Electronic Colloquium on Computational Complexity, Report No. 88 (2013)

AC^0 Pseudorandomness of Natural Operations

Zachary Remscrim* and Michael Sipser†
MIT
Cambridge, MA 02139

June 15, 2013

Abstract

A function $f : \Sigma^* \to \Sigma^*$ on strings is AC^0-pseudorandom if the pair $(x, \hat{f}(x))$ is AC^0-indistinguishable from a uniformly random pair (y, z) when x is chosen uniformly at random. Here $\hat{f}(x)$ is the string that is obtained from $f(x)$ by discarding some selected bits from $f(x)$.

It is shown that several naturally occurring functions are AC^0-pseudorandom, including convolution, nearly all homomorphisms, Boolean matrix multiplication, integer multiplication, finite field multiplication and division, several problems involving computing rank and determinant, and a variant of the algebraic integer problem.

1 Introduction

1.1 The Problem

Random-like behavior occurs naturally in many places in mathematics. For example, the binary representations of numbers π, e and $\sqrt{2}$ look random. Various conjectures about the distribution of prime numbers and the number of prime factors of an integer say that these behave randomly. However, very little progress has been made in proving that such behaviors are indeed pseudorandom in any formal sense. For example, it is not known that the binary representations of π, e or $\sqrt{2}$, contain all substrings with the expected frequencies or even that the substring 11 appears infinitely often.

In this paper, we propose to study the pseudorandom characteristics of naturally occurring mathematical functions by using the tools of complexity theory. The theory of pseudorandom generators provides a good starting point, but there the motivation is somewhat different than ours. Pseudorandom generators are used to good effect in cryptographic protocols and in derandomizing probabilistic algorithms, and they are designed with those goals in mind. Our objective is to study the basic operations themselves, such as Boolean convolution and integer multiplication, for their pseudorandom properties. These functions occur naturally—they have not been specifically designed to have pseudorandom behaviour—yet we can show that they do exhibit such behavior.

We use the integer multiplication function as a motivating example. Let X and Y be n-bit binary strings representing non-negative integers and let Z be the $2n$-bit string representing $Z = X \times Y$. Take X and Y to be selected uniformly at random from 0 to $2^n - 1$, and consider the characteristics of Z. Does Z look random? The low-order bit of Z certainly does not; it is 0 with probability 3/4. The other very low order bits look non-random for a similar reason. The

*remscrim@mit.edu Research supported in part by an Akamai Fellowship.
†sipser@math.mit.edu

ISSN 1433-8092

very high order bits of Z likewise appear non-random. However, if we discard these problematic very low and very high order bits from Z, the result could conceivably be pseudorandom in some appropriate sense.

We show that, for uniformly randomly selected X, Y, the string consisting of X, Y and all $2n$ bits of $X \times Y$, except the lowest and highest n^α bits, for any $\alpha > 0$, is indistinguishable from truly random strings by AC^0 circuits. In fact, we show something even stronger: for almost all Y, the string consisting of X and all $2n$ bits of $X \times Y$, except the lowest and highest n^α bits, is indistinguishable from random by AC^0 circuits that have Y built-in (the circuit is allowed to depend on Y).

AC^0 circuits are circuits consisting of AND, OR, and NOT gates of unbounded fan-in, such that the size of the circuit (the total number of gates) is polynomial in the size of the input and the depth of the circuit (the number of gates on the longest path from the input to the output) is a constant. Techniques for proving strong lower bounds on low-depth circuits [Ajt83],[FSS84],[Yao85],[Has86] enable us to prove the AC^0-pseudorandomness of explicit functions without using any unproven complexity-theoretic assumptions. Moreover AC^0 is powerful enough to describe basic tests for pseudorandomness.

We now formally define what it means for a function to look random to AC^0 circuits. For ease of exposition, we consider functions that operate on strings of a specific length, whereas we really have in mind a family of functions and their asymptotic properties. For a function $f : \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_h} \to \{0,1\}^k$, define the function $g : \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_h} \to \{0,1\}^n$, where $n = m_1 + \cdots + m_h + k$, such that $g(x_1, \ldots, x_h) = x_1 \circ \cdots \circ x_h \circ f(x_1, \ldots, x_h)$ is the concatenation of $x_1, \ldots, x_h$ and $f(x_1, \ldots, x_h)$. Let μ_n denote the distribution of $g(x_1, \ldots, x_h)$, when each x_i is drawn uniformly at random from $\{0,1\}^{m_i}$. For any binary predicate $P_n : \{0,1\}^n \to \{0,1\}$, let $E_{\mu_n}[P_n]$ denote the expected value of P_n when inputs are drawn according to the distribution μ_n and $E[P_n]$ denote the expected value of P_n when inputs are drawn uniformly at random from $\{0,1\}^n$. We say that the distribution μ_n ***ϵ-fools*** the function P_n if $|E_{\mu_n}[P_n] - E[P_n]| < \epsilon$ and that the original function f is ***AC^0-pseudorandom*** if the corresponding distribution μ_n ϵ-fools every P_n that is computable in AC^0, where $\epsilon = O(2^{-n^\kappa})$, for constant $\kappa > 0$. This is, of course, quite similar to the standard pseudorandom generator model for AC^0 circuits (see, for instance, [Nis91], [NW94]), with the exception of the fact that we impose the stronger requirement that both the input and output of the function together are indistinguishable from random bits, instead of only requiring that the output is indistinguishable. Also, while the focus of this paper is the pseudorandomness of functions, not the difficulty of actually computing the functions, it is still worth noting that the functions considered can be computed in a low complexity class such as $AC^0[2]$ (AC^0 circuits that are allowed unbounded fan-in parity gates) or TC^0 (constant depth circuits with unbounded fan-in majority gates), but still produce strings that are indistinguishable from truly random strings by AC^0 circuits.

A somewhat similar question was considered in the recent paper [Gre12], concerning the Möbius function $\mu : \mathbb{N} \to \{-1, 0, 1\}$, which is defined such that $\mu(1) = 1$, $\mu(x) = 0$ when x has a nontrivial perfect square factor, and $\mu(x) = (-1)^k$, when x has no nontrivial perfect square factors, where k is the number of distinct primes in the prime factorization of x. It was shown that μ is asymptotically orthogonal to any AC^0 computable function $f : \mathbb{N} \to \{-1, 1\}$ (that is to say $\frac{1}{N}\sum_{x=1}^{N} f(x)\mu(x) = o(1)$). Tools from complexity theory were used to show that a naturally occurring function looks random to AC^0 circuits. It is worth noting that the functions considered in our paper have much longer output than the Möbius function; we consider functions which, on an n bit input, produce a $\Omega(n)$ bit output, while the Möbius function maps an n bit input to only a constant sized output.

Another example of a natural problem studied for its pseudorandom properties is the alge-

braic number problem, which, as noted in [KLL84], was initially proposed by Manuel Blum. An algebraic number is a root of a polynomial with integer coefficients. For example, $\sqrt{2}, \sqrt{3}$, and $(1+\sqrt{5})/2$ are all algebraic numbers. The algebraic number problem involves selecting, uniformly at random, an algebraic number ζ of bounded degree d and height H (where the degree of ζ is the degree of the (unique) primitive irreducible polynomial that has ζ as a root, and the height is the Euclidean length of the coefficient vector of that polynomial). The string to be considered is a portion of the binary expansion of the fractional part of ζ. In [KLL84], it was shown that, given the first $O(d^2 + d \log H)$ bits of an algebraic number ζ, it is possible, in deterministic polynomial time to determine the minimal polynomial of ζ. Since the next bit of the binary expansion of ζ can easily be obtained if given the minimal polynomial of ζ, this immediately implied that such strings do not pass all polynomial time tests. We consider a closely related problem, which is identical to the above problem, except that we select ζ only from the ring of integers of certain algebraic number fields. By the argument used in [KLL84], this variant also does not pass all polynomial time tests. However, we show that it does pass all AC^0 tests. While this is certainly far away from showing anything about the pseudorandomness properties of a single value, such as $\sqrt{2}$, it might be a step in that direction.

1.2 Main Results

This paper illustrates two techniques for demonstrating that functions are AC^0-pseudorandom. The first technique makes use of the result in [Bra09] that resolved the long standing Linial-Nisan conjecture [LN90]. We use this technique to show that almost all "reasonably sized" homomorphisms are AC^0-pseudorandom, and, moreover, that convolution, integer multiplication and matrix multiplication are AC^0-pseudorandom.

Our second technique involves reducing the (provably hard) problem of computing parity to the problem of distinguishing certain distributions from random. The second technique is related to the method in [Nis91],[NW94], in that we show that the structure of certain multiplication problems is a naturally occuring example of the combinatorial designs they employ. We use this technique to show that an alternate form of the multiplication problem, where one multiplicand is substantially longer than the other, is AC^0-pseudorandom. One consequence of this result will be the existence of a simple, multiplication-based pseudorandom generator with the same stretch and hardness parameters as the Nisan-Wigderson generator. An additional consequence is the fact that no AC^0 circuit can compute the product of an n-bit number and a superpolylog(n)-bit number (that is to say, a sequence of numbers whose length grows faster than $\log^c n$, for all constants $c > 0$). This shows that the result from [CSV84], which states that an AC^0 circuit can compute the product of an n-bit and a $O(log^c n)$-bit value, is optimal.

Additionally, we show, via a reduction from the multiplication problem, that a certain variant of the algebraic integer problem looks random to AC^0. These same techniques can be used to show that a variety of additional problems, such as finite field multiplication and division, matrix inversion, computing determinants, and an iterated version of convolution are also AC^0-pseudorandom.

We prove the following theorems:

Let $Hom(\{0,1\}^m, \{0,1\}^k)$ denote the set of homomorphisms from $\{0,1\}^m$ to $\{0,1\}^k$ (or, in other words, the linear maps from the vector space $\{0,1\}^m$ to the vector space $\{0,1\}^k$).

Theorem 1. *If $k = m^u$, for any fixed constant $u > 0$, then all but an exponentially small fraction of all $f \in Hom(\{0,1\}^m, \{0,1\}^k)$ are AC^0-pseudorandom.*

Let $\mathrm{CONV}_{r,s,k} : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^k$ denote the Boolean convolution function, which

takes a $X \in \{0,1\}^r$ and $Y \in \{0,1\}^s$ to the middle k-bits of the $r+s-1$ bit long convolution of X and Y.

Theorem 2. *If $s = r^u$ and $k = r + s - (MIN(r,s))^\alpha$, for any fixed constants $u > 0$ and $0 < \alpha < 1$, then $CONV_{r,s,k}$ is AC^0-pseudorandom. In particular, if $r = s$ and $k = 2r - r^\alpha$, for any $0 < \alpha < 1$, then $CONV_{r,s,k}$ is AC^0-pseudorandom.*

Let $\text{MULT}_{r,s,k} : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^k$ denote the integer multiplication function, which takes a $X \in \{0,1\}^r$ and $Y \in \{0,1\}^s$ to the middle k-bits of the $r+s$ bit long product of X and Y.

Theorem 3. *If $s = r^u$ and $k = r + s - (MIN(r,s))^\alpha$, for any fixed constants $u > 0$ and $0 < \alpha < 1$, then $MULT_{r,s,k}$ is AC^0-pseudorandom. In particular, if $r = s$ and $k = 2r - r^\alpha$, for any $0 < \alpha < 1$, then $MULT_{r,s,k}$ is AC^0-pseudorandom.*

Let $\text{MATRIX-MULT}_{r,s} : \{0,1\}^{rs} \times \{0,1\}^{rs} \to \{0,1\}^{s^2}$ denote the matrix multiplication function, which, on input a $s \times r$ matrix A and a $r \times s$ matrix B (both of which are encoded as strings in $\{0,1\}^{rs}$ in the obvious way), produces the $s \times s$ matrix AB.

Theorem 4. *If $s = r^u$, for any fixed constant $u > 0$, then $MATRIX\text{-}MULT_{r,s}$ is AC^0-pseudorandom.*

2 The Linial-Nisan-Braverman Technique

2.1 Braverman's Theorem

Braverman [Bra09] resolved the long standing Linial-Nisan conjecture [LN90]. We now state this theorem, which provides a simple sufficient condition for a distribution to appear random to AC^0 circuits. For a distribution μ_n with support $\{0,1\}^n$, we say that μ_n is a (β, r)-approximation if every restriction of μ_n to r coordinates is β-close to the uniform distribution on $\{0,1\}^r$ (two distributions are β-close if the statistical distance between them is at most β). The theorem states that if a distribution μ_n is a $(\beta, r(s,d,\epsilon))$-approximation, for sufficiently large r and sufficiently small β, then it ϵ-fools all depth d AC^0 circuits of size s.

Theorem. *[Bra09] Every $(\beta, r(s,d,\epsilon))$-approximation ϵ-fools all depth d AC^0 circuits of size s, where*

$$r(s,d,\epsilon) = \left(\log \frac{s}{\epsilon}\right)^{O(d^2)}$$

and

$$\frac{\epsilon}{\beta} > 2n^{r(s,d,\epsilon)}.$$

In particular, every $(2^{-n^\gamma}, n^\delta)$-approximation, for constants $\kappa < \delta < \gamma < 1$, will 2^{-n^κ}-fool polynomial sized circuits of any constant depth, for sufficiently small constant α. In this paper, any function f for which the corresponding distribution μ_n, as defined above, meets this condition, will be said to have the Linial-Nisan-Braverman property, or LNB property for short. In fact, many of the functions considered will have an even stronger property: their corresponding distributions will be $(0, n^\delta)$-approximations (or, in other words, every restriction of μ_n to n^δ coordinates will simply be the uniform distribution, rather than being only close to the uniform distribution).

2.2 Application to Homomorphisms

Let us now restrict our attention to homomorphisms from $\{0,1\}^m$ to $\{0,1\}^k$, the set of which we denote by $\text{Hom}(\{0,1\}^m, \{0,1\}^k)$ (or, in other words, viewing $\{0,1\}^m$ and $\{0,1\}^k$ as vector spaces,

we consider the set of linear maps). It will be shown that it is particularly simple to determine if a given homomorphism has the Linial-Nisan-Braverman property, and, moreover, that many homomorphisms have this property, and hence appear random to AC^0 circuits.

Every $f \in \mathrm{Hom}(\{0,1\}^m, \{0,1\}^k)$ corresponds to a $k \times m$ matrix F, with entries in $\{0,1\}$, such that $f(X) = FX$, for $X \in \{0,1\}^m$. For any $R \subseteq \{1,\ldots,k\}$ and $C \subseteq \{1,\ldots,m\}$, let $F_{R,C}$ be the submatrix of F consisting of rows R and columns C. The following lemma shows that having the Linial-Nisan-Braverman property is equivalent to certain submatrices of F being full rank. As before, $n = m + k$.

Lemma 1. *$f \in Hom(\{0,1\}^m, \{0,1\}^k)$ has the Linial-Nisan-Braverman property if and only if $\exists \delta > 0$ such that $\forall R \subseteq \{1,\ldots,k\}, C \subseteq \{1,\ldots,m\}$ with $|R| + |C| = n^\delta$, the submatrix $F_{R,\overline{C}}$ is full rank, where $\overline{C} = \{1,\ldots,m\} \setminus C$.*

Proof. First, consider a function $f : \{0,1\}^m \to \{0,1\}^k$ whose corresponding matrix F meets the above condition. We show that f has the Linial-Nisan-Braverman property. To do this, let $X \in \{0,1\}^m$ be an arbitrary element, $Y \in \{0,1\}^n$ be the concatenation of X and $f(X)$, and μ_n be the distribution of Y given a uniformly randomly selected X. By definition, f has the Linial-Nisan-Braverman property if μ_n is n^δ-independent. To see that f has this property, imagine that an adversary selects some n^δ sized subset of coordinates of Y. We must show that the distribution μ_n, when restricted to these coordinates is the uniform distribution. Each coordinate is either a coordinate of the input X or a coordinate of the output $f(X)$. Of course, since X is selected uniformly at random, any such restriction on just the bits of X yields the uniform distribution. All that needs to be shown is that the conditional distribution of selected output coordinates is uniform, given any value of the selected input coordinates, or, in other words, that if the adversary is allowed to look at only a small number of input bits (fewer than n^δ) than the distribution of any small number of output bits due to the remaining inputs bits is still uniform. To see this, let $R \subseteq \{1,\ldots,k\}$ and $C \subseteq \{1,\ldots,m\}$ denote the selected coordinates of $f(X)$ and X, respectively, where $|R|+|C| = n^\delta$. Letting $f(X)_R$ denote the bits of the output corresponding to R (that is to say, the selected bits of the output), and defining X_C and $X_{\overline{C}}$ analogously (which are then the selected and unselected bits of the input, respectively), then we can write $f(X)_R = F_{R,C}X_C + F_{R,\overline{C}}X_{\overline{C}}$. Since F meets the above condition, we know that $F_{R,\overline{C}}$ is full rank, and so, as all of the (unseen) bits of $X_{\overline{C}}$ vary uniformly, $F_{R,\overline{C}}X_{\overline{C}}$ varies uniformly. One way to see this is to note that, since $F_{R,\overline{C}}$ is full rank, it contains a $|R| \times |R|$ invertible submatrix. Therefore, as the bits of $X_{\overline{C}}$ that correspond to this invertible submatrix vary over all possible values (with the other bits of $X_{\overline{C}}$ fixed), $F_{R,\overline{C}}X_{\overline{C}}$ indeed varies uniformly. Therefore, for any fixed X_C, $f(X)_R$ varies uniformly, and so f has the Linial-Nisan-Braverman property.

To prove the other direction, assume that F doesn't meet the above condition. This means that, $\forall \delta > 0$, $\exists R \subseteq \{1,\ldots,k\}, C \subseteq \{1,\ldots,m\}$ with $|R| + |C| = n^\delta$ the submatrix $F_{R,\overline{C}}$ is not full rank. Again, we write $f(X)_R = F_{R,C}X_C + F_{R,\overline{C}}X_{\overline{C}}$. Since $F_{R,\overline{C}}$ is not full rank, we have, by definition, that as $X_{\overline{C}}$ varies $F_{R,\overline{C}}X_{\overline{C}}$ doesn't even hit all possible values. In fact, it must miss at least half of all values, and so $f(X)_R$ is far from uniformly randomly distributed for any fixed X_C. □

Using the above result, we are now able to prove Theorem 1, which states that for any "reasonable" choice of m and k, almost every $f \in \mathrm{Hom}(\{0,1\}^m, \{0,1\}^k)$ is AC^0-pseudorandom. For convenience, we restate the theorem here.

Theorem 1. *If $k = m^u$, for any fixed constant $u > 0$, then all but an exponentially small fraction of all $f \in Hom(\{0,1\}^m, \{0,1\}^k)$ are AC^0-pseudorandom.*

Proof. Let $P_{h,w}$ denote the probability that an $h \times w$ matrix, where $w \geq h$, with entries drawn uniformly at random from $\{0,1\}$, is full rank (that is to say, has rank h). We have the following useful bound, which follows from the fact that, in order for the matrix not to be full rank, either the first row must be identically zero, or the second row is a multiple of the first, or, in general, the i^{th} row lies in the span of the first $i-1$ rows; combining these probabilities with a union bound gives:

$$P_{h,w} \geq 1 - 2^{-w} \sum_{i=1}^{h} 2^{i-1}.$$

For any particular m, k, the probability that a randomly selected $f \in \text{Hom}(\{0,1\}^m, \{0,1\}^k)$ is AC^0-pseudorandom is, by the above theorem, given by the probability that all appropriately sized submatrices of a randomly selected $k \times m$ matrix are full rank. To be precise, we are interested in the probability that all submatrices $F_{R,\overline{C}}$, where $|R| + |C| = n^\delta$ are full rank, when $m, k \gg n^\delta$. For any $h \leq k$ and $w \leq m$, the number of $h \times k$ submatrices of a $k \times m$ matrix is given by $\binom{k}{h}\binom{m}{w}$, and so, by a simple union bound, we have the following:

$$\begin{aligned}
\Pr(f \text{ doesn't have the LNB property}) &\leq \sum_{j=1}^{n^\delta - 1} \binom{k}{j} \binom{m}{m - (n^\delta - j)} (1 - P_{j, m-(n^\delta - j)}) \\
&\leq n^\delta \binom{k}{n^\delta} \binom{m}{n^\delta} 2^{-(m-n^\delta)} \sum_{i=1}^{n^\delta} 2^{i-1} \\
&\leq n^\delta \frac{k^{n^\delta}}{(n^\delta)!} \frac{m^{n^\delta}}{(n^\delta)!} 2^{-(m-n^\delta)} (2^{n^\delta+1} - 1) \\
&\leq \frac{(km)^{n^\delta}}{2^m} \\
&= \frac{(m^{u+1})^{n^\delta}}{2^m}
\end{aligned}$$

□

2.3 Convolution

In the previous section, it was shown that many functions in $\text{Hom}(\{0,1\}^m, \{0,1\}^k)$ appear random to AC^0 circuits, but no explicit example of such a function was given. This section shows that a particular function, namely the convolution function, satisfies this property. We begin by recalling the definition of convolution. Given some $X \in \{0,1\}^r$ and $Y \in \{0,1\}^s$, the convolution of X and Y, which will be denoted $X * Y$, is the $Z \in \{0,1\}^{r+s-1}$ where if $X_i, Y_i,$ and Z_i refer to the i^{th} bit (zero indexed, counting from the least significant bit up) of X, Y, Z, respectively, then

$$Z_i = \sum_{j=0}^{i} X_j Y_{i-j},$$

where $X_j Y_{i-j}$ denotes the AND of X_j and Y_{i-j}, any X_j or Y_j outside of the defined range is understood to be zero, and the sum is, of course, computed modulo 2.

The goal is to show that convolution is AC^0-pseudorandom. There are several reasonable ways to define this. Perhaps the most natural, immediate thought is to consider the function

$f : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^{r+s-1}$, which takes the pair (X, Y) to $X * Y$. Unpacking definitions, this means we consider the distribution (when X and Y are selected uniformly at random) of the string in $\{0,1\}^{2r+2s-1}$ where the first r bits are X, the next s bits are Y, and the final $r+s-1$ bits are $X * Y$. Observe that this distribution clearly does not look random to AC^0 circuits because some of the bits of $X * Y$ can be determined exactly by an AC^0 circuit. To be precise, letting n denote, as usual, the total size of the string ($n = 2r + 2s - 1$), we see that any of the first (or last) $O(\log^c n)$ bits of $X * Y$ is simply the parity of $O(\log^c n)$ bits, each of which is the AND of some bit of X with some bit of Y. Since a parity of $O(\log^c n)$ bits can (for any constant c) be computed easily in AC^0, we immediately conclude that including any of these bits will cause the resulting distribution to not appear random to AC^0 circuits. However, if we exclude these bits, we can show that the remainder does appear random to AC^0 circuits. We consider the function $\mathrm{CONV}_{r,s,k} : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^k$ where now $k = k(r,s) < r + s - 1$, and only the k "middle bits" of $X * Y$ are included (the k centermost bits). Such a function is not a homomorphism and so the technique of the previous section does not directly apply. Instead, we will consider a variant of this problem, to which that technique does apply. Doing so yields a stronger result that also immediately implies that $\mathrm{CONV}_{r,s,k}$ function does, in fact, appear random to AC^0 circuits.

Essentially, the idea is to consider a "fixed" Y (here we mean that there is a single fixed Y of each length; as mentioned earlier, the discussion involves the asymptotic properties of f, defined by a sequence of Y values, one for each length), and define the function $f_Y : \{0,1\}^r \to \{0,1\}^k$, (where again, as above, $k = k(r) < r + s - 1$) such that f_Y takes the r-bit value X to the middle k bits of $X * Y$. The difference between these two variants can be understood as follows. In the first variant, described in the previous paragraph, the distinguisher would be an AC^0 circuit family where the circuit whose input size is $r + s + k$ would be able to distinguish the string consisting of a uniformly randomly selected $X \in \{0,1\}^r$, a uniformly randomly selected $Y \in \{0,1\}^s$ and the middle $k(r,s)$ bits of $X * Y$ from a truly random string. In the second variant, the distinguisher can have Y built-in, and only needs to distinguish the string consisting of a uniformly randomly selected $X \in \{0,1\}^r$ and the middle $k(r)$ bits of $X * Y$ from a truly random string.

Since each f_Y is clearly a homomorphism, Lemma 1 applies. Moreover, if it can be shown that, for all sufficiently large r, all but an exponentially small fraction of choices for Y produce an f_Y that is AC^0-pseudorandom, then it immediately follows that the variant of the problem described in the previous paragraph, in which both X and Y are selected uniformly at random, also is AC^0-pseudorandom. Loosely speaking, claiming that this second variant is AC^0-pseudorandom is a stronger claim because being able to have a separate circuit for each Y could conceivably give a distinguisher more power.

We now prove Theorem 2, which is restated below.

Theorem 2. *If $s = r^u$ and $k = r + s - (MIN(r,s))^\alpha$, for any fixed constants $u > 0$ and $0 < \alpha < 1$, then $CONV_{r,s,k}$ is AC^0-pseudorandom. In particular, if $r = s$ and $k = 2r - r^\alpha$, for any $0 < \alpha < 1$, then $CONV_{r,s,k}$ is AC^0-pseudorandom.*

By the above logic, it suffices to show the following lemma.

Lemma 2. *For all but an exponentially small fraction of Y, the function $f_Y : \{0,1\}^r \to \{0,1\}^k$, where $k = 2(r - r^\alpha + 1)$ for any small constant $\alpha > 0$, has the Linial-Nisan-Braverman property.*

Proof. Let f denote an arbitrary element of the set $\{f_Y | Y \in \{0,1\}^s\}$. Since f is a homomorphism, there is a corresponding $k \times r$ matrix F such that $f(X) = FX$, for any $X \in \{0,1\}^r$. To show that, for almost all choices of Y, the corresponding function f has the Linial-Nisan-Braverman property, it suffices, by Lemma 1, to show that the appropriate submatrices of F are full rank.

The matrix F has a particularly simple structure, namely it has constant skew-diagonals. That is to say, if $F_{i,j}$ denotes the element of F in row i and column j then $F_{i,j} = F_{i-1,j+1}$. The first row of F consists of, from left to right, $r - r^\alpha$ zeros followed by the lowest r^α bits of Y, starting with the least significant bit of Y. Each subsequent row of F is obtained by shifting Y one index further to the left, filling empty entries with zeros. Consider an arbitrary submatrix $F_{R,\overline{C}}$ where $R \subseteq \{1, \ldots, k\}$ and $C \subseteq \{1, \ldots, r\}$ such that $|R| + |C| = n^\delta$ for $\delta < \alpha$, where $\overline{C} = \{1, \ldots, r\} \setminus C$ and $n = r + k$. For randomly selected Y, this submatrix is full rank with overwhelming probability. To see this, note that if $F_{R,\overline{C}}$ is not full rank, then there is some non-trivial linear combination of its rows that adds to 0. Let h and w be the height and width, respectively, of $F_{R,\overline{C}}$. Then there are $2^h - 1$ potential non-trivial linear combinations of the rows, because a linear combination is, by definition, a sum of the rows of $F_{R,\overline{C}}$ where each row has coefficient 0 or 1 (having all coefficients be 0 is the trivial linear combination). In other words, it is a sum of some subset of the rows of $F_{R,\overline{C}}$. Consider any fixed non-trivial linear combination. Let i denote the lowest row of $F_{R,\overline{C}}$ that has coefficient 1. Note that the probability (over Y) that this particular linear combination of the rows of $F_{R,\overline{C}}$ is zero is very small. While this fact would be immediate if $F_{R,\overline{C}}$ were simply a random unstructured matrix, some care must be given due to the structure of F (constant skew-diagonals) which forces all elements of F in the same skew-diagonal to be identical. To deal with this, consider the rows of $F_{R,\overline{C}}$ one at a time, from left to right. In order for the linear combination of the rows to be the zero vector, it must be the case, by definition, that the sum in each column is zero (where of course this sum is only over the subset of elements selected by the linear combination). Consider the element in position (i, j). This element is either some element of Y, if some part of Y was shifted over position (i, j), or is simply 0, if no part of Y was shifted to that position. In the first case, this value is completely independent of any previously considered entries that influence the linear combination. This is because, even though the value of the entry in position (i, j) forces the values of all other entries in the same skew-diagonal (in F), all other such entries are either to the right of this entry, and so haven't been considered yet, or to the left and below this entry, in which case they have coefficient 0 in the linear combination (because row i is the lowest row with coefficient 1). Since row i has coefficient 1, flipping the value of the element in position (i, j) flips the value of the sum in column j, and so the sum in this column is 0 with probability $\frac{1}{2}$. From this, we immediately conclude that the probability that the sum in all columns is 0 is 2^{-z}, where z is the number of entries in row i that come from Y (as opposed to being fixed 0s). Since each row of F has at least r^α such elements (because the output of f does not include the first or last r^α bits of $X * Y$), we conclude that this particular linear combination is 0 with probability at most 2^{-r^α}. Applying a union bound over all $2^h - 1$ non-trivial linear combinations, where $h < n^\delta \ll r^\alpha$, and then another union bound over all choices of R and C (as in the calculation in the previous section), we conclude that, for all but an exponentially small fraction of Y, F has the desired property, which completes the proof that convolution appears random to AC^0 circuits.

□

2.4 Integer Multiplication

Let $\text{MULT}_{r,s,k} : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^k$ denote the integer multiplication function, which takes a $X \in \{0,1\}^r$ and $Y \in \{0,1\}^s$ to the middle k-bits of the $r + s$ bit long product of X and Y. In this section, we will prove the following theorem.

Theorem 3. *If $s = r^u$ and $k = r + s - (MIN(r, s))^\alpha$, for any fixed constants $u > 0$ and $0 < \alpha < 1$, then $MULT_{r,s,k}$ is AC^0-pseudorandom. In particular, if $r = s$ and $k = 2r - r^\alpha$, for any $0 < \alpha < 1$, then $MULT_{r,s,k}$ is AC^0-pseudorandom.*

As was the case for convolution, there are two natural variants of the multiplication problem to consider. In the first variant, we select $X \in \{0,1\}^r$ and $Y \in \{0,1\}^s$ uniformly at random, then produce the product $P = X \times Y$, and finally we produce the string consisting of X, Y, and part of P. The hope is that the distribution of that string appears random to AC^0 circuits. It is necessary to include only part of P because, as was the case in convolution, the lowest and highest bits of P do not look random to AC^0 circuits. For example, the low $O(\log^c r)$ bits of the product can be calculated exactly, using the technique in [CSV84]. In the second variant, we consider "fixed" Y, in the sense that we have a single Y of each length, and the multiplication problem is defined such that a uniformly randomly selected $X \in \{0,1\}^r$ is multiplied by the fixed Y to produce the product $P = X \times Y$; the string of interest then consists of X and the middle part of P. Again, loosely speaking, the second variant is stronger in the sense that a potential distinguisher is allowed to have Y built-in.

In this section, we focus on the second variant and show that, for sufficiently large r, all but an exponentially small fraction of Y (of length s) lead to a multiplication problem that looks random to AC^0 circuits. Therefore, by the same logic as in the convolution problem, it immediately follows that the first variant is also AC^0-pseudorandom. We consider the function $f_Y : \{0,1\}^r \to \{0,1\}^k$, which takes the r-bit value X to the middle k bits of the product $X \times Y$. We will prove the following lemma, from which the above theorem immediately follows.

Lemma 3. *For all but an exponentially small fraction of $Y \in \{0,1\}^s$, where $s = r^u$, the function $f_Y : \{0,1\}^r \to \{0,1\}^k$, where $k = r + s - 2r^\alpha$ for any small constant $\alpha > 0$, has the Linial-Nisan-Braverman property.*

Proof. It suffices to establish the claim for almost all odd Y (because adding w trailing zeros to Y simply shifts the product $X \times Y$ by w bits to the left; all but an exponentially small fraction of Y have fewer than r^α trailing zeros), and so we restrict our attention to the case in which Y is odd. We begin by establishing some notation. Let $n = r + k$. Let $Z = Z_1 \cdots Z_n$ be the distribution of the set of all strings of the form $X \circ f_Y(X)$ (strings that are the concatenation of X with $f_Y(X)$), where X is an r-bit string. Then, by definition, f_Y has the Linial-Nisan-Braverman property if Z is a $(2^{-n^\gamma}, n^\delta)$-approximation for appropriate small constants $0 < \delta < \gamma < 1$, which is to say that, for every set of n^δ coordinates the restriction of μ_n to those coordinates is 2^{-n^γ}-close to the uniform distribution over $\{0,1\}^{n^\delta}$. To show this, we begin by recalling that the bias of a distribution Z on some set $I \subseteq \{1, \ldots, n\}$ is defined to be

$$\text{bias}_I(Z) = \text{E}[(-1)^{\sum_{i \in I} Z_i}].$$

We make use of the following lemma, variants of which appeared in, for example [Vaz86] and [AGM02].

Lemma 4. *[Vaz86], [AGM02] Every distribution Z that has bias as most ϵ on every non-empty subset I of size at most h is a $(2^{h/2}\epsilon, h)$-approximation.*

We will then show that Z has bias at most 2^{-n^ν}, for some constant $\nu > 0$, on all non-empty sets of size at most n^δ. The above lemma implies that Z is a $(2^{-n^\gamma}, n^\delta)$-approximation, as desired (for any $\delta < \gamma < \nu$). To see why, let X_i denote the i^{th} bit of X and let $f_{Y,j} : \{0,1\}^r \to \{-1,1\}$ be defined such that $f_{Y,j}(X) = 1$ when the j^{th} bit of $X \times Y$ is 0 and $f_{Y,j}(X) = -1$ when the j^{th} bit of $X \times Y$ is 1 (note that $f_{Y,j}$ corresponds to the j^{th} bit of $X \times Y$ not the j^{th} bit of $f_Y(X)$, where $f_Y(X)$ consists of all bits of $X \times Y$ except the lowest and highest r^α; this is done because it will be much cleaner to refer to bits by their position in the entire product). Clearly,

$$f_{Y,j}(X) = (-1)^{\lfloor \frac{XY}{2^{j-1}} \rfloor}.$$

For any $S \subseteq \{1, \ldots, r\}$, let $\hat{f}_{Y,j}(S)$ denote the Fourier-Walsh coefficients of $f_{Y,j}$, which are given by

$$\hat{f}_{Y,j}(S) = \mathrm{E}[f_{Y,j}(X)(-1)^{\sum_{i \in S} X_i}].$$

These are the Fourier coefficients of a function on $\mathbb{F}_2^r$ (we use the term Fourier-Walsh to avoid confusion with the "ordinary" Fourier coefficients of a function defined on $\mathbb{R}$, which will be used shortly). We partition the set I as $I = S \cup J$, where $S \subseteq \{1, \ldots, r\}$ are the indices of Z that correspond to bits of X and $J \subseteq \{r+1, \ldots, n\}$ are the indices of Z that correspond to bits of $f_Y(X)$.

There are two cases. First, if J is empty, then the set I consists only of bits of X, and so, trivially, Z has bias exactly 0 on this set, because X is uniformly random. The interesting case is when J is non-empty. For notational convenience, define the set $J' \subseteq \{r^\alpha + 1, \ldots, r + s - r^\alpha\}$ such that $J' = \{j' | j' + r - r^\alpha \in J\}$ (simply the set J shifted appropriately to index bits of $X \times Y$). Let $f_{Y,J'}(X) = \prod_{j \in J'} f_{Y,j}(X)$. Then the bias of Z on I is simply $\hat{f}_{Y,J'}(S)$. This follows from the fact that

$$\begin{aligned}
\mathrm{bias}_I(Z) &= \mathrm{E}[(-1)^{\sum_{i \in I} Z_i}] \\
&= \Pr[\oplus_{i \in I} Z_i = 0] - \Pr[\oplus_{i \in I} Z_i = 1] \\
&= \Pr[\oplus_{s \in S} Z_s = \oplus_{j \in J} Z_j] - \Pr[\oplus_{s \in S} Z_s \neq \oplus_{j \in J} Z_j] \\
&= \Pr[(-1)^{\sum_{s \in S} X_s} = f_{Y,J'}(X)] - \Pr[(-1)^{\sum_{s \in S} X_s} \neq f_{Y,J'}(X)] \\
&= \Pr[(-1)^{\sum_{s \in S} X_s} f_{Y,J'}(X) = 1] - \Pr[(-1)^{\sum_{s \in S} X_s} f_{Y,J'}(X) = -1] \\
&= \mathrm{E}[f_{Y,J'}(X)(-1)^{\sum_{s \in S} X_s}] \\
&= \hat{f}_{Y,J'}(S).
\end{aligned}$$

Rather than compute $\hat{f}_{Y,J'}(S)$ directly, we instead compute the Fourier coefficients of $f_{Y,J'}$ when viewed as a function on $\{0, \ldots, 2^r - 1\}$ (instead of on $\mathbb{F}_2^r$), and exploit a connection between these two types of Fourier coefficients. For a function $f : \{0, \ldots, 2^r - 1\} \to \{-1, 1\}$, define

$$\hat{f}(k) = \mathrm{E}[f(t) e^{-\frac{2\pi i k t}{2^r}}],$$

where $k \in \mathbb{Z}$. We have the following lemma, from [Gre12] (see also [Kat86]), which has been modified to fit our notation. We say that an integer k is a (b, m)-sparse number if it can be written in the form $k = k_1 2^{h_1} + \cdots + k_b 2^{h_b}$ where each $k_i \in \mathbb{Z}$, $|k_i| \leq m$, $h_i \in \mathbb{N}$.

Lemma 5. *Let $f : \{0, \ldots, 2^r - 1\} \to \{-1, 1\}$ be a function such that $\exists S \subseteq \{1, \ldots, r\}$ with Fourier-Walsh coefficient $\hat{f}(S)$ of magnitude at least ϵ, where $0 < \epsilon < \frac{1}{2}$. Then there is a $\left(|S|, \left(\frac{10|S|}{\epsilon}\right)\right)$-sparse number k such that the Fourier coefficient $\hat{f}(k)$ has magnitude at least $\left(\frac{\epsilon}{10|S|}\right)^{4|S|}$.*

Applying this lemma to the function $f_{Y,J'}$, with sets S of size at most n^δ, we immediately conclude that, in order to establish the necessary bounds on the Fourier-Walsh coefficients (which then implies that multiplication has the Linial-Nisan-Braverman property), it suffices to show that, for all $(n^\delta, 10n^\delta 2^{n^\nu})$-sparse numbers k, $|\hat{f}_{Y,J'}(k)| < 2^{-n^\rho}$ for a fixed constant ρ such that $\rho > \delta + \nu$. We say that a particular Fourier component is negligible if its magnitude has such a bound.

We now show that, for almost all Y, the required bound on $\hat{f}_{Y,J'}(k)$ holds. The main idea is that, for each j, $f_{Y,j}$ is simply a downsampled version of a square wave. This fact allows us to express the Fourier coefficients of $f_{Y,j}$ in terms of the Fourier coefficients of a square wave. This

is useful because the Fourier coefficients of a square wave are particularly simple. In the following, we make use of several standard facts about the Discrete Fourier Transform, which can be found in essentially any text that deal with Fourier Analysis, for example [OSB99]. We begin with a few definitions. Let $D_Y = \{0, \ldots, Y2^{r+s} - 1\}$. Let $s_j : D_Y \to \{-1, 1\}$ be the perfect square wave of period 2^j,

$$s_j(t) = (-1)^{\lfloor \frac{t}{2^{j-1}} \rfloor}.$$

Let $p_Y : D_Y \to \{0, 1\}$ be a pulse train with interval Y,

$$p_Y(t) = \begin{cases} 1, & t \equiv 0 \mod Y \\ 0, & t \not\equiv 0 \mod Y \end{cases}.$$

Let $h_Y(t) : D_Y \to \{0, 1\}$ be the step function

$$h_Y(t) = \begin{cases} 1, & t < Y2^r \\ 0, & t \geq Y2^r \end{cases}.$$

Finally, let $g_{Y,J'}(t) = Y2^s h_Y(t) p_Y(t) \prod_{j \in J'} s_j(t)$.

We then have

$$\begin{aligned}
\hat{f}_{Y,J'}(k) &= \frac{1}{2^r} \sum_{t=0}^{2^r-1} f_{Y,J'}(t) e^{-\frac{2\pi i k t}{2^r}} \\
&= \frac{1}{2^r} \sum_{t=0}^{2^r-1} \left(\prod_{j \in J'} (-1)^{\lfloor \frac{Yt}{2^{j-1}} \rfloor} \right) e^{-\frac{2\pi i k t}{2^r}} \\
&= \frac{1}{2^r} \sum_{t=0}^{Y2^r-1} p_Y(t) \left(\prod_{j \in J'} (-1)^{\lfloor \frac{t}{2^{j-1}} \rfloor} \right) e^{-\frac{2\pi i k t}{Y2^r}} \\
&= \frac{1}{2^r} \sum_{t=0}^{Y2^r-1} p_Y(t) \left(\prod_{j \in J'} s_j(t) \right) e^{-\frac{2\pi i k t}{Y2^r}} \\
&= \frac{1}{Y2^{r+s}} \sum_{t=0}^{Y2^{r+s}-1} Y2^s h_Y(t) p_Y(t) \left(\prod_{j \in J'} s_j(t) \right) e^{-\frac{2\pi i 2^s k t}{Y2^{r+s}}} \\
&= \frac{1}{Y2^{r+s}} \sum_{t=0}^{Y2^{r+s}-1} g_{Y,J'}(t) e^{-\frac{2\pi i 2^s k t}{Y2^{r+s}}} \\
&= \hat{g}_{Y,J'}(2^s k).
\end{aligned}$$

Therefore, it suffices to show that $\hat{g}_{Y,J'}(2^s k)$ is sufficiently small for the k values of interest. The convolution theorem implies that

$$\hat{g}_{Y,J'}(k) = Y2^s \hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \bigotimes_{j \in J'} \hat{s}_j(k),$$

where $\otimes$ denotes cyclic convolution.

Notice that, for each j, $\hat{s}_j(k)$ has a particularly simply structure.

$$\hat{s}_j(k) = \begin{cases} \frac{1}{2^{j-2}\left(1-e^{-\frac{2\pi i(2v+1)}{2^j}}\right)}, & k = (2v+1)Y2^{r+s-j} \\ 0, & \text{otherwise} \end{cases}.$$

Notice that $\hat{s}_j(k)$ is only nonzero at few locations; specifically, the odd multiples of $Y2^{r+s-j}$. Moreover, notice that the magnitude of the nonzero values falls off quickly. To be precise,

$$\sum_{\substack{v \\ |2v+1|>2^{n^\tau}}} |\hat{s}_j((2v+1)Y2^{r+s-j})| = O(2^{-n^\eta})$$

for constants η and τ such that $\delta < \eta < \tau \ll 1$. In other words, the only non-negligible part of $\hat{s}_j(k)$ is at values k given by small odd multiples of a shift of Y (Y shifted to the left by $r+s-j$ bits).

We then consider $\bigotimes_{j\in J'} \hat{s}_j(k)$. We split $\hat{s}_j(k)$ into a large low frequency component and a small high frequency component. That is to say, we write $\hat{s}_j(k) = \hat{u}_j(k) + \hat{v}_j(k)$, where

$$\hat{u}_j(k) = \begin{cases} \frac{1}{2^{j-2}\left(1-e^{-\frac{2\pi i(2v+1)}{2^j}}\right)}, & k = (2v+1)Y2^{r+s-j}, |2v+1| \le 2^{n^\tau} \\ 0, & \text{otherwise} \end{cases}$$

and

$$\hat{v}_j(k) = \begin{cases} \frac{1}{2^{j-2}\left(1-e^{-\frac{2\pi i(2v+1)}{2^j}}\right)}, & k = (2v+1)Y2^{r+s-j}, |2v+1| > 2^{n^\tau} \\ 0, & \text{otherwise} \end{cases}.$$

Therefore,

$$\bigotimes_{j\in J'} \hat{s}_j(k) = \bigotimes_{j\in J'} (\hat{u}_j(k) + \hat{v}_j(k))$$

$$= \sum_{\substack{J_1, J_2 \\ J_1 \cup J_2 = J'}} \left(\bigotimes_{j\in J_1} \hat{u}_j(k)\right) \otimes \left(\bigotimes_{j\in J_2} \hat{v}_j(k)\right).$$

Notice that there are at most 2^{n^δ} terms in the above expansion (because $|J'| \le n^\delta$). The term $\bigotimes_{j\in J'} \hat{u}_j(k)$ is only nonzero at k values of the form $(2v_1+1)Y2^{r+s-j_1} + \cdots + (2v_{|J'|}+1)Y2^{r+s-j_{|J'|}}$, where each v_i satisfies $|2v_i+1| \le 2^{n^\tau}$. All other terms are extremely small everywhere. To be precise, when $J_1 \neq J'$, every such term involves at least one $\hat{v}_j(k)$ factor and so we can write $\left(\bigotimes_{j\in J_1} \hat{u}_j(k)\right) \otimes \left(\bigotimes_{j\in J_2} \hat{v}_j(k)\right) = \hat{v}_j(k) \otimes \hat{q}(k)$ for some function $q : D_Y \to \{-1, 0, 1\}$. By combining the bound $\sum_k |\hat{v}_j(k)| = O(2^{-n^\eta})$ with the trivial bound $|\hat{q}(k)| \le 1$, we obtain $|\left(\bigotimes_{j\in J_1} \hat{u}_j(k)\right) \otimes \left(\bigotimes_{j\in J_2} \hat{v}_j(k)\right)| = O(2^{-n^\eta})$. Therefore, the total contribution of all terms except $\bigotimes_{j\in J'} \hat{u}_j(k)$ is negligible ($O(2^{-(n^\eta - n^\delta)})$). From the above, it is immediate that the only non-negligible Fourier components are values k of the form $(2v_1+1)Y2^{r+s-j_1} + \cdots + (2v_{|J'|}+1)Y2^{r+s-j_{|J'|}}$, where each v_i satisfies $|2v_i+1| \le 2^{n^\tau}$. Recall that each j satisfies $r^\alpha < j < r+s-r^\alpha$. Therefore, these values k are of the form Yk', where k' is a $(|J'|, 2^{n^\tau})$-sparse number with at least r^α trailing zeros and at most $r+s-r^\alpha$ trailing zeros.

Next, we consider $\hat{g}_{Y,J'}(k)$. We have

$$\hat{g}_{Y,J'}(k) = Y2^s\hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \bigotimes_{j\in J'} \hat{s}_j(k)$$

$$= Y2^s\hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \bigotimes_{j\in J'} (\hat{u}_j(k) + \hat{v}_j(k))$$

$$= Y2^s\hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \sum_{\substack{J_1,J_2 \\ J_1\cup J_2 = J'}} \left(\bigotimes_{j\in J_1} \hat{u}_j(k)\right) \otimes \left(\bigotimes_{j\in J_2} \hat{v}_j(k)\right)$$

$$= \sum_{\substack{J_1,J_2 \\ J_1\cup J_2 = J'}} Y2^s\hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \left(\bigotimes_{j\in J_1} \hat{u}_j(k)\right) \otimes \left(\bigotimes_{j\in J_2} \hat{v}_j(k)\right).$$

By the same logic as above, the total contribution of every term in the sum except the $J_1 = J'$ term is negligible everywhere (has total magnitude $O(2^{-(n^\eta - n^\delta)})$ at all k) and so if we define the function $\hat{g'}_{Y,J'}(k) = Y2^s\hat{h}_Y(k) \otimes \hat{p}_Y(k) \otimes \bigotimes_{j\in J'} \hat{u}_j(k)$, it suffices to show that $\hat{g'}_{Y,J'}$ is small at the k values of interest.

We have

$$\hat{p}_Y(k) = \begin{cases} 1, & k = u2^{r+s} \\ 0, & \text{otherwise} \end{cases}$$

and

$$Y2^s\hat{h}_Y(k) = \frac{1}{2^r}\frac{1 - e^{-\frac{2\pi ik}{2^s}}}{1 - e^{-\frac{2\pi ik}{Y2^{r+s}}}}.$$

Therefore, the only non-negligible values of $\hat{g'}_{Y,J'}(k)$ are those that are "close" to values of the form $Yk' \mod 2^{r+s}$. More precisely, the only non-negligible values of $\hat{g'}_{Y,J'}$ are of the form $k \equiv Yk' + u \mod 2^{r+s}$, where $|u| \leq 2^{s+n^\nu}$, and so the only non-negligible values of $\hat{f}_{Y,J'}(k) = \hat{g}_{Y,J'}(2^sk)$ are at values k such that $2^sk \equiv Yk' + u \mod 2^{r+s}$. Or equivalently, values k where $\exists k', u'$ where k' is (as above) a $(|J'|, 2^{n^\tau})$-spare number with at least r^α trailing zeros and at most $r + s - r^\alpha$ trailing zeros, $|u'| \leq 2^{n^\nu}$ such that $k + u'$ is equal to the high 2^r bits of $Yk' \mod 2^{r+s}$.

Therefore, for a particular value Y, the required bound on $|\hat{f}_{Y,J'}(k)|$ holds if, for every $(n^\delta, 10n^\delta 2^{n^\nu})$-sparse number k, we do not have $k + u'$ equal to the high 2^r bits of $Yk' \mod 2^{r+s}$, for any $(|J'|, 2^{n^\tau})$-sparse number k' with at least r^α trailing zeros and at most $r + s - r^\alpha$ trailing zeros. To see that this holds for all but an exponentially small fraction of Y, first notice that if k is a $(n^\delta, 10n^\delta 2^{n^\nu})$-sparse number, then $k+u'$ is a $(n^\delta+1, 10n^\delta 2^{n^\nu})$-sparse number. Set the constants τ and ν small enough such that $n^{\nu+\tau} \ll r^\alpha$ (this can be done because $n = 2r+s-2r^\alpha = 2r+r^u-2r^\alpha$ and so n is polynomial in r). Therefore, it suffices to show that, for almost all Y, if k' is a $(|J'|, 2^{n^\tau})$-sparse number with at least r^α trailing zeros and at most $r + s - r^\alpha$ trailing zeros then the high 2^r bits of $Yk' \mod 2^{r+s}$ is not a $(n^\delta + 1, 10n^\delta 2^{n^\nu})$-sparse number. To see this, notice that, for each pair of sparse numbers k, k'', there is at most a fraction $\frac{1}{2^{r^\alpha}}$ of all Y such that the high 2^r bits of $Yk' \mod 2^{r+s}$ are equal to k'' and so a simple union bound completes the proof.

□

2.5 Matrix Multiplication

We now show that matrix multiplication is AC^0-pseudorandom. Let MATRIX-MULT$_{r,s} : \{0,1\}^{rs} \times \{0,1\}^{rs} \to \{0,1\}^{s^2}$ denote the matrix multiplication function, which, on input a $s \times r$ matrix A and a $r \times s$ matrix B (both of which are encoded as strings in $\{0,1\}^{rs}$ in the obvious way), produces the $s \times s$ matrix AB.

Theorem 4. *If $s = r^u$, for any fixed constant $u > 0$, then MATRIX-MULT$_{r,s}$ is AC^0-pseudorandom.*

As was the case for the convolution and multiplication problems, we consider a stronger variant where one of the matrices is held fixed. We then prove the following lemma, from which the above theorem immediately follows.

Lemma 6. *For an $s \times r$ matrix A, let $f_A : \{0,1\}^{rs} \to \{0,1\}^{s^2}$ denote the function that, on input a $r \times s$ matrix B produces the $s \times s$ matrix $Z = AB$. Then all but an exponentially small fraction of A yield an f_A that is AC^0-pseudorandom.*

Proof. To see that almost all such f_A are AC^0-pseudorandom, let B_i and Z_i denote the i^{th} column of B and Z, respectively. Then, of course, $Z_i = AB_i$, and so we can interpret this problem as the concatenation of s independent instances of the homomorphism problem. That is to say, if we let $f'_A : \{0,1\}^r \to \{0,1\}^s$ be the homomorphism corresponding to A, then $Z_i = f'(B_i)$. The result then follows from Theorem 1.

□

3 The Reduction Technique

3.1 Next-Bit Test and Parity

In this section, another technique for proving that a function appears random to AC^0 circuits is presented, specifically, reducing a known hard problem to the next-bit test. The next-bit test is defined as follows. Given a distribution μ_n with support $\{0,1\}^n$, we say that μ_n passes the next-bit test if, given the first i bits of a string selected according to μ_n, no AC^0 circuit can predict the $(i+1)^{\text{th}}$ bit with non-negligible advantage, for any i. Formally, for any $Z \in \{0,1\}^n$, let Z_j denote the j^{th} bit of Z (1 indexed, counting from left to right) and $Z_{[j,k]}$ denote the substring of Z from positions j to k, inclusive. Then we say that μ_n passes the next-bit test if, for all $i \in \{1,\ldots,n\}$, and for all functions $Q_i : \{0,1\}^{i-1} \to \{0,1\}$ computable by AC^0 circuits, $|\Pr(Q_i(Z_{[1,i-1]}) = x_i) - \frac{1}{2}| = O(2^{-n^\kappa})$, for some constant $\kappa > 0$, where the probability is taken over values of $Z \in \{0,1\}^n$ drawn according to the distribution μ_n. It is known [Yao82] that a distribution μ_n passes the next-bit test if and only if μ_n $O(2^{-n^\kappa})$-fools all AC^0 circuits (strictly speaking, the result in [Yao82] was proven for probabilistic polynomial time algorithms, but the same technique applies just as well to AC^0 circuit families). Since, as stated in §1, we say that a function f is AC^0-pseudorandom if the distribution μ_n corresponding to it $O(2^{-n^\kappa})$-fools all AC^0 circuits, showing that μ_n passes the next-bit test is sufficient to prove the corresponding f is AC^0-pseudorandom.

The natural next question is how to prove that distributions arising from particular functions pass the next-bit test. One idea is to reduce a problem that is known to be hard for AC^0, such as the parity problem, to the next-bit test. The parity problem is defined as follows: given some $X \in \{0,1\}^*$, compute $\sum_i X_i \mod 2$. In other words, the parity of a string is 1 if there are an odd number of 1s in the string and 0 if there are an even number of 1s in the string. It is known that no AC^0 circuit family can compute parity [FSS84],[Ajt83]. In fact, parity can't even be non-negligibly approximated in AC^0 [Has86]. To be precise, if we define $h(s,d,n)$ to be the function such that no

depth d circuit of size 2^s computes parity correctly for more than a $\frac{1}{2} + h(s,d,n)$ fraction of the inputs, then we have the following (Theorem 8.1.iii in [Has86])

Theorem. *[Has86]* $h(s,d,n) < 2^{-\Omega((\frac{n}{s})^{\frac{1}{d-1}})}$ *for* $d > 2$ *and* $s < n^{\frac{1}{d}}$.

The goal is then to reduce the parity problem to the problem of computing the next bit of a string drawn according to μ_n, or, in other words, show that if some AC^0 circuit could predict the next bit with non-negligible advantage, then it could be used to produce another AC^0 circuit that approximates the parity problem, with non-negligible advantage. Since the parity problem cannot be approximated by such a circuit, we could then conclude that the original distribution must pass the next-bit test.

3.2 Integer Multiplication

As was already shown in Theorem 3, the function $\text{MULT}_{r,s,k}$ is AC^0-pseudorandom when $s = r^u$ and $k = r + s - (\text{MIN}(r,s))^\alpha$, for constants $u > 0$ and $0 < \alpha < 1$. This was done by considering a variant of the multiplication function in which one of the multiplicands is held fixed. Specifically, for $Y \in \{0,1\}^s$, we defined the function $f_Y : \{0,1\}^r \to \{0,1\}^k$ which takes a value $X \in \{0,1\}^r$ to the middle k bits of $X \times Y$. As shown in Lemma 3, f_Y is AC^0-pseudorandom for all but an exponentially small fraction of Y, when $s = r^u$ and $k = r + s - (\text{MIN}(r,s))^\alpha$. In this section, we will be interested in results that hold when s is much greater than r. Specifically, we are interested in the case when $s > r^u$ for all constants $u > 0$, but $r > \log^c s$ for all constants $c > 0$. Recall that we say a given function looks random to AC^0 circuits if the distribution corresponding to it can only be distinguished (by AC^0 circuits) from the uniform distribution with advantage $O(2^{-n^\kappa})$. In this section we relax this condition only slightly, and only require a bound on the advantage of the form $o(2^{-\log^c n})$ for all constants $c > 0$ (in other words, we require that no AC^0 circuit can distinguish with advantage one over any quasipolynomial in n). We show that, for certain Y, f_Y is AC^0 pseudorandom with these parameters. This has several interesting consequences. Firstly, this yields a simple, multiplication based pseudorandom generator with the same stretch and security parameters as the Nisan-Wigderson generator [Nis91]. Secondly, this shows that the result in [CSV84], which states that an AC^0 circuit can multiply an n-bit value Y by a $O(\log^c n)$ bit value X is tight,

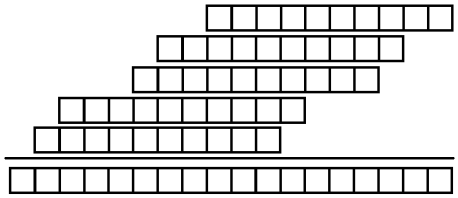
We restrict our attention to $Y \in \{0,1\}^s$ that are "sparse", in the sense that only a small number of the bits of Y are 1s. Specifically, we generate Y as follows: each bit is set to be 1 with probability $r^{-\epsilon}$, for a constant $0 < \epsilon < \frac{1}{2}$. As before, let $f_Y : \{0,1\}^r \to \{0,1\}^k$ be defined such that f_Y takes the value X to the middle k bits of the product $X \times Y$, where here $k = r + s - 2r^{2\epsilon}$. We prove the following theorem.

Theorem 5. *With high probability (where the probability is over the selection of* Y *according to the above distribution, and the statement high probability means within an exponentially small distance from probability* 1*),* f_Y *is* AC^0*-pseudorandom.*

Proof. As usual, we consider strings of the form $X \circ f_Y(X)$. For convenience, we assume that both X and the substring of $Z = X \times Y$ produced by f_Y are written from least significant bit to most significant bit, when read from left to right. We let n denote the total length of the string, and so $n = 2r + s - 2r^{2\epsilon}$. Consider the next-bit test applied to strings generated in this manner. Since the first r bits of the string are bits of the uniformly randomly generated number X, we conclude, for information theoretic reasons, that there is no hope of any AC^0 circuit predicting the i^{th} bit, given the first $i-1$ bits, for $i \in \{1,\ldots,r\}$. All that remains is to prove the same claim for

$i \in \{r+1, \ldots, n\}$, which will be done by showing that any AC^0 circuit that predicts such a bit with non-negligible advantage can be used to approximate the parity function, with non-negligible advantage, which we know is impossible. We assume, for contradiction, that we have an AC^0 circuit, call it C, that can predict some next-bit of our pseudorandom string, call it bit i, given the first $i-1$ bits. Using the circuit C, we will produce an AC^0 circuit D that predicts (with non-negligible advantage) the solution to a parity problem T of size r^ν, for some $\nu > 0$, which is impossible.

Begin by noting that, if Y_j denotes the j^{th} bit of Y (0 indexed, counting from least significant bit up), then we have $X \times Y = X \sum_{j=0}^{r-1} Y_j 2^j = \sum_{j=0}^{r-1} XY_j 2^j$. Thus, we can understand the multiplication of X by Y as the sum of many shifts of X, where the amount that X is shifted in determined by the locations of the 1s in Y. To be precise, for each j such that $Y_j = 1$, we include a copy of X shifted left by j indices. To produce the product $X \times Y$, we then sum all copies of X. This is illustrated in the figure below.



Each column contains certain bits of X. One way to characterize which bits appear in each particular column is to imagine sliding the strings X and Y^{REV} past one another, where Y^{REV} is the string Y flipped left-to-right. To be precise, start by aligning X and Y^{REV} such that the least significant bits of X and Y line up, and no other bits initially line up. To determine which bits of X lie in column j (where we number the columns from right to left, starting with 0), slide Y^{REV} j bits over; exactly the bits of X that lines up with a 1 in Y appear in column j. This is illustrated in the figure below.

Define sets $U_j \subseteq \{0, \ldots, r-1\}$ such that U_j consists of all indices of X that appear in column j. Let $S_j \subseteq \{0, \ldots, s-1\}$ be a collection of indices of Y. The exact manner in which the S_j are selected will be specified shortly. Let $V_j \subseteq U_j$ be indices of X that appear in column j because they lined up with a 1 in Y at one of the indices S_j. As noted above, we must have $i \in \{r+1, \ldots, n\}$ (the portion of the string containing bits of the product $Z = X \times Y$), and so we are predicting bit $i - r + r^{2\epsilon} - 1 =: k$ of the product. Notice that, if it weren't for the fact that there are carries when computing the sum of the various shifts of X, bit k of the product would simply be the parity of the bits of X selected by U_k. The key idea will be to construct the sets V_k so that they are individually large, $|V_k| > \log^c s$, for all constants $c > 0$, but have small intersection with any U_j, $|V_k \cap U_j| \leq 2$, $\forall j < k$, and then fill the bits of X specified by V_k with the bits of an instance of the parity problem. This is very similar to the notion of a combinatorial design, [Nis91], with the exception of the fact that here we consider subsets V_j of U_j.

The circuit D predicts the solution of the parity problem T by producing a multiplication instance to feed to C, that is to say the first $i-1$ bits of a string produced by multiplication. This string consists of a value X and some of the bits of the product XY. We construct this multiplication instance as follows. Begin by setting the bits of X selected by V_k to the bits of the parity instance T. To set the other bits of X, notice that if C can truly predict the next-bit test

with non-negligible advantage, then this means, by definition, that the advantage of C, averaged over all choices of X, is non-negligible. In particular, this means that there must exist at least one setting of the other bits of X such that C has non-negligible advantage as just the bits selected by V_k vary (uniformly). We then set the other bits of X to such a fixed value. To be clear, the claim is not that an AC^0 circuit can find a proper setting to the other bits of X, but rather that such a value can simply be built into D (because it is only a single fixed value, which depends only on the input size t of circuit D). In order to calculate the lowest $k-1$ bits of XY that must be fed to C, we write $X = X_{input} + X_{fixed}$ where X_{input} consists of the t bits of the input to D, which are assigned to the positions specified by V_k, as X_{fixed} corresponds to the fixed setting of the other bits of X. Since both Y and X_{fixed} are fixed values, we can also build the value YX_{fixed} into D. Therefore, if it were possible to compute in AC^0 the low $k-1$ bits of YX_{input}, then it would be possible to compute the low $k-1$ bits of XY because $XY = YX_{input} + YX_{fixed}$, and we can, of course, perform addition in AC^0. The key observation is that, with high probability over the choice of Y, it will be easy to compute YX_{input}.

To see this, notice that, with high probability over Y, there will be a choice of S_k such that $|V_k \cap U_j| \leq 2$, for $j \in \{0, \ldots, k-1\}$. This is simply the statement that each column of multiplication problem illustrated in the figure above contains at most two bits of X_{input}. Therefore, these bits can be packed into two numbers, whose sum (which is calculable in AC^0) will be the low bits of YX_{input}. To see that $|V_k \cap U_j| \leq 2$, with high probability, let Y' be identical to Y except that all bits outside of S_k are set to 0, and note that $|V_k \cap U_j|$ is simply the number of 1s that line up when Y and Y' are slid over one another, or, in other words, the number of h such that Y'_h and $Y_{h-(k-j)}$ are both 1. To bound the probability that $|V_k \cap U_j|$ fails to be at most 2 for every j, we show this failure probability (where, again, the probability is taken over the choice of Y) is extremely small for a single fixed j and union bound over the j. Fix j and define $Q_h = Y'_h Y_{h-(k-j)}$; then $|V_k \cap U_j| = \sum_h Q_h$. Unfortunately, the Q_h are not independent. To deal with this, partition the indices h into two classes, where the first class contains all h such that $h \mod 2(k-j)$ falls in the range $[0, k-j-1]$ and the second class contains all other h. Notice that h and $h-(k-j)$ always are in separate classes, and so the set of all Q_h such that h is in the first class are independent, and, similarly, the set of all Q_h such that h is in the second class are independent. We show that $\sum_h Q_h \leq 1$, where the sum is restricted to a single class. Recall that the bits of Y are generated (independently) such that each bit is 1 with probability $r^{-\epsilon}$ and that, if we select the special bits S_k at random (which is allowed because we need only show $\exists S_K$ that satisfies the above) such that each of the bits of Y that line up with a portion of X (when sliding Y over X, only part of Y lines up with actual indices of X at any given shift) are included in S_k with probability $r^{-(1-\epsilon)}$ then a bit of Y' is 1 with probability $r^{-(1-2\epsilon)}$. The result follows from a simple application of the Chernoff bound.

Thus far, we have shown that D can produce a multiplication instance to feed to C. To use the result produced by C (namely, the predicted next bit of the product) to determine the parity of T, notice that the correct value of the next bit of the product is simply the exclusive-or of the parity of T, the parity of those bits of X_{fixed} that appear in column k of the multiplication problem, and the carry bit that enters column k when the low $k-1$ bits of YX_{input} and YX_{fixed} are added to produce the low $k-1$ bits of the product XY. Since X_{fixed} is a single fixed value, the parity of those bits that appear in column k can be built in to D. As noted earlier, it is possible, in AC^0, to compute the sum of the low $k-1$ bits of YX_{input} and YX_{fixed}, including the carry into column k. Thus, if the next bit can be predicted with some advantage, then the parity of T can be predicted with the exact same advantage. This contradiction completes the proof that the multiplication problem, as defined above, looks random to AC^0.

□

It is worth noting that, while the above proof was only carried out in the case when $r < s^\alpha$ for all constants $\alpha > 0$, but $r > \log^c s$ for all constants c, the same technique would also work for other parameters, such as if $s = r^u$, for some constant u (the parameters of Lemma 3). Moreover, a similar argument would show that, if $r = O(\log^c s)$, then f_Y passes all AC^0 tests of depth at most d, where d depends on c.

4 The Algebraic Integer Problem

In this section, it is shown that the algebraic integer problem looks random to AC^0 circuits. We begin with a few definitions. An algebraic integer is a root of some monic polynomial with integer coefficients. An algebraic number field is a finite field extension of $\mathbb{Q}$. Given some algebraic number field K, the ring of integers of K, denoted O_K, is the ring that consists of all algebraic integers in K. For every K, O_K is a free $\mathbb{Z}$-module, and so has an integral basis (that is to say, $\exists b_1, \ldots b_h \in O_K$ such that every element of O_K can be uniquely expressed as $\sum_i a_i b_i$, for $a_i \in \mathbb{Z}$). For a particular basis B, we define the function $f_B : \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_h} \to \{0,1\}^k$ such that $f_B(a_1, \ldots, a_h)$ is the first k bits of the binary expansion of the fractional real part of $\sum_i a_i b_i$, where for $i > 1$, $m_i = m_1^{u_i}$ for some constant $u_i > 0$, and $k = m_1^u$, for any constant u. We show, via reduction from the multiplication problem, that certain f_B are AC^0-pseudorandom.

As an example, consider the algebraic number field $K = \mathbb{Q}(\sqrt{d})$, for d a squarefree positive integer. It can be shown that, when $d \equiv 2, 3 \mod 4$, then $\{1, \sqrt{d}\}$ is an integral basis for O_K and that when $d \equiv 1 \mod 4$, $\{1, (1+\sqrt{d})/2\}$ is an integral basis for O_K (of course, since d is squarefree, we can't have $d \equiv 0 \mod 4$). Let b_1 and b_2 denote the basis elements, in the order they appear above. Then $f_B(a_1, a_2)$ is simply the first k bits of the fractional part of $a_1 b_1 + a_2 b_2$, which is identical to the first k bits of the fractional part of $a_2 b_2$ (because $a_1, b_1 \in \mathbb{Z}$). It is straightforward to show that, for all sufficiently large n, and all strings $Y \in \{0,1\}^{\lfloor n/2 \rfloor - 1}$, there is an n bit value d for which the binary expansion of the fractional part of $\sqrt{d}$ starts with the string Y. In particular, if we consider a string Y such that the multiplication function f_Y is AC^0-pseudorandom, then the corresponding f_B is also AC^0-pseudorandom, because it is just the multiplication problem $a_2\sqrt{d}$ bit-shifted, possibly with $1/2$ added.

In general, consider any basis B of some O_K such that there is some basis element b_j in B such that the binary expansion of the fractional real part of b_j starts with a value Y for which f_Y is AC^0-pseudorandom. Rather than consider f_B directly, it will again be convenient to consider a variant of the function in which some of the inputs are held fixed. In particular, we wish to fix a_i for each $i \neq j$. Define the function $f_{B,j,a_1,\ldots,a_{j-1},a+j+1,a_h} : \{0,1\}^{m_j} \to \{0,1\}^k$ such that it maps the value a_j to the first k bits of $\sum_i a_i b_i$. By a straightforward reduction from the multiplication problem, it follows that $f_{B,j,a_1,\ldots,a_{j-1},a+j+1,a_h}$ is AC^0-pseudorandom, which then immediately implies that f_B is AC^0-pseudorandom.

References

[Ajt83] M. Ajtai, Σ_1^1-*Formulae on Finite Structure*, APAL (1983).

[AGM02] N. Alon, O. Goldreich, and Y. Mansour, *Almost k-wise independence versus k-wise independence*, Electronic Colloquium on Computational Complexity, Report TR02-048 (2002).

[Bra09] M. Braverman, *Poly-logarithmic independence fools AC0 circuits*, IEEE Conference on Computational Complexity (2009), 3-8.

[CSV84] A. K. Chandra, L. Stockmeyer, and U. Vishkin, *Constant depth reducibility*, SIAM Journal on Computing (1984), 13:423-439.

[FSS84] M. Furst, J. Saxe, and M. Sipser, *Parity, circuits, and the polynomial time hierarchy*, Mathematical Systems Theory (1984), 17:13-27.

[Gre12] B. Green, *On (not) computing the Möbius function using bounded depth circuits*, http://arxiv.org/pdf/1103.4991.pdf (2012).

[Has86] J. Hastad, *Computational limitations for small depth circuits*, MIT Press (1986), Ph. D. thesis.

[KLL84] R. Kannan, A. K. Lenstra, L. Lovasz, *Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers*, STOC (1984), 191-200.

[Kat86] I. Katai, *Distribution of digits of primes in q-ary canonical form*, Acta Math (1986), 47(3-4):341-359.

[LN90] N. Linial and N. Nisan, *Approximate inclusion-exclusion*, Cominatorica (1990), 10(4):349-365.

[Nis91] N. Nisan, *Pseudorandom bits for constant depth circuits*, Combinatorica (1991), 11(1):63-70.

[NW94] N. Nisan and A. Wigderson, *Hardness vs Randomness*, Journal of Computer and Systems Sciences (1994), 49(2):149-167.

[OSB99] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-Time Signal Processing*, Prentice Hall (1999).

[Vaz86] U. V. Vazirani, *Randomness, Adversaries and Computation*, Ph.D. Thesis, EECS, UC Berkeley (1986).

[Yao82] A. C. Yao, *Theory and application of trapdoor functions*, IEEE Symposium on Foundations of Computer Science (1982), 80-91.

[Yao85] A. C. Yao, *Separating the polynomial-time hierarchy by oracles*, Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science (1985), 1-10.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de