Tiny Families of Functions with Random Properties: A Quality–Size Trade–off for Hashing (Preliminary Version)

Oded Goldreich*
Department of Applied Mathematics
and Computer Science
Weizmann Institute of Science
Rehovot, Israel.

Avi Wigderson†
Institute for Computer Science
Hebrew University
Jerusalem, Israel.

Abstract

We present three explicit constructions of hash functions, which exhibit a trade-off between the size of the family (and hence the number of random bits needed to generate a member of the family), and the quality (or error parameter) of the pseudo-random property it achieves. Unlike previous constructions, most notably universal hashing, the size of our families is essentially independent of the size of the domain on which the functions operate.

The first construction is for the *mixing* property – mapping a proportional part of any subset of the domain to any other subset. The other two are for the *extraction* property – mapping any subset of the domain almost uniformly into a range smaller than it. The second and third constructions handle (respectively) the extreme situations when the range is very large or very small.

We provide lower bounds showing our constructions are nearly optimal, and mention some applications of the new constructions.

1 Introduction

In 1979, Carter and Wegman introduced the notion of universal hashing functions [7]. Though these functions were introduced with data storage application in mind, they found many applications to complexity theory [30, 32, 35, 18, 16, 21, 22, 19, 20, 27, 28, 37]. This wide range of applications owns its existence to two related 'random' properties of these succinct and efficiently computable functions: the *extraction* and the *mixing* properties.

For a family F of functions, each mapping n-bit strings to m-bit strings, the *extraction* property asserts the following. Every subset of $K \cdot 2^m$ strings in the domain $\{0,1\}^n$, is mapped almost uniformly to the range $\{0,1\}^m$, by all but a small fraction of the functions in the family. The parameter $K > 1$ determines the quality of the approximation to the uniform distribution and the fraction of bad functions in F (i.e. those that don't achieve this approximation). The extraction property is the heart of the Leftover Hash Lemma [21] and its precurors, which were key to numerous results, e.g. in saving randomness [22], weak random sources [37], pseudorandom generators [16, 21] and interactive proofs [18].

The *mixing* property is meaningful also in case $m = n$, and in fact it is usually used with this choice. Hence, we assume for simplicity that $m = n$. Loosely speaking, the mixing property asserts that, for all but a small fraction of the functions f in the family F, the membership in

*Research was supported in part by grant No. 92-00226 from the United States – Israel Binational Science Foundation (BSF), Jerusalem, Israel.

†Research was supported in part by the Wolfson Research Awards, administered by the Israel Academy of Sciences and Humanities.

STOC 94- 5/94 Montreal, Quebec, Canada
© 1994 ACM 0-89791-663-8/94/0005..$3.50

$A \times B$ of a pair $(a, f(a))$ with a being a random element from the domain, is essentially the same as that of a random pair (a, b) of elements. The prime use of the mixing property is in the logspace pseudorandom generators [27, 28].

In the definitions above, there is an error parameter ϵ (e.g. the fraction of bad functions, the distance from the uniform distribution etc.), which determines the quality of the mixing or extraction achieved by the family F. All the applications mentioned above take F to be a universal family of hash functions. This family achieves the best possible quality parameter: ϵ is exponentially small in m. However, while small enough for these applications, a universal family has to be large: exponential in n.

But in some applications we may be content with a larger ϵ (i.e. lower quality), say constant or $1/poly(n)$. Can we use much smaller families F in this case and achieve similar random properties? A straightforward counting argument shows (nonconstructively, of course) that there exist families F of size $poly(1/\epsilon)$ (resp. $poly(n/\epsilon)$) achieving the mixing (resp. extraction) properties with quality ϵ. Note that these bounds depend essentially only on the quality required, and not on the size of the domain.

The main contribution of this paper is in presenting explicit constructions of such families, thus yielding a trade-off between the size of the family and the desired quality. The first construction is for mixing, where we obtain a complete trade-off. The second and third constructions are for extraction, where we (respectively) handle two extreme cases: when $n-m \ll n$ and $m \ll n$. Our constructions are relatively simple. The first two of them combine universal hashing and expander graphs. (It is interesting to note that despite the similarity in these two constructions, the proofs are completely different). An alternative to the second construction, which is often more efficient, uses the extractors of [29] instead of universal hashing. The third construction uses small-bias probability spaces of small size. We provide lower bounds to show that the first construction is nearly optimal, and the third is nearly optimal for sufficiently small m. By nearly optimal here we mean that the number of bits needed to describe a member of the family in our constructions is within a constant factor of the lower bound. The second construction uses a number of random bits which is at most quadratic in the lower bound.

It is not surprising that these constructions already found some interesting applications. Using the first construction we reduce the randomness complexity of two generic procedures as follows:

1. For sampling procedures, which use an asymptoticaly optimal number of sample points, the amount of randomness required to generate the sample points is reduced by a factor of 2; and

2. The randomess complexity of Nisan's "generalized logspace" generator [27], is reduced by a logarithmic factor.

The second construction implies a randomness–efficient leftover hash lemma. The third construction turned out to be the main technical tool in the recent result of [34] (who independently discovered a similar construction). They completely resolve the simulation problem of BPP algorithms by Santha-Vazirani [33] sources. Recall that such a source is required to give the each successive output bit a nontrivial probability δ for both HEADS and TAILS. While the best previous result [36] required a constant $\delta > 0$, the new result requires only that on n-bit output, $\delta > n^{-\Omega(1)}$. A simple information theoretic argument [11] shows that this bound is best possible.

Despite the general interest in reducing the size of sample spaces achieving various random properties, very little was done for the proerties provided by universal hashing. The only previous result achieving such a quality-size trade-off is the paper [29]. They deal with extraction in the difficult range $m = \Theta(n)$ (which we cannot handle), via an ingenious construction. Moreover, they applied it to show that poly(S) random bits add no power at all to space(S) Turing machines! As mentioned above, [34] independently discovered a construction similar to our third one. They obtain the same bounds, but with a somewhat different proof.

Organization

The following three sections are devoted to the corresponding three constructions mentioned above. Each section starts with a brief intuitive summary of the results obtained. Next, comes a formal statement of the result and a description of the construction which achieves it. We briefly touch on the technical tools used in the proof that the construction works, and present the relevant lower bound. In addition, for the first construction, we describe two applications.

All technical tools used in the proofs apear in section ??. Further details and complete proofs can be found in our technical report [17].

2 Tiny Families of Functions with Mixing Properties

Recall that a function f is mixing for sets A, B of the domain, if membership in $A \times B$ of a pair $(a, f(a))$, with a being a random element in the domain, occurs roghly as often as it would for a random pair (a, b) of elements. The main result of this section is the explicit construction of an ϵ-mixing family of size poly$(1/\epsilon)$. Here ϵ stands both for distance from truely random behaviour, as well as the fraction of bad functions which do not achieve this distance. We state the precise theorem, then describe the construction. We prove that our family has optimal size up to a polynomial, and present an application to saving randomaness in the generalized logspace model of [27]. We conclude with a different perspective of this result, advocated by Linial.

Main result

Theorem 1 *For every $\epsilon > 2^{-\Omega(n)}$, there exists a family of functions, each mapping $\{0,1\}^n$ to itself, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{1}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(\epsilon) = O(\log \frac{1}{\epsilon})$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*
- mixing property: *For every two subsets $A, B \subseteq \{0,1\}^n$, all but an ϵ fraction of the functions f in the family satisfy*

  $$|\text{Prob}(X_n \in A \wedge f(X_n) \in B) - \rho(A)\rho(B)| \leq 2\epsilon$$

  where $\rho(S) \overset{\text{def}}{=} \frac{|S|}{2^n}$ denotes the density of the set S and X_n is a random variable uniformly distributed over $\{0,1\}^n$.

The Construction

The construction makes used of two basic tools which are frequently used for saving randomness: universal hashing functions and expander graphs.

We start by setting the parameters for the expander graph and the universal hashing family to be used. First, let G be an expander graph of degree d, second eigenvalue λ, and vertex set $\{0,1\}^n$, so that $\frac{\lambda}{d} \leq \epsilon^2$. Such expander graphs are easily constructible for $d = \frac{1}{\epsilon^{O(1)}}$ (cf., [15]) [1] Assume, without loss of generality, that d is a power of 2. For every $i \in [d] \overset{\text{def}}{=} \{1, 2..., d\}$ and $v \in \{0,1\}^n$, denote by $g_i(v)$ the vertex reached by moving along the i^{th} edge of the vertex v.

We next consider a universal family, denoted H, of hash functions, each mapping $l \overset{\text{def}}{=} 4\log_2(1/\epsilon)$-bit long strings to $[d]$ (where $[d] = \{0,1\}^m$, for some m). Namely, a *uniformly chosen function* $h \in H$ maps each string $\alpha \in \{0,1\}^l$ uniformly into $[d]$ so that every two strings are mapped in an independent manner.

[1] Actually, using Ramanujan Graphs, it suffices to have $d = \frac{4}{\epsilon^4}$ (cf., [24]). One may prefer the Gaber-Galil expander since it allows to avoid problems such as generating large primes and embedding $\{0,1\}^n$ in $GF(p)$, for a suitably large prime p.

We now define the functions in our family, denoted F. For each hashing function $h \in H$, we introduce a function $f \in F$ defined by

$$f(v) \stackrel{\text{def}}{=} g_{h(\text{lsb}(v))}(v)$$

where $\text{lsb}(v)$ returns the l least significant bits of $v \in \{0,1\}^n$. Namely, $f(v)$ is the vertex reached from v by following the i^{th} edge of v, where i is the image of the l least significant bits of v under the function h. (We remark that our choice of using the l least significant bits is arbitrary and any other efficient partition of $\{0,1\}^n$ into 2^l parts, of approximately the same size, will do.)

The full proof is given in our technical report [17]. The two main technical tools it uses are the Expander Mixing Lemma, and the pairwise independence of the universal hash function family.

Lower Bound

Theorem 2 *A family with mixing property of accuracy ϵ, must have size at least $\sqrt{\frac{4}{\epsilon}}$.*

Applications

We present two applications of the new construction. The first application is to reducing the randomness complexity of sampling techniques (specifically, by a factor of 2); and the second is to reducing the cost of the deterministic simulation of a generalization, due to Nisan [27], of random-logspace computations.

Sampling

In many settings repeated sampling is used to estimate the average value of a huge set of values. Namely, there is a value function ν defined over a huge space, say $\nu\colon\{0,1\}^n \mapsto [0,1]$, and one wishes to approximate $\bar{\nu} \stackrel{\text{def}}{=} \frac{1}{2^n}\sum_{x\in\{0,1\}^n}\nu(x)$. To this end, one may randomly select a small sample set S and compute $\frac{1}{|S|}\sum_{x\in S}\nu(x)$. Using a sample of $O(1/\epsilon^2)$ uniformly and independently selected points, one gets, with constant probability, an approximation that it within an additive factor of ϵ from the correct average. In fact, a set of $O(1/\epsilon^2)$ points selected in a pairwise-independent and uniform manner yields the same quality of approximation. Whereas generating t totally independent random points in $\{0,1\}^n$ requires $t \cdot n$ unbiased coin flips, one can generate t pairwise-independent random points using only $2 \cdot n$ unbiased coin flips [10]. Using the new family of functions, we further reduce the randomness complexity of the approximation problem to $n + O(\log(1/\epsilon))$, while almost maintaining the number of sample points.

Definition 1 (sampler): *A* sampler *is a randomized algorithm that on input parameters n* (length), *ϵ* (accuracy) *and δ* (error), *and oracle access to any function $\nu\colon\{0,1\}^n \mapsto [0,1]$, outputs, with probability at least $1-\delta$, a value that is at most ϵ away from $\bar{\nu}$. Namely,*

$$\text{Prob}(|\text{sampler}^{\nu}(n,\epsilon,\delta) - \bar{\nu}| > \epsilon) \ < \ \delta$$

Theorem 3 *There exists a* $\text{poly}(n,\epsilon,\delta)$*-time sampler which*

- *makes $O(\frac{1}{\delta\epsilon^2})$ oracle queries; and*
- *tosses $n + O(\log(1/\epsilon)) + O(\log(1/\delta))$ coins.*

We remark that samplers for Boolean functions can be obtained in a more direct way; and furthermore, these samplers use only n coin tosses (see our technical report [17]). Using the result of Bellare et al [5], we get the same reduction in the randomness complexity, while reducing the number of sample points.

Corollary 4 *There exists a* $\text{poly}(n,\epsilon,\log(1/\delta))$*-time sampler which*

- *makes $O(\frac{\log(1/\delta)}{\epsilon^2})$ oracle queries; and*
- *tosses $n + O(\log(1/\epsilon)) + O(\log(1/\delta))$ coins.*

The last sampler is optimal (up to a multiplicative factor) in its sample-complexity, and among the samplers with nearly optimal sample complexity the above is optimal (up to the additive logarithmic factors) in its randomness-complexity [6]. Previously, efficient samplers with optimal sample-complexity were known only for twice the randomness-complexity [5] (yet, [6] have proved, via an non-constructive argument, that samplers with sample and randomness complexities as in the corollary do exist).

Generalized Random Logspace

In [27], Nisan considered the problem of saving randomness in a context in which m randomized algorithms are executed and their output is fed to an s-space machine which then produces a final Boolean output. (Actually, the problem is not affected if the s-space machine is allowed to have output of length bounded by $O(s)$.) For simplicity, assume that each of the algorithms uses n coin flips. The obvious way of running the entire procedure requires $m \cdot n$ coin flips. In case we are willing to tolerate an ϵ additive error (respectively, deviation) in the final output, more randomness-efficient solutions are possible. In particular, Nisan showed [27] that the randomness complexity can be decreased to

$$O(\max\{n, s + \log(m/\epsilon)\} \cdot \log m)$$

Replacing the universal hash functions used in [27] by our family of mixing functions, we show

Theorem 5 *The above problem can be solved with randomness complexity*

$$n + O((s + \log(m/\epsilon)) \cdot \log m)$$

We remark that in many applications $n \gg s + \log(m/\epsilon)$. Specifically, it is reasonable to consider the case where m, $1/\epsilon$ and n are all polynomially related, and furthermore $s = O(\log m)$. For these cases, our improvement yields a logarithmic reduction in the randomness complexity.

A Different Perspective

The mixing property of families of functions should not be confused with the mixing property of graphs. Yet, the two are related as we shall see below. We say that a graph has a good mixing property if for every two subsets of vertices the fraction of edges connecting these subsets is approximately equal the product of the densities of these subsets. Clearly, a family of functions over $\{0,1\}^n$, with good mixing, induces a regular multi-graph[2] with good mixing. The converse is not obvious. Specifically, it was not even known whether the edges of some small degree graph with good mixing property (e.g., an expander) can be so colored that they induce a family of functions with a good mixing property. In fact, this problem has been advocated by Nati Linial.

Let us try to clarify the nature of this problem. Consider a d-degree expander with vertex-set $V \stackrel{\text{def}}{=} \{0,1\}^n$, and some d-coloring of its edges. For every two sets of vertices, A and B, denote by $E_i(A,B)$ the set of edges of color i that connect a vertex in A to a vertex in B. By the Expander Mixing Lemma (see following section), it follows that the *average* of $\frac{|E_i(A,B)|}{|V|}$, taken over all $1 \leq i \leq d$, is approximately $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$. The question is whether $\frac{|E_i(A,B)|}{|V|}$ is approximately $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$, *for almost all* $1 \leq i \leq d$. One can easily verify that, in general, the answer is negative. Specifically, for some Cayley expander graphs there are sets A and B for which *there exist no* i such that $\frac{|E_i(A,B)|}{|V|}$ approximates $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$. The problem raised by Nati Linial was to construct an expander for which the mixing property holds for most colors (and not only on the average).

We resolve this problem by presenting a transformation of edge-colored expanders to edge-colored expanders for which the mixing property holds for most colors (as required above). Our transformation preserves the vertex set and the expansion properties of the original expander, but increases the degree by a polynomial factor (i.e., from d to poly(d)). Although the transformation is not explicitly presented in this extended abstract, it can be easily derived from the description.

[2] A multi-graph is a graph in which parallel edges are allowed.

3 Tiny Families Extracting High Min-entropy

Recall that the *extraction* property, for a family of functions each mapping n-bit strings to m-bit strings, means that each subset of $K \cdot 2^m$ strings in $\{0,1\}^n$ is mapped almost uniformly to $\{0,1\}^m$, by all but a small fraction of the functions in the family. We consider the extraction problem in two special cases: the case where m is very small (in the next section) and the case m is very close to n (in this section). Actually, we consider a generalization of the extraction problem to random variables with an upper bound, of $\frac{1}{K\,2^m}$, on the probability function. Such a bound is called *min-entropy* (cf., Chor and Goldreich [9]).

Definition 2 (min-entropy): *Let X be a random variable. We say that X has* min-entropy *k if* $\text{Prob}(X\!=\!x) \leq 2^{-k}$*, for each x.*

Here we treat the case of random variables with min-entropy $n-k$ with $k \ll n$. We construct a family of $\text{poly}(2^k/\epsilon)$ functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, where $m = n - O(k)$. For each such random variable, all but a ϵ fraction of the functions, when applied to it, yield a random variable which is ϵ-close to uniform (in norm-1). Loosely speaking, this means that these functions are able to "smooth" almost the entire min-entropy; specifically, min-entropy $n-k$ is mapped to almost uniform distribution over the strings of length $n - O(k)$.

In a typical use of this extraction, most notably the applications of the leftover hash lemma, $\epsilon = 2^{-\Omega(k)}$. In these cases the size of our family is $\text{poly}(1/\epsilon)$ which is optimal by the lower bound we give. There may be cases, however, where ϵ may be allowed to be much larger. Here we have an alternative constcruction with size $k^{O(\log(1/\epsilon))}$, which will be much smaller than $2^k/\epsilon$ in this case. We mention both in the precise theorem.

Main Result

Theorem 6 *Let $k < n$, $m < n-k$ and $\epsilon > 2^{-(n-m-O(k))/O(1)}$.* (Typically, $m = n - O(k)$ and $\epsilon = 2^{-\Theta(n-m)}$.) *There exists a family of functions, each mapping $\{0,1\}^n$ to $\{0,1\}^m$, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{2^k}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(k,\epsilon) = O(k + \log\frac{1}{\epsilon})$. Alternatively, we can achieve size $k^{O(\log(1/\epsilon))}$, with the description length of each function being $l(k,\epsilon) = O((\log k)(\log(1/\epsilon)$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*
- extraction property: *For every random variable $X \in \{0,1\}^n$ of min-entropy $n-k$, all but an ϵ fraction of the functions f in the family satisfy*

$$\sum_{\alpha \in \{0,1\}^m} |\text{Prob}(f(X)\!=\!\alpha) - \frac{1}{2^m}| \leq \epsilon$$

The construction

We start with the main construction. At the end, of this subsection, we explain how to modify it and get the alternative construction. Again, we use universal hashing functions and expander graphs. This time we use an expander graph, G, degree d (power of two), second eigenvalue λ, and vertex set $\{0,1\}^m$, so that $\frac{\lambda}{d} \leq \frac{\epsilon}{2^{k/2}}$. (Recall that such an expander can be constructed for $d = \text{poly}(\frac{2^k}{\epsilon})$.) As before, for every $i \in [d] \stackrel{\text{def}}{=} \{1,2...,d\}$ and $v \in \{0,1\}^m$, denote by $g_i(v)$ the vertex reached by moving along the i^{th} edge of the vertex v. The universal family, denoted H, contains hash functions each mapping $(n-m)$-bit long strings to $[d]$.

We now define the functions in our family, denoted F. For each hashing function $h \in H$, we introduce a function $f \in F$ defined by

$$f(x) \stackrel{\text{def}}{=} g_{h(\text{lsb}(x))}(\text{msb}(x))$$

where $\text{lsb}(x)$ returns the $n-m$ least significant bits of $x \in \{0,1\}^n$, and $\text{msb}(x)$ returns the m

most significant bits of x. Namely, $f(x)$ is the vertex reached from the vertex $v \overset{\text{def}}{=} \text{msb}(x)$ by following the i^{th} edge of v, where i is the image of the $n-m$ least significant bits of x under the function h. (We remark that our choice of using the $n-m$ least significant bits is arbitrary and any other efficient partition of $\{0,1\}^n$ into 2^{n-m} parts, of approximately the same size, will do.)

Again, the full proof is provided in our technical report [17]. Despite the apperant similarity to the construction for mixing, the proof here is completely different. It is based on "stronger" technical tools : the Expander Smoothing Lemma and the Leftover Hash Lemma.

Finally, to obtain the alternative construction mentioned in the main theorem of this section, simply replace the hash function family H by a family of (δ, ϵ)-extractors from [29], with $\delta = \log k / \log(n-m)$, on the same domain and range used above. The proof for this construction is identical, since the extraction property of this family guarantees the same as the Leftover Hash Lemma for these parameters.

Lower Bound

We conclude with the lower bound. It shows, that for $\epsilon = 2^{-\Omega(k)}$, our first construction is optimal. It also shows that in the general, the number of bits used in the alternative construction is at most quadratic away from optimum.

Theorem 7 *A family of functions from $\{0,1\}^n$ to $\{0,1\}$, with extraction property of accuracy $\epsilon < 1$ with respect to random variables of min-entropy $n-k \leq n-1$, must have size at least* $\max\{k+1, (1/\epsilon)-1\}$.

4 Tiny Families Extracting Low Min-Entropy

Here we treat the case of random variables with min-entropy k, with $k \ll n$. we construct a family of $\text{poly}(2^k n/\epsilon)$ functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, where $m = \Omega(k)$. (Again, ϵ is the accuracy parameter.) Loosely speaking, this means that these functions are able to "smoothen" a constant fraction of the min-entropy; specifically, min-entropy k is mapped to almost uniform distribution over the strings of length $\Omega(k)$.

Main Result

Theorem 8 *Let* $5m < k < n$ *and* $\epsilon > 2^{-(k-5m)/3}$. (Typically, $m = \frac{k}{8}$ and $\epsilon = 2^{-m}$.) *There exists a family of functions, each mapping $\{0,1\}^n$ to $\{0,1\}^m$, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{2^m n}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(\frac{2^m n}{\epsilon}) = O(m + \log \frac{n}{\epsilon})$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*
- extraction property: *For every random variable $X \in \{0,1\}^n$ of min-entropy k, all but an ϵ fraction of the functions f in the family satisfy*

$$\sum_{\alpha \in \{0,1\}^m} |\text{Prob}(f(X) = \alpha) - \frac{1}{2^m}| \leq \epsilon$$

The Construction

We use a construction of small probability spaces with small bias. In particular, we consider a prime $p \approx 2^m$ and a construction of $t \overset{\text{def}}{=} \frac{n}{m}$ random variables, $(\xi_1, ..., \xi_t)$, each distributed over $GF(p)$ with the following *small bias* property:

> for every t-long sequence $(a_1, ..., a_t)$ of elements in $GF(p)$, so that not all a_i's are zero, the random variable $\sum_{i=1}^t a_i \xi_i$ is almost uniformly distributed over $GF(p)$ (i.e., its statistical distance from uniform is small).

Typically, such random variables are defined by the uniform distribution over some sample space

$S \subseteq GF(p)^t$, and they can be show to satisfy also a related technical condition [3]. We will use such a sample space, S, for bias $\epsilon' \stackrel{\text{def}}{=} \frac{\epsilon^3}{p^5}$. (Hence, using the sample space of [3, 14, 13], $|S| = \text{poly}(\frac{n2^k}{\epsilon})$.)

The functions in our family, denoted F, correspond to the samples in the small-bias space. Namely, for each $(s_1, ..., s_t) \in S$, we introduce the function $f \in F$ defined by

$$f(x) \stackrel{\text{def}}{=} \sum_{i=1}^{t} s_i x_i$$

where x_i is the i^{th} coordinate in $x \in GF(p)^t$ and the arithmetic is in $GF(p)$. The functions, so defined, map $GF(p)^t$ to $GF(p)$. Standard modifications can be applied to derive functions mapping $\{0,1\}^n$ to $\{0,1\}^m$ (recall $p \approx 2^m$).

The proof of this theorem is also given in our technical report [17]. Despite the need for a norm-1 bound, it uses the fact that the construction of small–bias spaces of [3, 13] give a bound in norm-2 on the realted exponential sum . We then prove a Lindsey–like lemma on near-othogonal vectors and combine it with the bound above to give the result.

Lower Bound

To illustrate that this construction is near optimal when $k = O(\log n)$ we restate Theorem 7 with the necessary change of parameters. We note that the BPP simulation of [34] mentioned in the introduction indeed uses this construction for this value of the parameter k.

Theorem 9 *A family of functions from* $\{0,1\}^n$ *to* $\{0,1\}$*, with extraction property of accuracy* $\epsilon < 1$ *with respect to random variables of min-entropy* $k \leq n-1$*, must have size at least* $\max\{n-k+1, (1/\epsilon)-1\}$.

5 Technical Tools

5.1 Universal Hashing

Loosely speaking, universal families of hashing functions consist of functions operating on the same domain-range pair so that a function uniformly selected in the family maps each pair of points in a pairwise independent and uniform manner. Specifically, a family, $H_{n,m}$, of functions from $\{0,1\}^n$ to $\{0,1\}^m$, is called *universal* if for every $x \neq y \in \{0,1\}^n$ and $\alpha, \beta \in \{0,1\}^m$ it holds

$$\text{Prob}(h(x) = \alpha \wedge h(y) = \beta) = 2^{-2m}$$

where the probability is taken over all choices of $h \in H_{n,m}$ with uniform probability distribution.

Several efficient families of universal hashing functions are known [7]. The functions in these families can be described using $O(n+m)$ bits and posses an efficient (e.g., polynomial-time and even logspace) evaluating algorithms.

The two main facts we will use about universal hash families are:

Pairwise Independence

Lemma 1 *The set of random variables* $\{h(x) | x \in \{0,1\}^n\}$ *defined by a random* $h \in H$ *are pairwise independent.*

Leftover Hash Lemma

This fundamental lemma of [21] asserts that a random hash function from a universal family will smooth min-entropy k (recall definition in the previous section) whenever the range M is smaller than k. More precisely

Lemma 2 (Leftover Hash Lemma [21]): *Let* X *be any random variable on* $\{0,1\}^n$ *with min-entropy* k*. Then the distribution* $(h, h(X))$*, with* h *chosen at random from* $H_{n,m}$*, has (norm-1) distance* $2^{(m-k)/2}$ *from the uniform distribution.*

5.2 Expanders

The Expander Mixing Lemma

The following lemma is folklore and has appeared in many papers. Loosely speaking, the lemma asserts that expander graphs (for which $d \gg \lambda$) have the property that the fraction of edges between two large sets

[3] I.e., the expectation of (the norm-2 of) $\omega^{\sum_{i=1}^{t} a_i \xi_i}$ is small, where ω is a p^{th} root of unity (in the complex field).

of vertices approximately equals the product of the densities of these sets. This property is called *mixing.*

Lemma 3 (Expander Mixing Lemma): *Let $G = (V, E)$ be an expander graph of degree d and λ be an upper bound on the absolute value of all eigenvalues, save the biggest one, of the adjacency matrix of the graph. Then for every two subsets, $A, B \subseteq V$, it holds*

$$\left|\frac{|(A \times B) \cap E|}{|E|} - \frac{|A|}{|V|} \cdot \frac{|B|}{|V|}\right| \leq \frac{\lambda\sqrt{|A| \cdot |B|}}{d \cdot |V|} < \frac{\lambda}{d}$$

The Expander Smoothing Lemma

The following lemma follows easily by the standard techniques of dealing with random walks on expander graphs.

Lemma 4 (Expander Smoothing Lemma): *Let $G = (V, E)$, d and λ be as in the previous lemma. Let X be a random variable, distributed over V, so that $\text{Prob}(X = v) \leq \frac{K}{|V|}$, for every $v \in V$, and Y denote the vertex reached from X by following a uniformly chosen edge. Then*

$$\sum_{v \in V} |\text{Prob}(Y = v) - \frac{1}{|V|}| < \frac{\lambda}{d} \cdot \sqrt{K - 1}$$

5.3 Small Probability Spaces with the Small Bias Property

The following definition of small-bias sample spaces implies the definition presented in section 4.

Definition 3 *Let k be an integer, p be a prime and ω be a p^{th} root of unity (in the complex field). A set $S \subseteq GF(p)^t$ ıs said to have ϵ* bias (sample space for *$GF(p)^t$) if, for every t-long sequence $(a_1, ..., a_t)$ of elements in $GF(p)$, so that not all $a_ı$'s are zero, the expectatıon of (the norm-2 of) $\omega^{\sum_{ı=1}^{t} a_ı s_ı}$, taken over all $(s_1, ..., s_t) \in S$ with uniform distribution, is bounded above by ϵ.*

By Alon et. al. [3] (see also Even et. al. [14])

Theorem 10 *For every integer t, prime p and $\epsilon > 0$, there exists an efficiently constructible ϵ-bias sample space of size $(t/\epsilon)^2$ for $GF(p)^t$.*

5.4 Nisan-Zuckerman Extractors

The following version (which is not completely general) of the results in [29] is sufficient for our purposes. A family of functions H mapping n-bit strings to m-bit strings is (δ, ϵ)–extracting, if for every random variable X of min-entropy δn, the distribution $(h, h(X))$ has distance ϵ from the uniform distribution, when h is chosen uniformly from H. Then [29] prove:

Theorem 11 [29]: *For every $\delta > 0$ and $\epsilon = \epsilon(n)$ there is an explicit construction of a (δ, ϵ)-extracting family with $m = \delta^2 n$ of size $n^{O(\log(1/\epsilon))}$.*

References

[1] M. Ajtai, J. Komlos, E. Szemerédi, "Deterministic Simulation in LOGSPACE", *Proc. 19th STOC*, 1987, pp. 132–140.

[2] N. Alon, "Eigenvalues, Geometric Expanders, Sorting in Rounds and Ramsey Theory", *Combinatorica*, 6 (1986), pp. 231–243.

[3] N. Alon, O. Goldreich, J. Hastad, R. Peralta, "Simple Constructions of Almost k-wise Independent Random Variables", *Journal of Random structures and Algorithms*, Vol. 3, No. 3, (1992), pp. 289–304.

[4] N. Alon and V.D. Milman, "Eigenvalues, Expanders and Superconcentrators", *25th FOCS*, 1984, pp. 320–322.

[5] M. Bellare, O. Goldreich, and S. Goldwasser "Randomness in Interactive Proofs", *31st FOCS*, 1990, pp. 318–326.

[6] R. Canetti, G. Even and O. Goldreich, "Lower Bounds for Sampling Algorithms", submitted to *IPL*, 1993.

[7] L. Carter and M. Wegman, "Universal Classes of Hash Functions", *J. Computer and System Sciences*, Vol. 18, pp. 143–154 (1979).

[8] A. Cohen, A. Wigderson, " Dispersers, Deterministic Amplification and Weak random Sources", *Proc. of the 30th FOCS*, pp. 14–19, 1989.

[9] B. Chor and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", *SIAM J. Comput.*, Vol. 17, No. 2, April 1988, pp. 230–261.

[10] B. Chor and O. Goldreich, "On the Power of Two–Point Based Sampling," *Jour. of Complexity*, Vol 5, 1989, pp. 96–106.

[11] A. Cohen and A. Wigderson, "Dispensers, Deterministic Amplification, and Weak Random Sources", *30th FOCS*, 1989, pp. 14–19.

[12] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, 1974.

[13] G. Even, "Construction of Small Probability Spaces for Deterministic Simulation", M.Sc. thesis, Computer Science Department, Technion, Haifa, Israel, 1991. (In Hebrew, abstract in English)

[14] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličković, "Approximations of General Independent Distributions", *24th STOC*, pp. 10–16, 1992.

[15] O. Gaber and Z. Galil, "Explicit Constructions of Linear Size Superconcentrators", *JCSS*, **22** (1981), pp. 407-420.

[16] O. Goldreich, H. Krawcyzk and M. Luby, "On the Existence of Pseudorandom Generators", *29th FOCS*, pp. 12–24, 1988.

[17] O. Goldreich and A. Wigderson, Technical Report CS94-03, Weizmann Institute, Israel, 1994.

[18] S. Goldwasser and M. Sipser, "Private Coins versus Public Coins in Interactive Proof Systems", *18th STOC*, pp. 59–68, 1986.

[19] R. Impagliazzo and M. Luby, "One-Way Functions are Essential for Complexity Based Cryptography", *30th FOCS*, pp. 230–235, 1989.

[20] R. Impagliazzo and L.A. Levin, "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random ", *31st FOCS*, pp. 812-821, 1990.

[21] R. Impagliazzo, L.A. Levin, and M.G. Luby, "Pseudorandom Generators from any One-Way Functions", *21st STOC*, pp. 12–24, 1989.

[22] R. Impagliazzo and D. Zuckerman, "How to Recycle Random Bits", *30th FOCS*, 1989, pp. 248-253.

[23] R.M. Karp, N. Pippinger and M. Sipser, "A Time-Randomness Tradeoff", *AMS Conference on Probabilistic Computational Complexity*, Durham, New Hampshire (1985).

[24] A. Lubotzky, R. Phillips, P. Sarnak, "Explicit Expanders and the Ramanujan Conjectures", *Proc. 18th STOC*, 1986, pp. 240-246.

[25] G.A. Margulis, "Explicit Construction of Concentrators", *Prob. Per. Infor.* 9 (4) (1973), 71–80. (English translation in *Problems of Infor. Trans.* (1975), 325–332.)

[26] J. Naor and M. Naor, "Small-bias Probability Spaces: Efficient Constructions and Applications", *22nd STOC*, 1990, pp. 213–223.

[27] N. Nisan, "Pseudorandom Generators for Space Bounded Machines", *22nd STOC*, pp. 204–212, 1990.

[28] N. Nisan, "$\mathcal{RL} \subseteq \mathcal{SC}$", *24th STOC*, pp. 619–623, 1992.

[29] N. Nisan and D. Zuckerman, "More Deterministic Simulation in LOGSPACE", *25th STOC*, pp. 235–244, 1993.

[30] M. Sipser, "A Complexity Theoretic Approach to Randomness", *15th STOC*, 1983, pp. 330–335.

[31] M. Sipser, "Expanders Randomness or Time vs Space", *Structure in Complexity Theory* (proceedings), 1986.

[32] L. Stockmeyer, "The Complexity of Approximate Counting", *15th STOC*, 1983, pp. 118–126.

[33] M. Santha and U. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources", *JCSS*, Vol. 33, No. 1, pp. 75–87, 1986.

[34] A. Srinivasan and D. Zuckerman, "Computing with Very Weak Random Sources", manuscript, 1993.

[35] L. Valiant and V.V. Vazirani, "NP is as Easy as Detecting Unique Solutions", *Theoretical Computer Science*, Vol. 47, 1986, pp. 85–93.

[36] U. Vazirani and V. Vazirani, "Random Polynomial Time Equal to Semi-Random Polynomial Time", *Proc. 26th FOCS*, pp. 417–428, 1985.

[37] D. Zuckerman, "Simulating BPP Using a General Weak Random Source," *32nd FOCS*, 1991, pp. 79–89.