepsilon = n^{-1/O(c^2 \log c)}$.*

[5] Equivalently, we proved that every subset of $< \gamma$ of the functions l_i are linearly independent.

7 Conclusions

Several questions remain open.

Even for the case $k = 3$, we only know how to break the generator assuming that the output length is a sufficiently large constant multiple than the seed length. It is not clear whether there is a linear test, or even a polynomial time algorithm, that breaks the case $k = 3$ when, say, $m = n + 1$.

It is still open whether there can be an ε-biased generator with negligible ε in the case $k = 4$. We conjecture that this is not the case for sufficiently large linear stretch, but we do not have a strong feeling about what happens for very small stretch.

The main open question is whether our generator for the case $k = 5$ can be broken by a polynomial time algorithm and, in general, whether polynomial time algorithms can break all NC^0 generators.

Mossel and Shpilka [MS03] give a construction of an ε-biased generator mapping n bits into $n^{\tilde{O}(\sqrt{k})}$ bits such that ε is negligible and every bit of the output depends on k bits of the seed. They also show that a generator mapping n bits into $n^{k/2}$ bits and such that every bit of the output depends on k bits of the seed can be broken by a linear test. It remains an interesting question to characterize the largest stretch achievable by an ε-biased generator in ${NC^0}_k$ with negligible ε.

Acknowledgements

I wish to thank David Wagner suggesting the relevance of correlation attacks. Thanks to Elchanan Mossel for helpful discussions and for the proof of Lemma 2.

References

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[BKPS98] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. On the complexity of unsatisfiability proofs for random k-cnf formulas. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[BOT02] Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded degree graphs. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2002.

[BSW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow: Resolution made simple. *Journal of the ACM*, 48(2), 2001.

[CM01] Mary Cryan and Peter B. Miltersen. On pseudorandom generators in NC0. In *Proceedings of MFCS'01*, 2001.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. Technical Report TR00-090, ECCC, 2000.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. Preliminary version in *Proc. of STOC'94*.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 1–10, 1997.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[IN96] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996.

[JJ99] T. Johansson and F. Jonsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In *Proceedings of EUROCRYPT'99*, 1999.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[LR88] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 2(17):373–386, 1988.

[MS03] E. Mossel and A. Shpilka. Personal communication, 2003.

[NN93] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications, 1993.

[Vaz86] U. Vazirani. *Randomness, Adversaries and Computation*. PhD thesis, University of California, Berkeley, 1986.

A Appendix

A.1 Proof of Lemma 8

Definition 12 *We say that a bipartite graph (L, R, E) is (σ, α)-expanding if for every subset $S \subseteq L$ of vertices on the left, if $|S| \leq \sigma$ then $|\Gamma(S)| > \alpha \cdot |S|$, where $\Gamma(S)$, defined as*

$$\Gamma(S) = \{v \in R : \exists u \in S.(u, v) \in E\}$$

is the neighborhood of S.

Lemma 13 *For every $c(n) = o(\sqrt{n}/(\log n)^{3/4})$ and sufficiently large n there is a $(\sigma, 3/2)$-expanding graph $([c(n) \cdot n], [n], E)$ with $\sigma = n/(4e^4c^2(n))$ such that every vertex on the left has degree 3.*

Proof: We construct the graph at random by connecting each vertex on the left to three distinct randomly chosen vertices on the right. (For different left vertices the random choices are independent.)

Fix a size s, $3 \leq s \leq n/(2e^2c)$, and consider the probability that there is a subset $S \subseteq [cn]$ of s vertices on the right whose neighborhood is contained into a set $T \subseteq [n]$ of $3s/2$ vertices on the left. This probability is less than $(\frac{3s}{2n})^{3s}$. The number of possible choices for S is $\binom{cn}{s}$ and the number of possible choices for T is $\binom{n}{3s/2}$, and, by a union bound, the probability that the construction fails to satisfy the required property is at most

$$\sum_{s=3}^{\sigma} \binom{cn}{s} \cdot \binom{n}{3s/2} \left(\frac{3s}{2n}\right)^{3s} \tag{3}$$

and using the inequality $\binom{n}{k} \leq (\frac{ne}{k})^k$ we can see that Expression (3) is at most

$$\sum_{s=3}^{\sigma} \left(\frac{ecn}{s}\right)^s \cdot \left(\frac{2en}{3s}\right)^{3s/2} \cdot \left(\frac{3s}{2n}\right)^{3s} \tag{4}$$

$$\leq \sum_{s=3}^{\sigma} \left(\frac{2e^2c\sqrt{s}}{\sqrt{n}}\right)^s \tag{5}$$

$$= O\left(\left(\frac{c}{\sqrt{n}}\right)^3 + \left(\frac{c}{\sqrt{n}}\right)^4 \cdot (\log n)^3\right) = o(1) \tag{6}$$

Where the last line can be verifier by breaking the sum in Expression (5) up into the the term $s = 3$, which is $O((c/\sqrt{n})^3)$, the terms $s = 4, \ldots, 2\log n$, each of which is at most $O(c\sqrt{\log n}/\sqrt{n})^4$, and the remaining terms, each of which is at most $1/n^2$. □

Now let us pick a $(\sigma, 1.5)$-expanding graph $G = ([cn], [n], E)$, with $\sigma = n/(4e^4c^2)$, as in the above lemma.

Then for every subset $S \subseteq [cn]$ of left vertices, with $|S| \leq \sigma$ we not there is an element $j \in \Gamma(S)$ such that j has a unique neighbor in S. This is because there are only $3|S|$ edges going between S and $\Gamma(S)$, and $|\Gamma(S)| > 3|S|/2$, and so it is not possible for all vertices in $\Gamma(S)$ to have two neighbors or more.

Consider the $cn \times n$ adjacency matrix A of G, defined so that for $i \in [cn]$ and $j \in [n]$ we have $A_{i,j} = 1$ if $(i,j) \in E$ and $A_{i,j} = 0$ otherwise. Then every row has precisely three non-zero entries.

We want to argue that every subset of $\leq \sigma$ rows are linearly independent.

Consider then a subset $\{A_i\}_{i \in S}$ of at most σ rows of A. If we think of S as a set of left vertices in G, then there is a right vertex $j \in [n]$ that has precisely one neighbor in S, that is, there is a $j \in [n]$ for which there is exactly one $i \in S$ such that $A_{i,j} = 1$. It then follows that the row A_i cannot be obtained as a linear combination of the other rows in S.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'