Extracting Randomness from Samplable Distributions

Luca Trevisan * Salil Vadhan†

Abstract

The standard notion of a randomness extractor *is a procedure which converts any weak source of randomness into an almost uniform distribution. The conversion necessarily uses a small amount of pure randomness, which can be eliminated by complete enumeration in some, but not all, applications.*

Here, we consider the problem of deterministically *converting a weak source of randomness into an almost uniform distribution. Previously, deterministic extraction procedures were known only for sources satisfying strong independence requirements. In this paper, we look at sources which are* samplable*, i.e. can be generated by an efficient sampling algorithm. We seek an efficient deterministic procedure that, given a sample from any samplable distribution of sufficiently large min-entropy, gives an almost uniformly distributed output. We explore the conditions under which such* deterministic extractors *exist.*

We observe that no deterministic extractor exists if the sampler is allowed to use more computational resources than the extractor. On the other hand, if the extractor is allowed (polynomially) more resources than the sampler, we show that deterministic extraction becomes possible. This is true unconditionally in the nonuniform setting (i.e., when the extractor can be computed by a small circuit), and (necessarily) relies on complexity assumptions in the uniform setting.

One of our uniform constructions is as follows: assuming that there are problems in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ *that are not solvable by subexponential-size circuits with* $\mathbf{\Sigma_6}$ *gates, there is an efficient extractor that transforms any samplable distribution of length* n *and min-entropy* $(1-\gamma)n$ *into an output distribution of length* $(1-O(\gamma)n)$*, where* γ *is any sufficiently small constant. The running time of the extractor is polynomial in* n *and the circuit complexity of the sampler. These extractors are based on a connection between deterministic extraction from samplable distributions and hardness against nondeterministic circuits, and on the use of nondeterminism to substantially speed up "list decoding" algorithms for error-correcting codes such as multivariate polynomial codes and Hadamard-like codes.*

1 Introduction

Randomness has proved to be a very useful tool in computer science. In algorithms, it has yielded the only known polynomial-time solutions for some problems, such as primality testing [SS77, Mil76, Rab80] and certain approximate counting problems [KLM89, JS89]. In distributed computing, there are several protocol problems, such as Byzantine agreement, which have only randomized solutions [FLP85]. And in cryptography, secret keys must be chosen at random (else they are not secret), and even the cryptographic algorithms themselves must often be randomized in order to be secure [GM84].

When randomness is used in the design of algorithms and protocols, the source of randomness is modeled as an ideal process that outputs *unbiased* and *independent* random bits. On the other hand, the conceivable sources of randomness that an algorithm can effectively access (e.g., statistics on disk access time, or on keyboard typing), while containing a noticeable amount of entropy, can be very biased and involve heavy dependencies. A large body of research, initiated in [Blu86, SV86, CG88, VV85], has been devoted to fill this gap between realistic sources of randomness with biases and dependencies and perfect sources of randomness. Ideally, one would like to have a "compiler" that, given an algorithm/protocol that is guaranteed to work well only with a perfect source of randomness, produces an algorithm/protocol that is guaranteed to work well with a large class of imperfect random sources.

1.1 Simulation of Probabilistic Algorithms Using Extractors

For the case of probabilistic algorithms, one way of designing such "compilers" is to design a *randomness extrac-*

*Columbia University, Department of Computer Science, and U.C. Berkeley, Computer Science Division. luca@cs.columbia.edu.

†Institute for Advanced Study, Princeton, NJ, and Harvard University, Cambridge, MA. E-mail: salil@deas.harvard.edu. Work done while at MIT, while supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

tor, as proposed by Nisan and Zuckerman [NZ96]. A randomness extractor is a procedure that on input a sample from a weak random source and a truly random string gives an output that is statistically close to uniform. Formally, a (k, ε)-*extractor* is a procedure $\text{EXT} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ such that if X is random variable of min-entropy at least k, and U_t is the uniform distribution over $\{0,1\}^t$, then $\text{EXT}(X, U_t)$ is ε-close to uniform.[1] A large body of research has produced explicit constructions where k can be essentially arbitrary, m is very close to k, and t is $O(\log n)$ (see, e.g., [NT99, Zuc97, Tre99, RSW00] and the references therein). By definition, once we have such a (k, ε)-extractor, we can perform any task which is designed to use m truly random bits using instead a single sample from a random source of min-entropy k *together with* t *truly random bits*. Since we still need some truly random bits, this does not yet achieve the goal of using only a weak source of randomness. However, in most algorithmic applications, the need for t additional truly random bits can be eliminated by enumerating all 2^t posibilities and combining the algorithm's outputs for each, e.g. by majority vote (for decision problems). This incurs a slowdown of factor of 2^t, but fortunately this is still polynomial since we use an extractor with $t = O(\log n)$.

Note that the fact that randomness extractors can be used to run randomized algorithms with only a weak random source (and no additional truly random bits) does not mean that one can *extract* almost uniform bits from a weak random source without additional truly random bits. Indeed, for any deterministic function $\text{EXT} : \{0,1\}^n \to \{0,1\}^m$, there is a distribution X of min-entropy $n-1$ for which $\text{EXT}(X)$ is very biased (in fact, one for which the first bit of $\text{EXT}(X)$ is constant) [CG88].

1.2 Deterministic Extraction

The reason why extractors can be used for the simulation of probabilistic algorithms is essentially that when a probabilistic algorithm uses t bits of randomness it can always be simulated deterministically at the price of a 2^t slowdown factor. In other applications of randomness, such as probabilistic encryption [GM84] or Byzantine agreement [FLP85], randomness is required by the very nature of the problem, and there is no possibility of trading off efficiency for randomness. For such applications, it appears unavoidable to look for extraction procedures that convert a weak random source into an almost uniform distribution *deterministically*, without the help of extra randomness. Because of the above-mentioned impossibility results, such deterministic extractors will not work for every source of sufficiently large min-entropy. However it is still possible that there are fairly general and natural families of weak random sources for which efficient deterministic extraction is possible.

When random bits are needed in practice (e.g., to generate keys in a cryptographic protocol), a typical approach is to collect weakly random data, and feed it into a cryptographic hash function. The output of the hash function is then used as if it were a sequence of random bits. However, we know of no theoretical justification for this way of using a fixed cryptographic hash function to do deterministic extraction.

On theoretical side, there is a considerable body of work devoted to the problem of deterministic extraction. In fact, most of the early work on the use of weak random sources was devoted to the construction of deterministic extractors for increasingly general classes of distributions. A classical algorithm by von Neumann [vN51] (improved by Elias [Eli72]) extracts randomness from a sequence of *independent* coin tosses of the same biased coin. Blum [Blu86], generalizing von Neumann's result, showed how to extract randomness from any distribution described by a Markov chain. Chor and Goldreich [CG88] (improving results of Santha and Vazirani [SV86] and Vazirani [Vaz87]) show how to extract randomness given two independent weak random sources with enough min-entropy. Another line of work considered the problem of deterministically extracting randomness from various types of sources where an adversary can fix some subset of the bits, mostly motivated by applications of such extractors in cryptography and distributed computing (cf., [CGH+85, BBR88, BL90, LLS89, CDH+00]).

The extraction algorithms presented in the above papers work for classes of distributions that satisfy fairly strong *independence* properties (which is a particularly problematic assumption for physical sources of randomness). Independence requirements are explicit in most of the works, and are also implicit in [Blu86], where the process that samples the distribution has limited memory and works on-line, so that far-away parts of the output of the distribution can only have limited dependencies. In order to circumvent the impossibility of deterministic extraction for many sources of interest (in particular, ones without strong independence guarantees), researchers were led to consider the weaker task of efficiently simulating randomized algorithms with such sources [VV85, CG88, Vaz84, CW89, Zuc96], and eventually to the notion of extractors which can use a small number of additional truly random bits [NZ96].

1.3 Our Results

Our aim is to identify as general a class of sources as possible for which efficient deterministic extraction can be

[1] A distribution X has *min-entropy* k if for any element a of its range $\Pr[X = a] \leq 2^{-k}$. Two distributions X and Y are ε-*close* if for any subset S of their range $|\Pr[X \in S] - \Pr[Y \in S]| \leq \varepsilon$.

done. Specifically, we examine *samplable distributions*; that is, sources that can be generated by an efficient sampling algorithm (or circuit). The only other requirement we place on the source is that it contains some randomness to be extracted (as measured by min-entropy). In particular, we do not impose any independence conditions on the source. This class of samplable distributions contains as special cases most of the previously studied sources for which deterministic extraction was found to be possible. In addition to their generality, one can argue that samplable distributions are a reasonable model for distributions actually arising in nature. Indeed, considerable efforts were made to extend Levin's theory of average-case complexity [Lev86] to this class of samplable distributions [BCGL92, IL90].

Having settled on this class of sources, what we're looking for are functions $\textsc{Ext} : \{0,1\}^n \to \{0,1\}^m$ with the following property: for every source X of some min-entropy k which is samplable by a circuit of some size s, $\textsc{Ext}(X)$ is ε-close to uniform. Note that although we are placing a computational restriction on the sampler, we are requiring the output of the extractor to be *statistically* close to uniform. The reason for this choice is that it is possible to achieve this stronger property (as shown by our constructions). In addition, for extracting a small number of bits (as in several of our results), there is no difference between statistical and computational indistinguishability.[2]

Nonuniform Extractors and Negative Results. Our first observation is that extracting randomness from samplable distributions is impossible unless the extractor is allowed to use more computational resources than the sampler. On the other hand, if we allow the running time of the extractor to be polynomially larger than the running time (or even circuit size) of the sampler, we show that extraction becomes possible. Our first result in this vein demonstrates the existence of good deterministic extractors[3] computed by polynomial-size circuits. More precisely, for every s and n, there is an extractor of size $\mathrm{poly}(s, n)$ that extracts almost all the randomness from any distribution on $\{0,1\}^n$ samplable by a circuit of size s.[4]

A Connection to Nondeterministic Average-case Hardness. While the nonuniform extractors mentioned above illustrates the feasibility of deterministic extraction, it would be preferable to have a construction in which the extractor is efficiently computable by a uniform algorithm. However, as we show in Section 3 that the existence of such extractors implies separations of complexity classes beyond what is currently known. Therefore, in order to construct uniform deterministic extractors, we will need to make complexity assumptions.

Let us consider for starters the task of extracting one almost unbiased bit (already a nontrivial problem). Our first result is that if a Boolean function is hard to compute by $\mathbf{NP}$-circuits (i.e., circuits that can have special gates solving SAT instances) of size s with advantage better than γ, then it is also a good extractor against samplers of size about s that sample a distribution of length n of min-entropy about $n - \log(1/\gamma)$. The basic idea in the proof of this result is quite simple: suppose that f is a function hard on average for $\mathbf{NP}$-circuits, and that X is a samplable distribution on which $f(X)$ is, say, biased towards 1. Then the following $\mathbf{NP}$ circuit can predict $f(x)$: given x, first check whether x is in the range of X, which is something that can be done efficiently using nondeterminism (since X is samplable). If x is in the range, then guess that $f(x)$ is 1, otherwise make a random guess. For a random x, this approach guesses $f(x)$ with an advantage that depends on the bias of $f(X)$ and on the min-entropy of X.[5]

Although the assumption that we have a function that is hard-on-average for $\mathbf{NP}$-circuits (as opposed to standard circuits) has been used before (e.g., by Arvind and Köbler [AK97]), it is still natural to ask whether the nondeterministic hardness assumption is really necessary. In Section 4, we observe that a Boolean function can be very hard-on-average against standard circuits, yet it may not be a good extractor for samplable distributions, even for min-entropy $n-1$. So it appears that a somewhat nonstandard hardness assumption is required. Still, it is of interest to weaken the assumption, as we do next.

Using Worst-case Hardness. Our next goal is to start with a reasonable *worst-case* complexity assumption, such as the one used by Klivans and van Melkebeek [KvM99]: that $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ contains a problem that is not solvable by $\mathbf{NP}$-circuits of size $2^{o(n)}$). We would like to show that such an assumption implies the existence of polynomial-time computable predicates with strong average-case hardness against $\mathbf{NP}$-circuits; by the previous results, such predicates would be good deterministic extractors. This looks like the standard problem of worst-case to average-case reduction, as solved in

[2] It should be noted that once one can extract a small number of bits that are statistically close to uniform, the pseudorandom generator constructions of [NW94, IW98] can transform these into many bits that are computationally indistinguishable from uniform (under complexity assumptions that are weaker than the ones we make).

[3] Here, and from this point on, the term *deterministic extractor* always refers to a deterministic extractor for samplable distributions.

[4] Dodis, Sahai, and Smith [DSS00] have used our proof to show that there are small circuits which compute good "adaptive statistical exposure-resilient functions," by observing that these are a special case of extractors for samplable distributions.

[5] This explanation is a bit oversimplified: our idea works as described only if X is a samplable "flat" distribution. For non-flat distributions, a more sophisticated reduction is needed, which involves the use of approximate counting algorithms with an $\mathbf{NP}$-oracle [Sip83, Sto85, JVV86].

[BFNW93, Imp95, IW97, STV99], and observed to extend to $\mathbf{NP}$-circuits in [KvM99]. However, in all such results, one gets predicates that are hard to predict with an advantage that is at least inversely proportional to the size of the adversary (and, for a stronger reason, on the time needed to compute the predicate). It then follows that an extractor computable in time $t(n)$ obtained using such techniques and the previously mentioned connection can only extract randomness from a source of min-entropy about $n - \log t(n)$.

In order to extract from sources of lower entropy, we exploit our ability to use nondeterminism in the reduction, in the spirit of the results of Feige and Lund [FL96] about the average-case complexity of the permanent. Our starting point is the worst-case to average-case reduction in [STV99]. That reduction uses an error-correcting code obtained by "concatenating" a multivariate polynomial code and a Hadamard code, and is analysed by providing a "list-decoding" procedure for the polynomial code and using the Goldreich–Levin [GL89] list-decoding procedure for the Hadamard code. We show that the use of "approximate counting" (implementable with an $\mathbf{NP}$ oracle [Sto85, Sip83, JVV86]) can greatly improve the efficiency of the list-decoding algorithm for the polynomial code. But we do not know whether a similar improvement is possible for the Hadamard code. Instead, we show how to use approximate counting and uniform sampling (also using an $\mathbf{NP}$ oracle [JVV86, BGP98]) to get a very efficient solution to a somewhat different problem that still suffices for deterministic extraction.

The final result is that starting from a problem in $\mathbf{E}$ that does not admit circuits of size $2^{o(n)}$ with $\mathbf{\Sigma_5}$-gates, we get an efficient extractor that extracts one almost unbiased bit from any samplable distribution of length n and min-entropy $(1 - \gamma)n$, for some constant $\gamma > 0$. In order to extract from distributions samplable by circuits of size s, the extractor needs running time $\mathrm{poly}(s)$. The somewhat unusual assumption can be thought of as a "scaling" of $\mathbf{EXP} \not\subseteq \mathbf{\Sigma_5}/\mathrm{poly}$, which is true as long as the Polynomial Hierachy does not collapse.

Extracting Many Bits. So far, we described results giving extractors that only produce one almost unbiased bit, while it is of course much preferable to extract a number of random bits that be as close as possible to the entropy of the source. We first show that our coding-theoretic methods can be used to extract approximately a logarithmic number of random bits. To this end, we use the same polynomial code as before, but in place of the Hadamard code, we use a similar code on a bigger alphabet. Once we have these logarithmic number of random bits, we can use them as the truly random bits for the extractor of Zuckerman [Zuc97], which we then use to extract almost all the entropy from our source. Formally, we prove that if there is a problem in $\mathbf{E}$ that does not admit circuits of size $2^{o(n)}$ with $\mathbf{\Sigma_6}$ gates, there is an efficient deterministic extractor that extracts $(1 - O(\gamma))n$ bits from any samplable distribution of min-entropy $(1-\gamma)n$, where γ is any sufficiently small constant. Again, the running time of the extractor is polynomial in the circuit complexity of the samplers.

1.4 Perspective

Our main motivation for studying samplable distributions is their generality. However, this generality has a price; the extractor must use more computational resources than the sampler, and has to rely on complexity assumptions. Given the current state-of-the-art in complexity theory, it seems unavoidable that even under strong assumptions, to get an extractor for distributions of length n sampled by circuits of size, say, $O(n \log n)$ one has to come up with a fairly complex and impractical solution. On the other hand, we think that it is interesting to explore the limits of the possibility of deterministic extraction, and it seems that samplable distributions are a good and natural borderline example.

Seemingly, our definition is orthogonal to the one used by Chor and Goldreich [CG88] for two independent weak random sources. In the Chor–Goldreich setting, distributions can be arbitrarily complex, but they satisfy a strong independence requirement. In our case, distributions have to be samplable but can involve arbitrary dependencies. However, there is a connection. In this paper, we give "computational" constructions, using a hard predicate to build our deterministic extractors; when the result is not a deterministic extractor, a reduction shows that the predicate is not hard. As shown in [Tre99], such computational constructions can have interesting and unexpected information-theoretic interpretations, and it is natural to look for the information-theoretic interpretation of the results of this paper. As it turns out, the information-theoretic analogue of deterministic extractors for samplable distributions is exactly the problem of extracting randomness from two independent weak random sources! Briefly, if we have two independent weak random sources X_1 and X_2, then X_2 has a large description size (i.e., Kolmogorov complexity) even conditioned on $X_1 = x_1$ for any x_1. Thus, similar to [Tre99], we can view X_2 as the truth table of a hard predicate relative to X_1, which can be used to deterministically extract randomness from X_1. Such an interpretation of our results gives (unconditional) constructions of deterministic extractors for two independent weak random sources, for the case where the two sources have different lengths, and the longer one has a very low entropy rate. The details of these corollaries will be given in the full version of the paper (including additional optimizations that be done in the information-theoretic setting).

Part of the purpose of this paper is to point out the need for a further development of the theory of deterministic extractors, and to invite the reader to come up with alternative definitions and constructions. We believe that it would be very good to come up with a natural and general class of distributions that admit an efficient (implementable!) deterministic extractor. Such a deterministic extractor could then be used in place of cryptographic hash functions in order to extract randomness in practice, with the advantage of having a sound motivation for its use.

One natural direction for such research would be to seek deterministic extractors for distributions which have space-bounded samplers, rather than time-bounded ones as we have. As with pseudorandom generators (cf., [Sak96]), there is hope for *unconditional* results in the space-bounded setting. The samplers considered by Blum [Blu86], namely finite-state Markov chains, can be viewed as a limited form of space-bounded of computation, but the extractors given there only work when the number of bits received from the source is much greater than the number of states in the Markov chain. A much richer class of sources would be obtained by looking at distributions on $\{0,1\}^n$ sampled by an $O(\log n)$-space machine.

2 Preliminaries

Probability Distributions. Let X and Y be probability distributions on a discrete universe $\mathcal{U}$. X is said have *min-entropy* k if for all $x \in \mathcal{U}$, $\Pr[X = x] \leq 2^{-k}$. It will also be convenient for us to have the following equivalent terminology. X has *density* δ in $\mathcal{U}$ if $\max_{x\in\mathcal{U}} \Pr[X = x] = 1/(\delta \cdot \mathcal{U})$. Note that if X is uniform over a subset S of $\mathcal{U}$, then δ is the density of S in $\mathcal{U}$ (hence the terminology). Note that a distribution has density at least δ in $\{0,1\}^n$ iff it has min-entropy $n - \log(1/\delta)$.

The *statistical difference* between X and Y is defined to be

$$\mathrm{SD}(X,Y) \stackrel{\text{def}}{=} \max_{S\subset\mathcal{U}} |\Pr[X \in S] - \Pr[Y \in S]|$$

$$= \frac{1}{2} \cdot \sum_{x\in\mathcal{U}} |\Pr[X = x] - \Pr[Y = x]| \,.$$

If $\mathrm{SD}(X,Y) \leq \varepsilon$, we say that X and Y are ε*-close*. U_m denotes the uniform distribution on $\{0,1\}^m$. If X is a distribution on $\{0,1\}$, then we call $\mathrm{SD}(X,U_1)$ the *bias* of X.

We will consider probability distributions given by sampling algorithms. If A is a probabilistic algorithm (Turing machine), we write $A(x;y)$ for the output of A on input x and random coins y. $A(x)$ denotes the output distribution of A on input x when the coins y are chosen uniformly at random. A *probabilistic circuit* is a Boolean circuit $C : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^n$. For $x \in \{0,1\}^m$, we write $C(x)$ for the distribution on $\{0,1\}^n$ obtained by selecting y uniformly in $\{0,1\}^r$ and evaluating $C(x;y)$.

We say that a probability distribution is *samplable by size* s if there is a circuit of size s which samples from it. An ensemble $\{X_n\}$ of probability distributions is *uniformly samplable in time* $t(n)$ if there is a probabilistic algorithm A such that $A(1^n) = X_n$ for every n and the running time of A on input 1^n is at most $t(n)$.

Extractors. A function $\textsc{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ε)*-extractor* if for every distribution X on $\{0,1\}^n$ of min-entropy k, $\textsc{Ext}(X,U_d)$ is ε-close to U_m.[6] As shown by Nisan and Zuckerman [NZ96] it is necessary to invest $d \geq \Omega(\log(n-k) + \log 1/\varepsilon)$ truly random bits for any nontrivial extraction (i.e., when $m \leq d-1$ and $k \leq n-1$).[7] In order to make extraction possible without investing any truly random bits, we restrict to samplable distributions:

Definition 2.1 *A function* $\textsc{Ext} : \{0,1\}^n \to \{0,1\}^m$ *is an* (k,ε)-deterministic extractor against circuit-size s *if for every distribution* X *on* $\{0,1\}^n$ *which has min-entropy* k *and is samplable by size* s*,* $\textsc{Ext}(X)$ *is* ε*-close to* U_m*.*

Definition 2.2 *A family of functions* $\{\textsc{Ext}_n : \{0,1\}^n \to \{0,1\}^{m(n)}\}$ *is a* $(k(n),\varepsilon(n))$-deterministic extractor against time $t(n)$ *if for every ensemble of distributions* $\mathcal{X} = \{X_n\}$ *such that* $\mathcal{X}$ *is uniformly samplable in time* $t(n)$ *and* X_n *is a distribution on* $\{0,1\}^n$ *of min-entropy* $k(n)$*, we have* $\textsc{Ext}(X_n)$ *is* $\varepsilon(n)$*-close to* $U_{m(n)}$ *for all sufficiently large* n*.*

Nondeterministic Circuits. We denote the levels of the polynomial-time hierarchy as follows: $\mathbf{\Sigma_0} = \mathbf{P}$, $\mathbf{\Sigma_{i+1}} = \mathbf{NP}^{\mathbf{\Sigma_i}}$. A Σ_i*-algorithm* is an algorithm with an oracle for Σi. Similarly, a Σ_i*-circuit* is a Boolean circuit which can have gates for some fixed $\mathbf{\Sigma_i}$-complete problem (e.g., QBF_{i-1}) in addition to the usual $\wedge$, $\vee$, and $\neg$ gates. By replacing "algorithm" or "circuit" with "Σ_i-algorithm" or "Σ_i-circuit" in the definitions above, we can also define *probabilistic* Σ_i*-algorithms*, *probabilistic* Σ_i*-circuits*, distributions *samplable by* Σ_i*-circuits of size* s, (k,ε)*-deterministic extractors against* Σ_i*-circuits of size* s, etc.

Definition 2.3 *A function* $f : \{0,1\}^n \to \{0,1\}$ *is* (s,ε)-hard for Σ_i-circuits *if for every* Σ_i*-circuit* C *of size at most* s*, we have*

$$\Pr[f(x) = C(x)] \leq 1/2 + \varepsilon/2$$

[6] This definition of extractor, taken from [NT99], is weaker than the original definition proposed in [NZ96] (which requires that the d-bit seed be explicitly included in the output). But this definition suffices for most applications of extractors.

[7] Better (and tight) bounds on d can be found in [RT97].

3 Nonuniform Extractors & Negative Results

Proposition 3.1 *For every s, n, $k \le n$, and ε, there exists an (k,ε)-extractor* $\text{EXT} : \{0,1\}^n \to \{0,1\}^m$ *against circuit-size s, with $m = k - 2\log(1/\varepsilon) - O(\log s)$. Moreover,* EXT *can be computed by a circuit of size* $\text{poly}(s,n)$.

The proof, that will appear in the full version of this paper, is based on a version of the Leftover Hash Lemma for t-wise independent functions that shows that there is an $\exp(-\Omega(t))$ probability that a function picked from the family is not a good extractor for a fixed weak random source. (The standard Leftover Hash Lemma for pairwise independent families [ILL89] does not give a high enough success probability for our purposes.) A union bound then shows that with high probability over the choice of the function from the family, it is simultaneously a good deterministic extractors for all weak random sources having small samplers.

Note that in Proposition 3.1, the extractor has a higher circuit complexity than the samplers from which it extracts. This is necessary, even if we only want to extract one bit from a distribution of min-entropy $n-1$:

Proposition 3.2 *There is a constant c such that no function* $\text{EXT} : \{0,1\}^n \to \{0,1\}$ *computable by a circuit of size s is a $(n-1, 1/5)$-deterministic extractor against circuit size $c \cdot s$.*[8]

Proof: Without loss of generality, we may assume that $\text{EXT}(x) = 1$ for at least half of its inputs. Consider the distribution X sampled by the following algorithm:

1. Select x uniformly in $\{0,1\}^n$.
2. If $\text{EXT}(x) = 1$, output x. Otherwise, output a uniformly selected $x' \in \{0,1\}^n$.

It is easy to see that X has min-entropy $n-1$ and is samplable by size $O(s)$. Moreover, $\text{EXT}(X) = 1$ with probability at least $3/4$. ■

In subsequent sections, we aim to construct deterministic extractors that are efficiently computable by *uniform* algorithms. The following two corollaries show that such extractors imply separations between deterministic complexity classes and nonuniform or probabilistic ones. Since such separations are beyond the current state-of-the-art in complexity theory, our constructions should (and will) be based on complexity-theoretic assumptions.

Corollary 3.3 *Suppose* $\{\text{EXT}_n : \{0,1\}^n \to \{0,1\}\}$ *is a family of functions computable in time $t(n)$ such that, for every n,* EXT_n *is an $(n-1, 1/5)$-deterministic extractor against circuit-size $s(n)$. Then there is a language in* $\mathbf{DTIME}(t(n))$ *of circuit complexity at least* $\Omega(s(n))$.

Proof: Let $L = \{x \in \{0,1\}^* : \text{EXT}_{|x|}(x) = 1\}$. Proposition 3.2 implies that this language has circuit complexity at least $s(n)/c$. ■

A similar argument holds in the uniform setting:

Corollary 3.4 *Suppose* $\{\text{EXT}_n : \{0,1\}^n \to \{0,1\}\}$ *is family of functions computable in time $t(n)$ and is an $(n-1, 1/5)$-deterministic extractor against time $t'(n)$. Then there is a language in* $\mathbf{DTIME}(t(n)) \setminus \mathbf{BPTIME}(\Omega(t'(n)))$.

4 Extractors from Average-Case Hardness

The following lemma relates the average-case complexity of a Boolean function to its extraction property.

Lemma 4.1 *Let $f : \{0,1\}^n \to \{0,1\}$ be (s,ε)-hard for Σ_1-circuits. Let X be a flat distribution on $\{0,1\}^n$ of min-entropy $n - \Delta$ samplable by a circuit of size $s - O(n)$. Then $f(X)$ is $2^{\Delta} \cdot \varepsilon$-close to uniform.*

The (omitted) proof of Lemma 4.1 follows the intuition outlined in the introduction: If there were a samplable distribution X of high min-entropy on which f were biased towards 1, then a Σ_1-circuit could obtain an advantage in computing f on a random input x by testing whether x were in the support of X.

In the standard information-theoretic setting, if a function extracts randomness out of every flat distribution of min-entropy k, then it follows that it also extracts randomness out of any (not necessarily flat) distribution of min-entropy k (cf., [CG88]). This is due to the fact that any distribution of min-entropy k is a convex combination of flat distributions of min-entropy k. In our framework, it is no more true (or at least no longer clear) that any *samplable* distribution of min-entropy k is a convex combination of flat *samplable* distributions of min-entropy k. So we need an additional technical step in order to remove the flatness requirement.

Before continuing, let us pause for a moment to consider the nondeterministic complexity assumption that we made in the above lemma, and discuss its strength. As seen in the previous section, it is necessary to make a complexity assumption in order to construct uniform deterministic extractors. However, it is not natural that the assumption should be about nondeterministic hardness, and it would be more appealing to have a construction based on standard average-case hardness. Even though we do not know whether nondeterministic hardness assumptions are *necessary* to construct deterministic extractors, we can argue that

[8] The constant of $1/5$ can be replaced by any constant less than 1, at the price of increasing c.

standard hardness is not sufficient. Let π be a one-way permutation, and let B be a hard-core predicate for π: then $f(x) = B(\pi^{-1}(x))$ is a hard-on-average function; however, it is not an extractor because it is easy to sample from the conditional distribution of x such that $f(x) = 0$ (and this distribution has min-entropy $n - 1$). We can conclude that, if one-way permutations exist, it's not possible to prove that every hard-on-average predicate is a deterministic extractor against small samplers.

Now we proceed to relate nondeterministic hardness to deterministic extraction for samplable distributions that are not necessarily flat. Now, when trying to compute $f(x)$, it will no longer suffice to test whether an input x is merely in the range of a distribution X on which f is biased. Instead, we will guess the value $f(x)$ randomly, with a bias that depends on (an approximation to) the probability mass of x under X. This can be accomplished by a Σ_1-circuit because approximate counting can be can be performed with an $\mathbf{NP}$ oracle [Sip83, Sto85, JVV86].

Lemma 4.2 *Let $f : \{0,1\}^n \to \{0,1\}$ be (s,ε)-hard for Σ_1-circuits. Then, for every $\Delta \leq n$, f is a $(n-\Delta, 2^{\Delta+1} \cdot \varepsilon)$ extractor against circuit-size $(2^{\Delta}\varepsilon s)^{\Omega(1)}$.*

We remark that Lemma 4.2 generalizes to give deterministic extractors which extract several bits from nonboolean functions which are sufficiently hard-on-average for Σ_1-circuits.

5 Extractors from Worst-Case Hardness

In the previous section, we saw that the property of a function being a deterministic extractor is in some sense a generalization of a function being hard to compute on average. In this section, we show how to construct deterministic extractors from functions that are hard to compute in the *worst case*. To do this, we follow the usual paradigm for transforming a worst-case hard function f to an average-case hard function $\hat{f}$: we take $\hat{f}$ to be an encoding of f in an appropriate error-correcting code [BFNW93, STV99]. To prove the correctness of such a construction, one typically argues that given any small circuit C which computes $\hat{f}$ on average, i.e. has some advantage δ over "random guessing", one can can use a decoding algorithm for the error-correcting code to build another small circuit C' which computes f everywhere, contradicting the worst-case hardness of f. However, existing results of this form will not yield the results we desire. The reason is that these decoding procedures typically produce a C' of size polynomial in $1/\delta$, whereas we are interested in values of $1/\delta$ that are much larger than the hardness of f. (If we are extracting from a source of min-entropy k, δ will be comparable to $1/2^{n-k}$, whereas the circuit complexity of f will be at most the running time of the extractor, which we would like to be $\mathrm{poly}(n)$.)

In the spirit of the results of Feige and Lund [FL96] about the average-case complexity of the permanent, we overcome this difficulty by exploiting *nondeterminism* in our reduction. Specifically, by augmenting the polynomial reconstruction algorithm given in [STV99] with nondeterminism, we obtain the following result:

Lemma 5.1 *Let $\mathbb{F}$ be a finite field (with some fixed, efficient representation), and let $p : \mathbb{F}^t \to \mathbb{F}$ be a polynomial of total degree at most d. If there is a Σ_i-circuit C which computes p correctly on at least a $\delta = c\sqrt{d/|\mathbb{F}|}$ fraction of points (where c is a universal constant), then there is a Σ_{i+2}-circuit C' of size $\mathrm{poly}(|C|, d)$ which computes p correctly everywhere.*[9]

Proof sketch: It is shown in [STV99] that there exists a point $z \in \mathbb{F}^m$ such that for at least a 15/16 fraction of lines ℓ through z,

1. $p|_\ell$ and $C|_\ell$ agree on at least a $\delta/2$ fraction of points on ℓ.

2. There does not exist any degree d polynomial $h : \ell \to \mathbb{F}$ other than $p|_\ell$ which agrees with $C|_\ell$ in at least a $\delta/4$ fraction of points on ℓ and $h(z) = p(z)$.

Fix such a z, and consider the following procedure, which attempts to compute p at $x \in \mathbb{F}^m$:

1. Let ℓ be the line through x and z.

2. Nondeterministically guess a polynomial $h : \ell \to \mathbb{F}$.

3. Verify that h agrees with C on approximately a $\delta/2$ fraction of points on ℓ and $h(z) = p(z)$. If so, output $h(x)$. Otherwise, reject h.

This procedure can be implemented efficiently using nonuniformity (to hardwire z, $p(z)$, and C) and two levels of nondeterminism (one to guess h and one to perform approximate counting [Sip83, Sto85, JVV86]). Thus, we obtain a Σ_{i+2}-circuit computing p in at least a $15/16$ fraction of points, which can be converted into one which computes p everywhere via the "self-corrector" of [GLR+91]. $\square$

This lemma implies that if we start with a function f which is worst-case hard for Σ_3-circuits and encode it as a low-degree polynomial, we obtain a function $\hat{f}$ which is very hard on average for Σ_1-circuits, as desired. However, there is still a problem. While $\delta = c\sqrt{d/|\mathbb{F}|}$ is very small, it is still a substantial *relative* advantage over random guessing, which would give success probability $1/|\mathbb{F}|$.

[9] The size of C' does not explicitly refer to $\log|F|$ and t because the size of C is at least the length of its input, which is $t \log |F|$.

The usual method for getting around this difficulty, is to "concatenate" the polynomial encoding with an "inner" encoding whose output lies in a much smaller alphabet (e.g., $\{0,1\}$). By combining the decoding procedure for the polynomial encoding with an analogous one for the inner code, one proves that no small circuit can compute the new function in a $1/2+\delta'$ fraction of points. Unfortunately, we know of no such inner code where we do not incur the $\mathrm{poly}(1/\delta')$ blow-up in decoding that we hoped to avoid, even if we use nondeterminism.

To solve this problem, we exploit the fact that what we need for deterministic extraction is weaker than standard average-case hardness, and it turns out that the most commonly used inner code has the properties we need. For $w \in \{0,1\}^n$, the *Hadamard encoding* of w is the function $\mathrm{Had}_w : \{0,1\}^n \to \{0,1\}$ obtained by setting $\mathrm{Had}_w(x)$ to be the mod-2 inner product of w and x. The following lemma lists the only property of this code that we will use (aside from the fact that, given x and w, $\mathrm{Had}_w(x)$ can be computed in time $\mathrm{poly}(n)$).

Lemma 5.2 *Let X be any distribution on $\{0,1\}^n$ of density δ and let $\varepsilon > 0$. Then*

$$\#\{w : \mathrm{Had}_w(X) \textit{ has bias at least } \varepsilon\} \leq \frac{1}{\delta \cdot \varepsilon^2}.$$

Forms of Lemma 5.2 have been proven by Lindsey, Alon [Alo86], and Chor and Goldreich [CG88]. In the full version of our paper, we give a direct proof.

Although Lemma 5.2 does not explicitly give an efficient decoding algorithm, we can easily obtain one using nondeterminism:

Lemma 5.3 *For every fixed i, there is a probabilistic Σ_{i+2}-algorithm $\texttt{HadDecode}_i$ with the following property: Let C be a probabilistic Σ_i-circuit which samples a distribution X on $\{0,1\}^n$ of density δ and let $w \in \{0,1\}^n$ be such that $\mathrm{Had}_w(X)$ has bias at least ε. Then $\texttt{HadDecode}_i(C,\varepsilon)$ runs in time $\mathrm{poly}(|C|, 1/\varepsilon)$ and outputs w with probability $\Omega(\delta \cdot \varepsilon^2)$.*

The key point is that although the success probability of the decoding procedure depends on δ, the running time does not.

Proof sketch: With one level of nondeterminism, approximate counting [Sip83, Sto85, JVV86] can be used to distinguish those v such that $\mathrm{Had}_v(X)$ has bias at least ε from those such that $\mathrm{Had}_v(X)$ has bias at most $\varepsilon/2$. With one more level of nondeterminism, we can use [JVV86, BGP98] to uniformly sample from the set of v that pass this test. $\square$

To obtain deterministic extractors, we combine the polynomial encoding and Hadamard code via the standard "concatenation" technique. Let $\mathbb{F} = \mathrm{GF}(2^q)$,[10] and for a function $p : \mathbb{F}^t \to \mathbb{F}$, define the *Hadamard encoding* of p to be the function $p' : \mathbb{F}^t \times \{0,1\}^q \to \{0,1\}$ defined by $p'(x,y) = \mathrm{Had}_{p(x)}(y)$, where we view $p(x) \in \mathbb{F}$ as a an element of $\{0,1\}^q$.

By combining the decoding procedures of Lemmas 5.1 and 5.3, we obtain the following procedure for decoding the concatenated code.

Theorem 5.4 *Let $\mathbb{F} = \mathrm{GF}(2^q)$, let $p : \mathbb{F}^t \to \mathbb{F}$ be a polynomial of degree at most d, and let $p' : \mathbb{F}^t \times \{0,1\}^q \to \{0,1\}$ be its Hadamard encoding. Suppose there is a distribution X on $\mathbb{F}^t \times \{0,1\}^q$ which is of density δ and is samplable by size s such that $p'(X)$ has bias ε. Then there is a Σ_5-circuit*[11] *of size $\mathrm{poly}(s, d, 1/\varepsilon)$ which computes p' everywhere, provided that*

$$\delta^2 \cdot \varepsilon^4 \geq c\sqrt{\frac{d}{|\mathbb{F}|}},$$

where c is a universal constant.

Proof sketch: For every $x \in \mathbb{F}^t$, let X^x denote the conditional distribution on $\{0,1\}^q$ induced by conditioning the first component of X to be x. From the fact that $p'(X)$ has bias ε, it follows that for at least an $\Omega(\delta\varepsilon)$ fraction of $x \in \mathbb{F}^t$, X^x has density $\Omega(\delta\varepsilon)$ in $\{0,1\}^q$ and $p'(x, X^x)$ has bias $\Omega(\varepsilon)$. Each X^x is samplable by a Σ_1-circuit, so using Lemma 5.3 we obtain a Σ_3-circuit computing p on at least a $\Omega(\delta\varepsilon) \cdot \Omega(\delta\varepsilon) \cdot \varepsilon^2$ fraction of points. The theorem now follows from Lemma 5.1. $\square$

This immediately gives us a construction of deterministic extractors from Boolean functions that are worst-case hard for Σ_5-circuits.

Theorem 5.5 *If there is a problem in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ which has Σ_5-circuit complexity $2^{\Omega(n)}$ for all n, then there is a constant $\gamma > 0$ such that for all n and s satisfying $n \leq s \leq 2^{\gamma n}$, there is a $((1-\gamma)n, 1/s)$-deterministic extractor $\mathrm{Ext}_{n,s} : \{0,1\}^n \to \{0,1\}$ against circuit-size s such that $\mathrm{Ext}_{n,s}$ is computable in time $\mathrm{poly}(s)$.*

Proof sketch: Let $f : \{0,1\}^\ell \to \{0,1\}$ be a function computable in time $2^{O(\ell)}$ with Σ_5-circuit complexity $2^{\Omega(\ell)}$. Our extractor will be the Hadamard encoding of an appropriate polynomial $p : \mathbb{F}^t \to \mathbb{F}$ extending f. We can set the parameters so that d, s, and $1/\varepsilon$ are each $2^{\Omega(\ell)}$, $|\mathbb{F}|$ is anywhere

[10] The restriction to fields of characteristic 2 is inessential and only done to make passing between field elements and strings over $\{0,1\}$ cleaner.

[11] By "sharing" some of the nondeterminism at different levels of the reduction, the number of levels of nondeterminism introduced can be reduced a bit. For the sake of modularity in the exposition, we have chosen not to optimize this parameter.

between d^2 and $\exp(2^{\Omega(\ell)})$, $\delta = 1/\mathbb{F}^{\Omega(1)}$, and $t = O(1)$, while $\delta^2 \cdot \varepsilon^4 \geq c\sqrt{d/|\mathbb{F}|}$ and the conclusion of Theorem 5.4 produces a circuit of size smaller than the circuit complexity of f. This implies that p' extracts on bit from any distribution on $\mathbb{F}^t$ of density δ (which is samplable by size s). Now note that the input length of p' is $n = t \log |F|$, and it extracts one bit from from samplable distributions of min-entropy $n - \log(1/\delta) = (1 - \Omega(1))n$. Furthermore, the extractor can be computed in time $2^{O(\ell)} = \mathrm{poly}(s)$. □

6 Extracting Many Bits

We begin by describing the replacement for the Hadamard code which will enable us to extract a logarithmic number of bits. The construction we use is the "hard-core function" of Goldreich and Levin [GL89]. For $y \in \{0,1\}^{n+m}$, define $\mathrm{HCF}_y : \{0,1\}^n \to \{0,1\}^m$ by $\mathrm{HCF}_y(x) = \mathrm{HCF}_y^1(x)\mathrm{HCF}_y^2(x) \cdots \mathrm{HCF}_y^m(x)$, where $\mathrm{HCF}_y^i(x)$ is the mod-2 inner product of x and $y_i y_{i+1} \cdots y_{i+n-1}$.

Analogous to Lemma 5.2, the only property of HCF is the following.

Lemma 6.1 *Let X be a distribution over $\{0,1\}^n$ of density δ. Then the number of strings $y \in \{0,1\}^{n+m}$ such that $\exists\, a \in \{0,1\}^m \Pr[\mathbf{C}(X,y) = a] > 2^{-m} + \varepsilon$. is at most $2^{2m}/\delta\varepsilon^2$.*

The (omitted) proof of this lemma proceeds via a reduction to Lemma 5.2, using Vazirani's XOR Lemma [Vaz84] and the linearity of the inner-product operation.

Following the same line of reasoning as in Section 4, we obtain the following extractors which extract logarithmically many bits:

Theorem 6.2 *If there is a problem in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ which has Σ_5-circuit complexity $2^{\Omega(n)}$ for all n, then there is a constant $\gamma > 0$ such that for all n and s satisfying $n \leq s \leq 2^{\gamma n}$, there is a $((1-\gamma)n, 1/s)$-deterministic extractor $\mathrm{EXT}_{n,s} : \{0,1\}^n \to \{0,1\}^{\log s}$ against circuit size s such that $\mathrm{EXT}_{n,s}$ is computable in time* $\mathrm{poly}(s)$.

Now, to extract more than a logarithmic number of bits, we use a simple observation about high min-entropy sources from [GW97]: If we partition a high min-entropy source into a prefix and suffix, then these two part each contain a lot of "independent randomness". More precisely, if $X = (X_1, X_2)$ is of length $n = n_1 + n_2$ (where n_i is the length of X_i) and has min-entropy $n - \Delta$, then X_1 has min-entropy $n_1 - \Delta$, and even conditioned on (most values of) X_1, X_2 has min-entropy $n_2 - \Delta$. Thus, if X is samplable, we can use the extractor of Theorem 6.2 to deterministically extract logarithmically many bits from X_2 that are (almost) uniform and independent of X_1. These bits can then be used as a seed for a standard extractor, such as the one of Zuckerman [Zuc97], to extract lots of randomness from X_1. (The extractors in [Zuc97] are very good for sources whose min-entropy is at least a constant fraction of their length; they use a logarithmic-length seed and extract a large constant fraction of the randomness from the source.) One small subtlety in this argument is that we need the conditional distribution of X_2 given X_1 to be samplable, which does not follow from the samplability of X. This conditional distribution is, however, samplable with an $\mathbf{NP}$ oracle, so we just have to move everything one level higher in the hierarchy.

Putting these ideas together, we obtain:

Theorem 6.3 *If there is a problem in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ which has Σ_6-circuit complexity $2^{\Omega(n)}$ for all n, then for every sufficiently small constant γ and for every s, there is a $((1-\gamma)n, 1/n)$-extractor $\mathrm{EXT}_{n,s,\gamma} : \{0,1\}^n \to \{0,1\}^m$ against circuit size s where $m = (1 - O(\gamma))n$. EXT is computable in time* $\mathrm{poly}(s)$, *where the exponent of the polynomial depends on γ.*

Acknowledgments We thank Avi Wigderson and Oded Goldreich for helpful discussions; Yevgeniy Dodis for pointers to previous work; and Oded Goldreich, David Zuckerman, and the anonymous referees for comments on the presentation.

References

[Alo86] N. Alon. Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory. *Combinatorica*, 6(3):207–219, 1986.

[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.

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

[BGP98] Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of NP-witnesses using an NP-oracle. Technical Report TR98-032, Electronic Colloquium on Computational Complexity, June 1998. To appear in *Information and Computation*.

[BCGL92] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average-case complexity. *J. of Computer and System Sciences*, 44(2):193–219, 1992.

[BL90] Michael Ben-Or and Nathan Linial. Collective coin-flipping. In Silvio Micali, editor, *Randomness and*

Computation, pages 91–115. Academic Press, New York, 1990.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. on Computing*, 17(2):210–229, April 1988.

[Blu86] M. Blum. Independent unbiased coin flips from a correlated biased source—a finite state Markov chain. *Combinatorica*, 6(2):97–108, 1986. Theory of computing (Singer Island, Fla., 1984).

[CDH+00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In Bart Preneel, editor, *Advances in Cryptology—EUROCRYPT 00*, Lecture Notes in Computer Science. Springer-Verlag, 14–18 May 2000.

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

[CGH+85] Benny Chor, Oded Goldreich, Johan Hastad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions (preliminary version). In *26th Annual Symposium on Foundations of Computer Science*, pages 396–407, Portland, Oregon, 21–23 October 1985. IEEE.

[CW89] Aviad Cohen and Avi Wigderson. Dispersers, deterministic amplification, and weak random sources (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 14–19, Research Triangle Park, North Carolina, 30 October–1 November 1989. IEEE.

[DSS00] Yevgeniy Dodis, Amit Sahai, and Adam Smith. Optimal lower bound for perfect all-or-nothing transforms. In preparation, April 2000.

[Eli72] P. Elias. The efficient construction of an umbiased random sequence. *Annals of Math. Stat.*, 42(3):865–870, 1972.

[FL96] Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Computational Complexity*, 6(2):101–132, 1996.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *J. Assoc. Comput. Mach.*, 32(2):374–382, 1985.

[GLR+91] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 32–42, New Orleans, Louisiana, 6–8 May 1991.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15–17 May 1989.

[GW97] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[IL90] R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st IEE Simposium on Foundations of Computer Science*, pages 812–821, 1990.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science*, pages 538–545, Milwaukee, Wisconsin, 23–25 October 1995. IEEE.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, 15–17 May 1989.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, El Paso, Texas, 4–6 May 1997.

[IW98] Russell Impagliazzo and Avi Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *36th Annual Symposium on Foundations of Computer Science*, Palo Alto, CA, November 8–11 1998. IEEE.

[JVV86] Marc R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43(2–3):169–188, 1986.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[KLM89] Richard M. Karp, Michael Luby, and Neal Madras. Monte Carlo approximation algorithms for enumeration problems. *J. Algorithms*, 10(3):429–448, 1989.

[KvM99] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Proceedings of 31st ACM Symposium on Theory of Computing*, pages 659–667, 1999.

[Lev86] Leonid A. Levin. Average case complete problems. *SIAM J. on Computing*, 15(1):285–286, February 1986.

[LLS89] D. Lichtenstein, N. Linial, and M. Saks. Some extremal problems arising from discrete control processes. *Combinatorica*, 9(3):269–287, 1989.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, December 1976.

[NT99] Noam Nisan and Amnon Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, February 1999.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[Rab80] Michael O. Rabin. Probabilistic algorithm for testing primality. *J. Number Theory*, 12(1):128–138, 1980.

[RT97] Jaikumar Radhakrishnan and Amnon Ta-Shma. Tight bounds for depth-two superconcentrators. In *38th Annual Symposium on Foundations of Computer Science*, pages 585–594, Miami Beach, Florida, 20–22 October 1997. IEEE.

[RSW00] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. In *These proceedings*, 2000.

[Sak96] Michael Saks. Randomization and derandomization in space-bounded computation. In *Proceedings, Eleventh Annual IEEE Conference on Computational Complexity*, pages 128–149, Philadelphia, Pennsylvania, 24–27 May 1996. IEEE Computer Society Press.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, August 1986.

[Sip83] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 April 1983.

[SS77] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM J. Comput.*, 6(1):84–85, 1977.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM J. on Computing*, 14(4):849–861, November 1985.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma [extended abstract]. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 537–546, Atlanta, Georgia, 1–4 May 1999.

[Tre99] Luca Trevisan. Constructions of near-optimal extractors using pseudo-random generators. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 141–148, Atlanta, Georgia, 1–4 May 1999.

[Vaz84] Umesh V. Vazirani. *Randomness, Adversaries, and Computation*. PhD thesis, University of California, Berkeley, 1984.

[Vaz87] Umesh V. Vazirani. Strong communication complexity or generating quasirandom sequences from two communicating semirandom sources. *Combinatorica*, 7(4):375–392, 1987.

[VV85] Umesh V. Vazirani and Vijay V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *26th Annual Symposium on Foundations of Computer Science*, pages 417–428, Portland, Oregon, 21–23 October 1985. IEEE.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Mathematics Series*, 12:36–38, 1951.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, October/November 1996.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.