Extractors and Pseudorandom Generators*

Luca Trevisan†

February 26, 1999

Abstract

We introduce a new approach to constructing extractors. Extractors are algorithms that transform a "weakly random" distribution into an almost uniform distribution. Explicit constructions of extractors have a variety of important applications, and tend to be very difficult to obtain.

We demonstrate an unsuspected connection between extractors and pseudorandom generators. In fact, we show that every pseudorandom generator of a certain kind is an extractor.

A pseudorandom generator construction due to Impagliazzo and Wigderson, once reinterpreted via our connection, is already an extractor that beats most known constructions and solves an important open question. We also show that, using the simpler Nisan-Wigderson generator and standard error-correcting codes, one can build even better extractors with the additional advantage that both the construction and the analysis are extremely simple and admit a short self-contained treatment.

1 Introduction

An *extractor* is an algorithm that converts a "weak source of randomness" into an almost uniform distribution by using a small number of additional truly random bits. Extractors have several important applications. In this paper we show that pseudorandom generator constructions of a certain kind are extractors. Using our connection and some new ideas we describe constructions of extractors that improves most previously known constructions and that are exceedingly simpler than previous ones.

In this section we give a definition of extractors, a survey of their applications, and we discuss our improvements and the techniques that lead to them.

1.1 Definitions

We now give the formal definition of an extractor and state some previous results. We first need to define the notions of *min-entropy* and *statistical difference*.

We say that (the distribution of) a random variable X of range $\{0,1\}^n$ has min-entropy at least k if for every $x \in \{0,1\}^n$ it holds $\mathbf{Pr}[X = x] \leq 2^{-k}$. If k is an integer, then a canonical example of a distribution having min-entropy k is the uniform distribution over a set $S \subseteq \{0,1\}^n$ of cardinality 2^k. Indeed, it is implicit in [CG88] that if a distribution has min-entropy k then it is a convex combination of distributions each one of which is uniform over a set of size 2^k. We will consider

*An extended abstract of this paper is to appear in the Proceedings of the 31st ACM Symposium on Theory of Computing [Tre99].

†luca@cs.columbia.edu. Computer Science Department, Columbia University.

distributions of min-entropy k as the formalization of the notion of weak sources of randomness containing k "hidden" bits of randomness. The use of min-entropy to measure "hidden randomness" has been advocated by Chor and Goldreich [CG88] and, in full generality, by Zuckerman [Zuc90]. The statistical difference between two random variables X and Y with range $\{0,1\}^n$ is defined as

$$||X - Y|| = \max_{T:\{0,1\}^n \to \{0,1\}} |\,\mathbf{Pr}[T(X) = 1] - \mathbf{Pr}[T(Y) = 1]|$$

and we say that X and Y are ε-close if $||X - Y|| \leq \varepsilon$. For an integer l we denote by U_l a random variable that is uniform over $\{0,1\}^l$.

A function $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a (k,ε) extractor if for every random variable X of min entropy at least k it holds that $Ext(X, U_t)$ is ε-close to the uniform distribution over $\{0,1\}^m$. A weaker kind of combinatorial construction has also been considered: A function $Disp : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a (k,ε) disperser if for every subset $S \subseteq \{0,1\}^m$ such that $|S| > \varepsilon 2^m$ and for every X of min-entropy k it holds $\mathbf{Pr}[Disp(X, U_t) \in S] > 0$.

One would like to have, for every n and k, constructions where t is very small and m is as close to k as possible. There are some limitations towards this goal: One can show that, for a certain range of k, it must be the case that $t \geq \Omega(\log(n/\varepsilon))$ [NZ93], and also it must be the case that $m \leq k + t - 2\log(1/\varepsilon) + O(1)$ [RTS97]. It is possible to show (non-constructively) that for every n, k, ε, there is a (k,ε)-extractor $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ where $t = O(\log n/\varepsilon)$ and $m = k + t - 2\log(1/\varepsilon) - O(1)$. It is an open question to match such bounds with polynomial-time computable functions Ext.

1.2 Previous Work and Applications

The natural application of extractors is to allow the simulation of randomized algorithms even in (realistic) settings where only weak sources of randomness are available. This line of research has a long history, that dates back at least to von Neumann's algorithm for generating a sequence of unbiased bits from a source of biased but identically distributed and independent bits. More recent work by Santha and Vazirani [SV86] and Vazirani and Vazirani [VV85] considered much weaker sources of randomness (that they call "slightly random" sources) that are still sufficient to allow simulations of arbitrary randomized algorithms. These results were generalized by Chor and Goldreich [CG88] and Cohen and Wigderson [CW89], and finally by Zuckerman [Zuc90], who introduced the modern definition (based on min-entropy) of weak random sources and a construction of extractors (although the term *extractor* was coined later, in [NZ93]). Improved constructions of extractors appeared in [NZ93, SZ94, TS96, Zuc96b]. None of these constructions implies an optimal simulation of randomized algorithms. Dispersers are objects similar to, but less powerful than, extractors. Randomized algorithm having one-sided error probability can be simulated by using weak random sources and dispersers. Saks et al. [SSZ98] give a construction of dispersers that implies an optimal simulation of one-sided error randomized algorithms with weak random sources. Andreev et al. [ACRT97] show how to use the dispersers of Saks et al. in order to give optimal simulations of general randomized algorithms using weak random sources. The result of Andreev et al. leaves open the question of whether there exist a construction of extractors that is good enough to imply directly such optimal simulations.

Extractors are also used to derandomize randomized space-bounded computations [NZ93] and for randomness-efficient reduction of error in randomized algorithms (see [Zuc96b, GZ97] and references therein). They yield oblivious samplers (as defined in [BR94]), that have applications to interactive proofs and leader election in anonymous networks (see [Zuc96b] and references therein).

Reference	Min entropy k	Output length m	Additional randomness t	Type
[Zuc96b]	$k = \Omega(n)$	$m = (1-\alpha)k$	$t = O(\log n)$	Extractor
[TS96]	any k	$m = k$	$t = O((\log n)^9)$	Extractor
[TS96]	$k = n^{\Omega(1)}$	$m = k^{1-\alpha}$	$t = O(\log n \log \cdots \log n)$	Extractor
[SSZ98]	$k = n^{\Omega(1)}$	$m = k^{1-\alpha}$	$t = O(\log n)$	Disperser
[TS98]	any k	$m = k - \mathrm{poly}\log n$	$t = O(\log n)$	Disperser
This paper	$k = n^{\Omega(1)}$	$m = k^{1-\alpha}$	$t = O(\log n)$	Extractor
	any k	$m = k^{1-\alpha}$	$t = O((\log^2 n)/\log k)$	Extractor

Table 1: Parameters in previous constructions and our construction of (k, ε) extractors $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$. In the expressions, ε is fixed and arbitrarily small, and $\alpha > 0$ is an arbitrarily small constant. $O()$ notations hide dependencies on ε and α.

They also yield expander graphs, as discovered by Wigderson and Zuckerman [WZ93], that in turn have applications to superconcentrators, sorting in rounds, and routing in optical networks. Constructions of expanders via constructions of extractors and the Wigderson-Zuckerman connection appeared in [NZ93, SZ94, TS96], among others. Extractors can also be used to give simple proofs of certain complexity-theoretic results [GZ97], and to prove certain hardness of approximation results [Zuc96a]. An excellent introduction to extractors and their applications is given by a recent survey by Nisan [Nis96] (see also [NTS98]).

In Table 1 we summarize the best known constructions, for different combination of the parameters, and we state the parameters of two special cases of our construction.

1.3 Our Result

The extractors constructed in this paper work for any min-entropy $k = n^{\Omega(1)}$, extracts a slightly sub-linear fraction of the original randomness (i.e. the length of the output is $m = k^{1-\alpha}$ for an arbitrarily small α) and use $O(\log n)$ bits of true randomness. Indeed, a more general result holds, as formalized below.

Theorem 1 (Main) *There is an algorithm that on input parameters n, $k \leq n$, $m \leq k/2$, $0 < \varepsilon < 2^{-m/6}$, computes in $\mathrm{poly}(n,t)$ time a (k,ε)-extractor $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ where*

$$t = O\left(e^{\log k/(\log k/2m)} \frac{(\log n/\varepsilon)^2}{\log(k/2m)} \right) .$$

In particular, for any fixed constants $\varepsilon > 0$ and $0 < \gamma' < \gamma < 1$ we have for every n an explicit construction of an (n^γ, ε)-extractor $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{n^{\gamma'}}$.

Our construction improves on the construction of Saks, Srinivasan and Zhou [SSZ98] since we construct an extractor rather than a disperser, and improves over the constructions of Ta-Shma [TS96] since the additional randomness is logarithmic instead of slightly super-logarithmic. The best previous construction of extractors using $O(\log n)$ additional randomness was the one of Zuckerman [Zuc96b], that only works when the min-entropy is a constant fraction of the input length, while in our construction every min-entropy of the form n^γ is admissible. (On the other hand, the extractors of [Zuc96b] and [TS96] extract a constant factor of the entropy and the whole entropy, respectively, while we only extract a polynomial fraction.) Our construction yields a way of using weak random sources to simulate every randomized procedure that is optimal (among polynomial-time simulations) from the point of view of the rate of the entropy of the source as a function of n. In contrast to the result of [ACRT97] we can use a weak random source to generate almost uniformly distributed random bits independently of the purpose for which the random bits are to be used.[1]

Our construction is not yet the best possible, since we lose part of the randomness of the source and because the additional randomness is logarithmic only as long as $k = n^{\Omega(1)}$.

1.4 Techniques

This paper contains two main contributions.

The first one is a connection (outlined in Section 2) between pseudorandom generators of a certain kind and extractors. Our connection applies to certain pseudorandom generator constructions that are based on the (conjectured) existence of predicates (decision problems) that can be uniformly computed in time $t(n)$ but cannot be solved by circuits of size much smaller than $t(n)$. The analysis of such constructions shows that if the predicate is hard, then it is also hard to distinguish the output of the generator from the uniform distribution. This implication is proved by means of a reduction showing how a circuit that is able to distinguish the output of the generator from the uniform distribution can be transformed into a slightly larger circuit that computes the predicate. (Impagliazzo and Wigderson [IW97] present one such construction with very strong parameters.) Our result is that if the (truth table of the) predicate is chosen *randomly*, according to a distribution with sufficiently high min-entropy, then the output of the generator is *statistically close* to uniform. This statement is incomparable with standard analyses from scratch: we use a stronger assumption (that the predicate is *random* instead of *fixed and hard*) and prove a stronger conclusion (that the output is *statistically close* to, instead of *indistinguishable* from, the uniform distribution). An immediate application is that a pseudorandom generator construction of this kind *is* an extractor. Our result has a straightforward proof, based on a simple counting argument. The main contribution, indeed, is the *statement* of the result, rather than its *proof*, since it involves a new, more general, way of looking at pseudorandom generator constructions. The Impagliazzo-Wigderson generator, using our connection, is an extractor that beats some previous constructions and that is good enough to imply optimal simulations of randomized algorithms. We stress that even if the Impagliazzo-Wigderson generator is a pseudorandom generator under unproved conjectures, it *provably is* a good extractor (i.e. we do not use any complexity-theoretic assumption in our work).

Our second contribution is a construction that is simpler to describe and analyse (the generator of Impagliazzo and Wigderson is quite complicated) and that has somewhat better parameters. Our idea is to use a pseudorandom generator construction due to Nisan and Wigderson [NW94], that is considerably simpler than the one of Impagliazzo and Wigderson (indeed the construction of Impagliazzo and Wigderson contains the one of Nisan and Wigderson as one of its several components). The Nisan-Wigderson generator has weaker properties than the Impagliazzo-Wigderson generator, and our ideas outlined in Section 2 would not imply that it is an extractor as well. In Section 3 we show how to use error-correcting codes in order to turn the Nisan-Wigderson generator into a very good extractor. Section 3 contains a completely self-contained treatment of the construction and the analysis.

[1] Andreev et al. [ACRT97] show how to produce a sequence of bits that "look random" to a specific algorithm, and their construction works by having oracle access to the algorithm. So it is not possible to generate random bits "offline" before fixing the application where the bits will be used.

1.5 Relevance of Our Results

At the very least, our construction improves upon previous ones and solves the question of constructing extractors that use a logarithmic amount of randomness, work for any min-entropy that is polynomially related to the length of the input and have an output that is polynomially related to the amount of entropy. Such a construction has been considered a relevant open question (e.g. in [NTS98, Gol99]), even after Andreev et al. [ACRT97] showed that one does not need such extractors in order to develop an optimal simulation of randomized algorithms via weak random sources. Indeed, it was not clear whether the kind of approach introduced by Andreev et al. was *necessary* in order to have optimal simulations, or whether a more traditional approach based on extractors was still possible. Our result clarifies this point.

Perhaps more importantly, our construction is simpler to describe and analyse then the most recent previous constructions, and it uses a very different approach. Hopefully, our approach offer more room for improvement than previous, deeply exploited, ones. Raz et al. [RRV99] have already found improvements to our construction (see below). Tight results may come from some combination of our ideas and previous ones.

Our use of results about pseudorandomness in the construction of extractors may come as a surprise: pseudorandom generation deals with (and takes advantage of) a *computational* definition of randomness, while extractors are combinatorial objects used in a framework where *information-theoretic* randomness is being considered. In the past there have been some instances of results about computational randomness inspired by (typically trivial) information-theoretic analogs, e.g. the celebrated Yao's XOR Lemma and various kind of "direct product" results (see e.g. [GNW95]). On the other hand, it seemed "clear" that one could not go the other way, and have information-theoretic applications of computational results. This prejudice might be one reason why the connection discovered in this paper has been missed by the several people who worked on weak random sources and on pseudorandomness in the past decade (including those who did foundational work in both areas). Perhaps other important results might be proved along similar lines. On the other hand, it might also be that our results are just an isolated exception to the "rule" that computational randomness results are not useful in information theoretic settings. For example, we note in Section 4 that the pseudorandom generator construction of Blum, Micali and Yao [BM84, Yao82] does not yield an extractor using our techniques.

1.6 Later results

Shortly after the development of the results of this paper, Raz, Reingold and Vadhan [RRV99] devised an improvement to our construction. In our construction, if the input has min-entropy k and the output is required to be of length m, then the additional randomness is $O(m^{1/\log(k/2m)}(\log n)^2/\log(k/2m))$. In [RRV99], the dependency is $O((\log n)^2/\log(k/m))$. Raz et al. [RRV99] also show how to recursively compose their construction with itself (along the lines of [WZ93]) and they obtain in this way another construction where $k = m$ and the additional randomness is $O(\log^3 n)$. Constructions of extractors with parameters $k = m$ have applications to the explicit construction of expander graphs [WZ93]. In particular, Raz et al. [RRV99] present constructions of expander graphs and of superconcentrators that improve previous ones by Ta-Shma [TS96]. Raz et al. [RRV99] also improve the dependency that we have between additional randomness and error parameter ε.

Organization of the Paper

We present in Section 2 our connection between pseudorandom generator constructions and extractors. The main result of Section 2 is that the Impagliazzo-Wigderson generator [IW97] is a good extractor. In Section 3 we describe and analyse a simpler construction based on the Nisan-Wigderson generator [NW94] and on error correcting codes. Some concluding remarks are contained in Section 4. Section 3 might be read independently of Section 2.

2 The Connection Between Pseudorandom Generators and Extractors

This section describes our main idea of how to view a certain kind of pseudorandom generator as an extractor. Our presentation is specialized on the Impagliazzo-Wigderson generator, but results might be stated in a more general fashion. See the conference version of this paper [Tre99] for a more abstract presentation.

We start by defining the notion of computational indistinguishability, and pseudorandom generators, due to Blum, Goldwasser, Micali and Yao [GM84, BM84, Yao82].

We denote by U_n the uniform distribution over $\{0,1\}^n$. We say that two random variables X and Y with the same range $\{0,1\}^n$ are (S,ε)-indistinguishable if for every $T : \{0,1\}^n \to \{0,1\}$ computable by a circuit of size S it holds

$$|\,\mathbf{Pr}[T(X)=1] - \mathbf{Pr}[T(Y)=1]| \leq \varepsilon$$

One can see the notion of ε-closeness (that is when statistical difference is less than ε) as the "limit" of the notion of (S,ε)-indistinguishability for unbounded S.

Informally, a pseudorandom generator is an algorithm $G : \{0,1\}^t \to \{0,1\}^m$ where $t << m$ and $G(U_t)$ is (S,ε)-indistinguishable from U_m, with large S and small ε. In complexity theory, one looks for generators, say, $G : \{0,1\}^{O(\log m)} \to \{0,1\}^m$ where $G(U_{O(\log m)})$ is $(O(m), 1/3)$-indistinguishable from U_m. Such generators (if they were uniformly computable in time poly(m)) would imply deterministic polynomial-time simulations of randomized polynomial-time algorithms.

Given current techniques, all interesting constructions of pseudorandom generators have to rely on complexity-theoretic conjectures. For example the Blum-Micali-Yao [BM84, Yao82] construction (that has different parameters from the ones exemplified above) requires the existence of strong one-way permutations. In a line of work initiated by Nisan and Wigderson [NW94], there have been results showing that the existence of hard-on-average decision problems is sufficient to construct pseudorandom generators. Impagliazzo and Wigderson [IW97] present a construction that only requires the existence of decision problems having high worst-case complexity.[2]

The main result of Impagliazzo and Wigderson is that if there is a decision problem solvable in time $2^{O(n)}$ that cannot be solved by circuits of size less than $2^{\gamma n}$, for some $\gamma > 0$, then $P = BPP$, i.e. every randomized polynomial time algorithm has a deterministic polynomial-time simulation. A precise statement of the result of Impagliazzo and Wigderson follows.

[2] This is an oversimplified account. Both [NW94] and [IW97] require a *non-uniform* kind of hardness, and recent work has concentrated on *uniform* conditions [IW98]. A construction that only needed worst-case non-uniform assumptions was already given by Babai et al. [BFNW93], but the parameters were worse than in the later construction of Impagliazzo and Wigderson [IW97].

Theorem 2 ([IW97]) *Suppose that there exists a family $\{P_l\}_{l\geq 0}$ of predicates $P_l : \{0,1\}^l \to \{0,1\}$ that is decidable in time $2^{O(l)}$, and a constant $\gamma > 0$ such that, for every sufficiently large l, P_l has circuit complexity at least $2^{\gamma l}$.*

Then there exists a pseudorandom generator $IW^{(m)} : \{0,1\}^{O(\log m)} \to \{0,1\}^m$ computable in time $\text{poly}(m)$ such that $IW^{(m)}(U_{O(\log m)})$ is $(O(m), 1/3)$-indistinguishable from the uniform distribution, and $P = BPP$.

Results about pseudorandomness are typically proved by contradiction. Impagliazzo and Wigderson prove Theorem 2 by establishing the following result.

Lemma 3 ([IW97]) *For every pair of constants $\varepsilon > 0$ and $\delta > 0$ there exists an algorithm that on input a length parameter l and having oracle access to a predicate $P : \{0,1\}^l \to \{0,1\}$ computes a function $IW_P : \{0,1\}^t \to \{0,1\}^m$ in $\text{poly}(m)$ time, where $t = O(l)$ and $m = 2^{\delta l}$ such that if $T : \{0,1\}^m \to \{0,1\}$ statisfies*

$$|\,\mathbf{Pr}[T(IW_P(U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| > \varepsilon$$

then P is computed by a circuit A that uses T-gates and whose size is at most $2^{c_\varepsilon \delta l}$, where c_ε is a constant that depends only on ε.

By a "circuit with T-gates" we mean a circuit that can use ordinary fan-in-2 AND and OR gates and fan-in-1 NOT gates, as well as a special gate (of fan-in m) that computes T with unit cost. This is the non-uniform analog of a computation that makes oracle access to T.

It might not be immediate to see how Theorem 2 follows from Lemma 3. The idea is that if we have a predicate $P : \{0,1\}^l \to \{0,1\}$ that cannot be computed by circuits of size $2^{(c_{1/3}+1)\delta l}$, then $IW_P(U_t)$ is $(2^\delta, 1/3)$-indistinguishable from uniform. This can be proved by contradiction: if T is computed by a circuit of size $2^{\delta l}$ and is such that

$$|\,\mathbf{Pr}[T(IW_P(U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| > 1/3$$

then there exists a circuit A of size at most $2^{c_{1/3}\delta l}$ that uses T-gates such that A computes P. If each T-gate is replaced by the circuit that computes T, we end up with a circuit of size at most $2^{(c_{1/3}+1)\delta l}$ that computes P, a contradiction to our initial assumption.

We stress that Lemma 3 has not been stated in this form by Impagliazzo and Wigderson [IW97]. In particular, the fact that their analysis applies to every function T, *regardless of its circuit-complexity*, was not explicitely stated. On the other hand, this is not a peculiar or surprising property of the Impagliazzo-Wigderson construction: in general in complexity theory and in cryptography the correctness of a transformation of an object with certain assumed properties (e.g. a predicate with large circuit complexity) into an object with other properties (e.g. a generator whose output is indistinguishable from uniform) is always proved by "black-box" reductions, that work by making "oracle access" to the object and making no assumptions about it. The fact that almost all known analysis techniques yield "black-box" reductions motivates results such as the one by Impagliazzo and Rudich [IR90], who show that such reductions cannot tansform a one-way permutation into a trapdoor permutation (unless the analysis of the reduction also contains a proof that $P \neq NP$).

We also mention that two recent papers exploit the hidden generality of the pseudorandom generator construction of Impagliazzo and Wigderson, and of the earlier one by Nisan and Wigderson [NW94]. Arvind and Köbler [AK97] observe that the analysis of the Nisan-Wigderson generator extends to "non-deterministic circuits," which implies the existence of pseudorandom generators for non-deterministic computations, under certain assumptions. Klivans and Van Melkebeek [KvM99] observe similar generalizations of the Impagliazzo-Wigderson generator for arbitrary non-uniform complexity measures having certain closure properties (which does not include non-deterministic circuit size, but includes the related measure "size of circuits having an NP-oracle").

A novel aspect in our view (that is not explicit in [IW97, AK97, KvM99]) is to see the Impagliazzo-Wigderson construction as an algorithm that takes two inputs: the truth-table of a predicate and a seed. The Impagliazzo-Wigderson analysis says something meaningful even when the predicate is not *fixed and hard* (for an appropriate complexity measure), but rather *supplied dynamically* to the generator. In the rest of this section, we will see that if the (truth table of the) predicate is sampled from a weak random source of sufficiently large min-entropy, then the output of the Impagliazzo-Wigderson generator is statistically close to uniform: that is, the Impagliazzo-Wigderson generator *is* an extractor.

Let us introduce the following additional piece of notation: for a string $x \in \{0,1\}^{2^l}$ we denote by $< x >: \{0,1\}^l \to \{0,1\}$ the predicate whose truth-table is x. Then consider the function $Ext : \{0,1\}^{2^l} \times \{0,1\}^t \to \{0,1\}^m$ defined as

$$Ext(x,s) = IW_{<x>}(s)$$

Now we claim that it is an extractor, indeed a $(mc\delta l 2^{c\delta l} + \log 1/\varepsilon, 2\varepsilon)$-extractor, where $c = c_\varepsilon$.

Before delving into the proof, we give a rough overview: suppose for the sake of contradiction that for some random variable X of large min-entropy the distribution $Ext(X, U_t)$ is far from uniform. Then there exists a test T that distinguishes $Ext(X, U_t)$ from uniform, that is

$$\mathbf{Pr}[T(Ext(X,U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]$$

is noticeably large. So we also have that there is a noticeable probability of sampling an x from X so that

$$\mathbf{Pr}[T(Ext(x,U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1] \tag{1}$$

is large. It follows from the analysis of the Impagliazzo-Wigderson generator that each such x is computed by a small circuit of size, say, S. Since there are about 2^S such circuits, there are at most that many values of x for which the difference in Expression (1) can be large. If the entropy of X is $>> s$, then the probability that one of them is sampled has to be small, and we reach a contradiction.

We now give a detailed proof. We have to prove that for every random variable X of min entropy $k \geq m\delta l 2^{c\delta l} + \log 1/\varepsilon$ and for every $T : \{0,1\}^m \to \{0,1\}$ we have

$$|\,\mathbf{Pr}[T(Ext(X,U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| \leq 2\varepsilon \tag{2}$$

Let us fix X and T and prove that Expression (2) holds for them. Let us call $B \subseteq \{0,1\}^{2^l}$ the set of *bad* strings x for which

$$|\,\mathbf{Pr}[T(Ext(x,U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| > \varepsilon \tag{3}$$

For each such x, the analysis of Impagliazzo and Wigderson implies that there is a circuit of size $2^{c\delta l}$ that uses T-gates and that computes $< x >$. Since T is fixed, and any two different predicates are computed by two different circuits, we can conclude that the total number of elements of B is at most the number of circuits of size $S = 2^{c\delta l}$ with gates of fan-in at most m. So we have $|B| \leq 2^{mS \log S}$.

Since X has high min-entropy, and B is small, the probability of picking an element of B when sampling from X is small, that is

$$\mathbf{Pr}[X \in B] \leq |B| 2^{-k} \leq 2^{mS \log S} \cdot 2^{-(mS \log S + \log 1/\varepsilon)} = \varepsilon \ . \tag{4}$$

Then we have

$$\begin{aligned} |\, \mathbf{Pr}[T(Ext(X, U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| &\leq \mathop{\mathbf{E}}_{x \leftarrow X}[|\, \mathbf{Pr}[T(Ext(x, U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]|] \\ &\leq 2\varepsilon \end{aligned}$$

where the first inequality is an application of the triangle inequality and the second follows from Expression (4) and the definition of B by using a Markov argument.

If we translate the parameters in a more traditional format we have the following extractor construction.

Theorem 4 *For every positive constants δ and ε there is a $\gamma > 0$ and an explicit construction of (k, ε)-extractor $Ext_n : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ where $t = O(\log n)$, $k = n^\delta$ and $m = k^\gamma$.*

This is already a very good construction of extractors, and it implies an optimal simulation of randomized algorithms using weak random sources.[3] By deriving a more accurate estimation of the parameters in the Impagliazzo-Wigderson construction, it would be possible to improve on the statement of Theorem 4. More specifically, it could be possible to have an explicit dependency of the parameter t from δ and ε, and an explicit expression for c_ε. However, such improved analysis would not be better than the analysis of the construction that we present in the next section, and so we do not pursue such direction.

3 Main Result

In this section we present a construction of extractors based on the Nisan-Wigderson generator and error-correcting codes.

The advantages of the construction of this section over the construction of the previous section are better quantitative parameters and the possibility of giving a self-contained and relatively simple presentation. The subsequent work of Raz et al [RRV99] took the construction of this section as a starting point, and improved the primitives that we use.

3.1 Overview

The Nisan-Wigderson generator works similarly to the Impagliazzo-Wigderson one: it has access to a predicate, and its output is indistinguishable from uniform provided that the predicate is hard.[4] This is proved by means of a reduction that shows that if T is a test that distinguishes the output of the generator with predicate P from uniform, then there is a small circuit with one T-gate that *approximately* computes P.[5]

[3] The simulation would be optimal from the point of view of the "entropy rate" of the source; the slowdown in the simulation can be reduced substantially — even if by no more than a polynomial factor. The extractors presented in the next section give an improvement in this direction.

[4] But, as will be explained in a moment, a different notion of hardness is being used.

[5] That is, the circuit computes a predicate that agrees with P on a fraction of inputs noticeably bounded away from 1/2.

Due to this analysis, we can say that the output of the Nisan-Wigderson generator is indistinguishable from uniform provided that the predicate being used is hard to compute approximately, as opposed to be merely hard to compute exactly, as in the case of the Impagliazzo-Wigderson generator.

We may be tempted to defined and analyse an extractor Ext based on the Nisan-Wigderson generator using exactly the same approach of the previous section. Then, as before, we would assume for the sake of contradiction that a test T distinguishes the output $Ext(X, U_t)$ of the extractor from the uniform distribution, and we would look at how many strings x are there such that $|\,\mathbf{Pr}[T(Ext(x, U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]|$ can be large. For each such x we can say that there is a circuit of size S that describes a string *whose Hamming distance from* x is less than $1/2$. Since there are about 2^S such circuits, the total number of bad strings x is at most 2^S times the number of strings that can belong to a Hamming sphere of radius close to $1/2$. If X is sampled from a distribution whose min-entropy is much bigger then the logarithm of the number of possible bad x, then we reach a contradiction to the assumption that T was distinguishing $Ext(X, U_t)$ from uniform.

When this calculation is done with the actual parameters of the Nisan-Wigderson generator, the result is very bad, because the number of strings that belong to a Hamming sphere of the proper radius is huge. This is also the *only* point where the approach of the previous section breaks down.

Our solution is to use *error-correcting codes*, specifically, codes with the property that every Hamming sphere of a certain radius contains a small number of codewords. We then define an extractor $Ext()$ that on input x and s first encodes x using the error-correcting code, and then applies the Nisan-Wigderson generator using s as a seed and the encoding of x as the truth table of the predicate.

Thanks to the property of the error-correcting code, the counting argument of the previous section works well again.

3.2 Preliminaries

In this section we state some known technical results that will be used in the analysis of our extractor. For an integer n we denote by $[n]$ the set $\{1, \ldots, n\}$. We denote by $u_1 \cdot u_2$ the string obtained by concatenating the strings u_1 and u_2.

Lemma 5 (Error Correcting Codes) *For every n and δ there is an efficient encoding $EC : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ where $\bar{n} = \mathrm{poly}(n, 1/\delta)$ such that every ball of Hamming radius $(1/2 - \delta)\bar{n}$ in $\{0,1\}^{\bar{n}}$ contains at most $1/\delta^2$ codewords. Furthermore $\bar{n}$ can be assumed to be a power of 2.*

Stronger parameters are achievable. In particular the length of the encoding can be $\bar{n} = n\mathrm{poly}(1/\delta)$. However, the stronger bounds would not improve our constructions. Standard codes are good enough to prove Lemma 5. We sketch a proof of the lemma in the Appendix.

Lemma 6 (Design [NW94]) *For every positive integers m, $a \le m$ and l there exists an efficiently constructible family of sets $\mathcal{S} = S_1, \ldots, S_m$ such that*

- $S_i \subseteq [t]$, *where* $t = O(e^{(\log m)/a} l^2/a)$
- $|S_i| = l$
- $|S_i \cap S_j| \le a$.

The family $\mathcal{S}$ will be called an (m, l, a)-design.

Lemma 6 was proved in [NW94] for the special case of $a = \log m$. The general case follows using the same proof, but a little care is required while doing a certain probabilistic argument (one has to choose the right Chernoff bound). A sketch of the proof is given in the Appendix.

The following notation will be useful in the next definition: if $S \subseteq [t]$, with $S = \{s_1, \ldots, s_l\}$ (where $s_1 < s_2 < \cdots < s_l$) and $y \in \{0,1\}^t$, then we denote by $y_{|S} \in \{0,1\}^l$ the string $y_{s_1} \cdot y_{s_2} \cdots y_{s_l}$.

Definition 7 (NW Generator [NW94]) *For a function $f : \{0,1\}^l \to \{0,1\}$ and an (m,l,a)-design $\mathcal{S} = (S_1, \ldots, S_m)$, the Nisan-Wigderson generator $NW_{f,\mathcal{S}} : \{0,1\}^t \to \{0,1\}^m$ is defined as*

$$NW_{f,\mathcal{S}}(y) = f(y_{|S_1}) \cdots f(y_{|S_m})$$

For two functions $f, g : \{0,1\}^l \to \{0,1\}$ and a number $0 \leq \rho \leq 1$ we say that g *approximates* f within a factor ρ if f and g agree on at least a fraction ρ of their domain, i.e. $\mathbf{Pr}_x[f(x) = g(x)] \geq \rho$.

The following result is similar to Lemma 2.4 in [NW94].

Lemma 8 (Analysis of the NW Generator) *Let $\mathcal{S}$ be an (m,l,a)-design, and $T : \{0,1\}^m \to \{0,1\}$. Then there exists a family $\mathcal{G}_T$ of at most 2^{m2^a} functions such that for every function $f : \{0,1\}^l \to \{0,1\}$ satisfying*

$$|\mathbf{Pr}_{y \in \{0,1\}^d}[T(NW_{f,\mathcal{S}}(y)) = 1] - \mathbf{Pr}_{r \in \{0,1\}^m}[T(r) = 1]| \geq \varepsilon \ .$$

there exists a function $g : \{0,1\}^l \to \{0,1\}^m$, $g \in \mathcal{G}_T$, such that either $T(g(\cdot))$ or its complement approximates $f(\cdot)$ within $1/2 + \varepsilon/m$.

The following result will be used in the proof of Lemma 8. It is typically attributed to Yao [Yao82].

Lemma 9 (Distinguishability versus Predictability) *Let $T : \{0,1\}^m \to \{0,1\}$, $g : \{0,1\}^l \to \{0,1\}^{m-1}$, $f : \{0,1\}^l \to \{0,1\}$ and $\varepsilon > 0$; if*

$$\left| \mathbf{Pr}_{x \in \{0,1\}^l}[T(g(x), f(x)) = 1] - \mathbf{Pr}_{x \in \{0,1\}^l, r \in \{0,1\}}[T(g(x), r) = 1] \right| \geq \varepsilon$$

then there exist two bits $b_0, b_1 \in \{0,1\}$ such that the function $b_0 \oplus T(g(x), b_1)$ agrees with $f(x)$ on at least a fraction $1/2 + \varepsilon$ of the inputs.

We now prove Lemma 8.

Proof: [Of Lemma 8] We follow the proof of Lemma 2.4 in [NW94]. The main idea is that if T distinguishes $NW_{f,\mathcal{S}}(\cdot)$ from the uniform distribution, then we can find a bit of the output where this distinction is noticeable, and then we will apply Lemma 9. In order to find the "right bit", we will use the so-called *hybrid argument* of Goldwasser and Micali [GM84]. We define $m + 1$ distributions $D_0, \ldots, D_m$; D_i is defined as follows: sample a string $v = NW_{f.\mathcal{S}}(y)$ for a random y, and then sample a string $r \in \{0,1\}^m$ according to the uniform distribution, then concatenate the first i bits of v with the last $m - i$ bits of r. By definition, D_0 is distributed as $NW_{f.\mathcal{S}}(y)$ and

D_m is the uniform distribution over $\{0,1\}^m$. Using the hypothesis of the Lemma and the triangle inequality we have

$$\begin{aligned}
\varepsilon &\leq |\mathbf{Pr}_y[T(NW_{f.\mathcal{S}}(y)) = 1] - \mathbf{Pr}_r[T(r)]| \\
&= |\mathbf{Pr}[T(D_0) = 1] - \mathbf{Pr}[T(D_m) = 1]| \\
&= |\sum_{i=0}^{m-1} (\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1])| \\
&\leq \sum_{i=0}^{m-1} |\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1]|
\end{aligned}$$

In particular, there exists an index i such that

$$|\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1]| \geq \varepsilon/m \tag{5}$$

and there exists a bit $b \in \{0,1\}$ such that

$$\mathbf{Pr}[b \oplus T(D_i) = 1] - \mathbf{Pr}[b \oplus T(D_{i+1}) = 1] \geq \varepsilon/m \tag{6}$$

Now, recall that

$$D_i = f(y_{|S_1}) \cdots f(y_{|S_{i-1}}) r_i r_{i+1} \cdot r_m$$

and

$$D_{i+1} = f(y_{|S_1}) \cdots f(y_{|S_{i-1}}) f(y_{|S_i}) r_{i+1} \cdot r_m \ .$$

We can use an averaging argument to claim that we can fix $r_{i+1}, \ldots, r_m$ to some values $c_{i+1} \cdots c_m$, as well as all the bits of y except those in S_i, and still have an expression like (6). So, we have the relation

$$\mathbf{Pr}[b \oplus T(g_1(x) \cdots g_{i-1}(x) r_i c_{i+1} \cdots c_m) = 1] -$$
$$\mathbf{Pr}[b \oplus T(g_1(x) \cdots g_{i-1}(x) f(x) c_{i+1} \cdots c_m) = 1] \geq \varepsilon/m$$

where $g_j(x)$ is $f(y_{|S_j})$ and y is the string whose bits in S_i are fixed according to x, and whose other bits had been set non-uniformly. Since, by the property of the sets $S_1, \ldots, S_m$, every set S_j contains at most a elements of S_i, it follows that $g_j(x)$ depends on at most a bits of its input and therefore it is totally specified by at most 2^a bits. We can now apply Lemma 9 and we have that $b_0 \oplus b \oplus T(g_1(x) \cdots g_{i-1}(x) b_1 c_{i+1} \cdots c_m)$ agrees with f on a fraction $1/2 + \varepsilon/m$ of the inputs. The m-tuple $(g_1(x) \cdots g_{i-1}(x) b_1 c_{i+1} \cdots c_m)$ defines a function g that satisfies the statement of the lemma, and that is entirely specified given at most $m2^a$ bits of information. This is why $\mathcal{G}_T$, that is the set of all such functions over all possible f, contains at most 2^{m2^a} functions. □

3.3 Construction

The construction has parameters n, $k \leq n$, $m \leq k/2$ and $\varepsilon > 0$. We assume that $1 + 3\log(1/\varepsilon) + 2\log m \leq m$, which is true if $m \geq 16$ and $\varepsilon \geq 2^{-m/6}$. This simplifies the expressions below, but is not really necessary for the sake of the construction.

Let $EC : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ be as in Lemma 5, with $\delta = \varepsilon/m$, so that $\bar{n} = \text{poly}(n, 1/\varepsilon)$, and define $l = \log \bar{n} = O(\log n/\varepsilon)$.

For an element $u \in \{0,1\}^n$, define $\bar{u} =< EC(u) >: \{0,1\}^l \to \{0,1\}$. Let $\mathcal{S} = S_1, \ldots, S_m$ be as in Lemma 6, such that

- $S_i \subseteq [t]$,
- $|S_i| = l$,
- $|S_i \cap S_j| \leq a = \log(k/2m)$, and
- $t = O\left(l^2 e^{(\log m)/\log k/2m} \frac{1}{\log(k/2m)}\right)$.

Then we define $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ as

$$Ext(u, y) = NW_{\bar{u},\mathcal{S}}(y) = \bar{u}(y_{|S_1}) \cdots \bar{u}(y_{|S_m}) \ .$$

3.4 Analysis

Lemma 10 *For every fixed predicate $T : \{0,1\}^m \to \{0,1\}$, there are at most $2^{1+m2^a} \cdot (m/\varepsilon)^2$ strings $u \in \{0,1\}^n$ such that*

$$|\mathbf{Pr}[T(Ext(u, U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| \geq \varepsilon \tag{7}$$

PROOF: It follows from the definition of Ext and from Lemma 8 that if u is such that (7) holds, then there exists a function $g : \{0,1\}^l \to \{0,1\}^m$ in $\mathcal{G}_T$ and a bit $b \in \{0,1\}$ such that the function $b \oplus T(g(\cdot))$ approximates $\bar{u}(\cdot)$ within $1/2 - \varepsilon/m = 1/2 - \delta$.

There are at most 2^{m2^a} functions $g \in \mathcal{G}_T$, furthermore, each such function can be within relative distance $1/2 - \varepsilon/m$ from at most $(m/\varepsilon)^2$ functions $\bar{u}(\cdot)$ coming from the error correcting code of Lemma 5.

We conclude that $2(m/\varepsilon)^2 2^{m2^a}$ is an upper bound on the number of strings u for which Expression (7) can occur. □

Theorem 11 *Ext as described above is a $(k, 2\varepsilon)$-extractor.*

PROOF: Fix a predicate $T : \{0,1\}^m \to \{0,1\}$. From Lemma 10 we have that the probability that sampling a u from a source X of min-entropy k we can have

$$|\,\mathbf{Pr}[T(Ext(u, U_t)) = 1] - \mathbf{Pr}_r[T(U_m) = 1]| \geq \varepsilon$$

is at most $2^{1+m2^a} \cdot \frac{m^2}{\varepsilon^2} \cdot 2^{-k}$ which is at most ε by our choice of parameters. A Markov argument shows that

$$|\,\mathbf{Pr}[T(Ext(X, U_t)) = 1] - \mathbf{Pr}[T(U_m) = 1]| \leq 2\varepsilon$$

□

4 Final Remarks

The idea of applying results on pseudorandomness to the context of information-theoretic randomness was inspired by previous work of Andreev et al. [ACRT97]. The use of error-correcting codes was inspired by an alternative proof of the results of [IW97] due to Sudan et al. [STV99].

Both the error correcting codes of Lemma 5 and the design of Lemma 6 can be constructed in logarithmic space. The construction of designs in logarithmic space requires a logarithmic amount of randomness, and only succeeds with high probability (see [IW97] and also [AR98, Section 5] for details), but both these limitations are not a problem in our construction, since the randomness can be taken from the seed, and a small error probability only contributes to a slight increase of the final statistical difference from the uniform distribution. Therefore, our extractors can be constructed in logarithmic space, unlike the dispersers of [SSZ98, TS98] and the extractors of [TS96].

One may wonder whether the Blum-Micali-Yao [BM84, Yao82] pseudorandom generator construction can be seen as an extractor. We propose an abstract view of their construction and see why our approach does not prove that it is an extractor.

Their construction (using the Goldreich-Levin predicate [GL89]) can be seen as an algorithm that given parameters l, m and having oracle access to a permutation $\pi : \{0,1\}^l \to \{0,1\}^l$ computes a generator $BMY_\pi : \{0,1\}^{2l} \to \{0,1\}^m$, with the property that if there exists a function $T : \{0,1\}^m \to \{0,1\}$ such that

$$|\,\mathbf{Pr}[T(BMY_\pi(U_{2l})) = 1] - \mathbf{Pr}[T(U_m) = 1]| > \varepsilon$$

then there exists a circuit A of size $\mathrm{poly}(ml/\varepsilon)$ that, with "oracle access" to T *and* π, computes the *inverse* of π on a fraction $\mathrm{poly}(\varepsilon/ml)$ of the inputs. This property is of no use in our framework: suppose that π is generated according to a distribution with a certain min-entropy, and that we are trying to prove that the number of permutations π for which

$$|\,\mathbf{Pr}[T(BMY_\pi(U_{2l})) = 1] - \mathbf{Pr}[T(U_m) = 1]| > \varepsilon$$

is small. Then we cannot upper bound the number of such π by the number of circuits that *having access to* π compute π^{-1}. From an information-theoretic viewpoint, there is no difference between computing π and π^{-1}. In fact, this is the reason why cryptography is based on complexity theory and not on information theory, and also why most people take it for granted that results about computational randomness do not have interesting applications in information theory.

We stress that, at this point, we are not saying that the Blum-Micali-Yao generator is definitely not an extractor, but only that our techniques do not prove that it is an extractor. We believe that one can find a distribution Π of permutations having large min-entropy and such that the Blum-Micali-Yao generator applied to a random seed and to a permutation sampled from Π is not close to the uniform distribution. It would be nice to prove such a result.

Acknowledgments

Oded Goldreich contributed an important idea in a critical phase of this research; he also contributed very valuable suggestions on how to present the results of this paper. I thank Oded Goldreich, Madhu Sudan, Salil Vadhan, Amnon Ta-Shma, and Avi Wigderson for several helpful conversations. This paper would have not been possible without the help of Adam Klivans, Danny Lewin, Salil Vadhan, Yevgeny Dodis, Venkatesan Guruswami, and Amit Sahai in assimilating the ideas of [NW94, BFNW93, Imp95, IW97]. Thanks also to Madhu Sudan for hosting our reading group in the Spring'98 Complexity Seminars at MIT. I thank Shafi Goldwasser for asking about the Blum-Micali-Yao generator, and for not being satisfied by my initial answer.

This work was mostly done while the author was at MIT, partially supported by a grant of the Italian CNR. Part of this work was also done while the author was at DIMACS, supported by a DIMACS post-doctoral fellowship.

References

[ACRT97] A.E. Andreev, A.E.F. Clementi, J.D.P. Rolim, and L. Trevisan. Weak random sources, hitting sets, and BPP simulations. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 264–272, 1997.

[AK97] V. Arvind and J. Köbler. On resource-bounded measure and pseudorandomness. In *Proceedings of the 17th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 235–249. LNCS 1346, Springer-Verlag, 1997.

[AR98] E. Allender and K. Reinhardt. Isolation, matching, and counting. Technical Report TR98-019, Electronic Colloquium on Computational Complexity, 1998.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998. Preliminary version in *Proc. of FOCS'95*.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. Preliminary version in *Proc. of FOCS'82*.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 276–287, 1994.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[CW89] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. Preliminary Version in *Proc. of STOC'82*.

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95-50, Electronic Colloquium on Computational Complexity, 1995.

[Gol99] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Springer-Verlag, 1999.

[GZ97] O. Goldreich and D. Zuckerman. Another proof that $BPP \subseteq PH$ (and more). Technical Report TR97-045, Electronic Colloquium on Computational Complexity, 1997.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[IR90] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of CRYPTO'88*, pages 8–26. LNCS 403, Springer-Verlag, 1990.

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ unless E has sub-exponential circuits. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[IW98] R. Impagliazzo and A. Wigderson. Randomness versus time: De-randomization under a uniform assumption. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, pages 734–743, 1998.

[KvM99] A. Klivans and D. van Milkebeek. Graph non-isomorphism has subexponential size proofs unless the polynomial hierarchy collapses. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, 1999.

[MS77] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[Nis96] N. Nisan. Extracting randomness: How and why. In *Proceedings of the 11th IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[NTS98] N. Nisan and A. Ta-Shma. Extrating randomness : A survey and new constructions. *Journal of Computer and System Sciences*, 1998. To appear. Preliminary versions in [Nis96, TS96].

[NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994. Preliminary version in *Proc. of FOCS'88*.

[NZ93] N. Nisan and D. Zuckerman. More deterministic simulation in Logspace. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[RRV99] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, 1999.

[RTS97] J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 585–594, 1997.

[SSZ98] M. Saks, A. Srinivasan, and S. Zhou. Explicit OR-dispersers with polylogarithmic degree. *Journal of the ACM*, 45(1):123–154, 1998. Preliminary version in *Proc. of STOC'95*.

[STV99] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, 1999.

[SV86] M. Santha and U. Vazirani. Generating quasi-random sequences from slightly random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[SZ94] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 264–275, 1994.

[Tre99] L. Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, 1999.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 276–285, 1996.

[TS98] A. Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[VV85] U. Vazirani and V. Vazirani. Random polynomial time is equal to slightly random polynomial time. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 417–428, 1985.

[WZ93] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 245–251, 1993.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23th IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[Zuc90] D. Zuckerman. General weak random sources. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 534–543, 1990.

[Zuc96a] D. Zuckerman. On unapproximable versions of *NP*-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996. Preliminary Version in *Proc. of Structures'93*.

[Zuc96b] D. Zuckerman. Randomness-optimal sampling, extractors and constructive leader election. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 286–295, 1996.

A Appendix

A.1 A Discussion on Lemma 5

It is a standard result that if an error-correcting code has large minimum distance then there can be few codewords in every large ball. In particular, the following bound holds.

Lemma 12 *Suppose C is an error-correcting code with (relative) minimum distance $\geq 1/2 - \beta/2$. Then every Hamming ball of (relative) radius $1/2 - \sqrt{\beta}$ contains at most $1/3\beta$ codewords.*

A proof can be found e.g. in [BGS98, Lemma A.1]. The following result is well known, even if we do not know of a source where it is clearly stated in this way.

Lemma 13 *For every δ and n there exists an error-correcting code with 2^n codewords of length $\bar{n} = \mathrm{poly}(n, 1/\delta)$ and with minimum distance $(1/2 - \delta)\bar{n}$. The code admits a polynomial-time encoding algorithm.*

Several constructions meet this requirement. In particular one can use a Reed-Solomon code concatenated with a Hadamard code. See e.g. [MS77] for a treatment of error correcting codes. Lemma 5 follows from Lemmas 12 and 13.

A.2 A Sketch of the Proof of Lemma 6

The following version of the Chernoff bound will be used.

Lemma 14 *Let $X_1, \ldots, X_n$ be 0/1 mutually independent random variables such that $\mathbf{E}\left[\sum_i X_i\right] = \mu$. Then, for every $\alpha > 1$ it holds*

$$\mathbf{Pr}\left[\sum_i X_i \geq \alpha\mu\right] \leq e^{-((\ln \alpha)+\frac{1}{\alpha}-1)\alpha\mu}$$

This bound is proved in the usual way, but we have not find a standard reference for this particular statement, so we present below a proof (taken from lecture notes by Leighton and Vempala).
PROOF: Let $p = \mathbf{Pr}[X_i = 1] = \mathbf{E}[X_i] = \mu/n$. Then

$$\begin{aligned}
\mathbf{Pr}\left[\sum_i X_i \geq \alpha\mu\right] &= \mathbf{Pr}\left[\alpha^{\sum_i X_i} \geq \alpha^{\alpha\mu}\right] \\
&\leq \frac{\mathbf{E}\left[\alpha^{\sum_i X_i}\right]}{\alpha^{\alpha\mu}} \\
&= \frac{\prod_i \mathbf{E}\left[\alpha^{X_i}\right]}{\alpha^{\alpha\mu}} \\
&= \frac{(1-p+p\alpha)^n}{\alpha^{\alpha\mu}} \\
&\leq \frac{e^{(-p+p\alpha)n}}{\alpha^{\alpha\mu}} \\
&= e^{-\mu+\alpha\mu-\alpha\mu\ln\alpha}
\end{aligned}$$

The second step uses Markov, the third uses independence, and the fifth uses the inequality $(1-x) \leq e^{-x}$. □

We can now sketch the proof of Lemma 6 as it was carried on in [NW94].

PROOF:[Of Lemma 6] Sequentially choose m subsets of $[t]$ such that any of the chosen subsets intersects the previously chosen ones in less than a points. A probabilistic argument using the above Chernoff bound shows that the algorithm is always able to choose a new subset as long as the total number of sets in no more than m (in the probabilistic argument we will choose a multi-set of elements, so as to be able to use the Chernoff bound, and then we will discard duplicates.) □