Extracting Randomness from Samplable Distributions

Luca Trevisan *　　　Salil Vadhan†

April 28, 2000

Abstract

Randomness extractors convert weak sources of randomness into an almost uniform distribution; the conversion uses a small amount of pure randomness. In algorithmic applications, the use of extra randomness can be simulated by complete enumeration (alas, at the price of a considerable slow-down), but in other applications (e.g. in cryptography) the use of extra randomness is undesirable.

In this paper, 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 classes of distributions having strong independence requirement. Under complexity assumptions, we show how to extract randomness from any *samplable* distribution, i.e. a distribution that can be generated by an efficient sampling algorithm.

Assuming that there are problems in E that are not solvable by subexponential-size circuits with $\mathbf{\Sigma_5}$ gates, we give a polynomial-time extractor that is able to transform any distribution of length n and min entropy $(1-\delta)n$ into an output distribution of length $(1-O(\delta)n)$ that is close to uniform, as long as the input distribution is samplable by a circuit whose size is a constant root of the running time of the extractor.

Our result is 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.

Keywords: extractors, list decoding, random self-reducibility

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

†MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139. E-mail: salil@theory.lcs.mit.edu. URL: http://theory.lcs.mit.edu/~salil. Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

1 Introduction

Randomness has proved to be a very useful tool in computer science. In algorithms, randomization 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]. In cryptography, secret keys must be chosen at random (otherwise, they are not secret), and even the cryptographic algorithms themselves, such as encryption, must 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. collecting 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 extractor*, 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 are known where k can be essentially arbitrary, m is very close to k, and t is $O(\log n)$ (see [ISW00] and 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).

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

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

of a 2^t slowdown factor. In other applications of randomness, such as probabilistic encryption [GM84], randomness is required by the very nature of the problem, and there is no possibility of trading off efficiency versus 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 general and interesting 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, as far as we know, there is no result providing a theoretical justification for this way using of 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] extracts randomness from a sequence of *independent* coin tosses of the same biased coin. An improved version by Elias [Eli72] extracts randomness at a rate close to the entropy of the source. Blum [Blu84], generalizing bon 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 [CGH$^+$85, BBR88, BL90, KKL88, LLS89, Fri92, CDH$^+$00, Dod00].

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 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 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, such as the model of [Blu86]. In addition to their generality, one can argue that samplable distributions are a reasonable model for distributions actually arising in nature (as argued, for example, by Levin [Lev86]).

Having settled on this class of sources, what we're looking for are functions $\text{EXT} : \{0,1\}^n \to \{0,1\}^m$ with the following property: for every source X of some min-entropy k which is samplable by a circuit of some size s, $\text{EXT}(X)$ 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.

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. The results that we obtain about such deterministic extractors are described below. As a first "plausibility" result, we show in Section A.1 the existence of good deterministic extractors[2] computed by polynomial-size circuits. Essentially, it's enough to properly pick a function from a collection of poly-wise independent hash functions. These results are reported in the appendix.

A Connection to Nondeterministic Average-case Hardness. While the above observations about nonuniform extractors 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, we show in Section A.1 that the existence of such extractors implies separations of complexity classes beyond what's currently known. Therefore, in order to construct uniform deterministic extractor, we will need to make complexity assumptions.

Let us consider for starters the task of extracting one almost unbiased bit (already a fairly non-trivial 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 $f(x)$ in the following way: given x, first check whether x is in the range of X, which is something that can be done efficiently using nondeterminism, if 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.[3]

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 3, 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 non-standard 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 [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 an inverse polynomial in the size of the

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

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

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 from [GL89]. 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 extractors.

The final result is that starting from a problem in $\mathbf{E}$ that does not admit circuits of size smaller than $2^{\delta n}$ with $\mathbf{\Sigma_4}$-gates, we get an efficient extractor that extracts one almost unbiased bit from any distribution of length n and min-entropy $(1 - O(\delta))n$ which is samplable by a circuit of size $s = s(n)$; the extractor runs in time $\mathrm{poly}(s^{1/\delta})$.

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 smaller than $2^{\delta n}$ with $\mathbf{\Sigma_5}$ gates, we get an efficient extractor that works for distributions of length n and min-entropy $(1 - \alpha)n$ sampled by circuits of size $s(n)$; the extractor has an output of length $(1 - O(\alpha))n$ and runs in time $\mathrm{poly}(s^{1/\alpha})$, where α is an arbitrarily small constant.

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 very complex and impractical solution. On the other hand, we think it's interesting to try and 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 are omitted in this abstract.

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 definition for 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.

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 $x \in \mathcal{U}$, $\Pr[X = x] \le 2^{-k}$. It will also be convenient for us to have the following equivalent terminology. X has *density* δ in $\mathcal{U}$ if for all $\max_{x\in\mathcal{U}} \Pr[X = x] = 1/(\delta \cdot \mathcal{U})$. 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 $\mathrm{SD}(X,Y) \le \varepsilon$, 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 $x \in \{0,1\}^n$, 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.[4] As shown by Nisan and Zuckerman [NZ96] it is necessary to invest $d \ge \Omega(\log(n-k) + \log 1/\varepsilon)$ truly random bits for any nontrivial extraction (i.e., when $m \le d-1$ and $k \le n-1$).[5] In order to make extraction possible without investing any truly random bits, we restrict to samplable distributions:

[4]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.

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

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* $U_{m(n)}$*.*

Nondeterministic circuits. We denote the levels of the polynomial-time hierarchy as follows: $\mathbf{\Delta_0} = \mathbf{\Sigma_0} = \mathbf{P}$, $\mathbf{\Sigma_{i+1}} = \mathbf{NP^{\Sigma_i}}$. 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*

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

We will make extensive use of the fact that that approximate counting and uniform sampling can be done in the hierarchy:

Theorem 2.4 ([Sto85, Sip83, JVV86]) *For any fixed* i*, there is a probabilistic* Σ_{i+1}*-algorithm* $\texttt{Approx}_i$ *such that for any* Σ_i*-circuit* $C : \{0,1\}^m \to \{0,1\}$*,*

$$\Pr\left[(1+\varepsilon) \cdot N \geq \texttt{Approx}_i(C, \varepsilon, \delta) \geq (1-\varepsilon) \cdot N\right] \geq 1 - \delta,$$

where $N = |\{x : C(x) = 1\}|$*. Moreover the running time of* $\texttt{Approx}_i(C, \varepsilon, \delta)$ *is* $\mathrm{poly}(|C|, 1/\varepsilon, \log(1/\delta))$*.*

Theorem 2.5 ([JVV86, BGP98]) *For any fixed* i*, there is a probabilistic polynomial-time* Σ_{i+1}*-algorithm* $\texttt{Sample}_i$ *such that for any* Σ_i*-circuit* $C : \{0,1\}^m \to \{0,1\}$*,* $\texttt{Sample}_i(C)$ *outputs a uniformly selected element of* $\mathrm{Acc}(C) \stackrel{\text{def}}{=} \{x \in \{0,1\}^m : C(x) = 1\}$*.*[6]

3 Extractors from Average-Case Hardness

Lemma 3.1 *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* $2^{\Delta} \cdot \epsilon$*-close to uniform.*

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 (see [CG88]). This is essentially 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 of samplable distributions, it is no more true (or at least no longer clear) that any samplable distribution of min-entropy k

[6]Actually, we allow $\texttt{Sample}_i(C)$ to output a failure symbol with some probability ($\leq 1/2$) and only require that its output be uniform over $\mathrm{Acc}(C)$ conditioned on non-failure. The failure probability can be reduced to an arbitrary δ by $\log(1\delta)$ independent trials.

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 let us 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 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 $B(x) = 0$ (and such 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.

Lemma 3.2 *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 $(n - \Delta, 2^\Delta \cdot \epsilon)$ extractor against circuit-size $(\epsilon s)^{\Omega(1)}$.*

4 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 δ that are much smaller 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 4.1 *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 $\delta = c\sqrt{d/|F|}$ fraction of points (where c is a universal constant), then there is a Σ_{i+1}-circuit C' of size $\mathrm{poly}(|C|, d)$ which computes p correctly everywhere.*[7]

This lemma implies that if we start with a function f which is worst-case hard for Σ_2-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 $\delta = c\sqrt{d/|F|}$ is very small, it is still a substantial

[7]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 $t \log |F|$.

relative advantage over random guessing, which would give success probability $1/|F|$. 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 4.2 *Let X be any distribution on $\{0,1\}^n$ of density δ and let $\varepsilon > 0$. Then*

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

The special case of Lemma 4.2 for flat distributions X can be deduced from a result of Chor and Goldreich [CG88]. Below we give a direct proof for arbitrary distributions.

Although Lemma 4.2 does not explicitly give an efficient decoding algorithm, we can easily obtain one using nondeterminism:

Lemma 4.3 *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.

To obtain deterministic extractors, we combine the polynomial encoding and Hadamard code via the standard "concatenation" technique. Let $\mathbb{F} = \mathrm{GF}(2^q)$,[8] 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 $\{0,1\}^q$.

In order to analyze this construction, we will need to argue that if a concatenated codeword (like p') is biased on on some distribution of sufficient density, then a noticeable fraction of the inner codewords (i.e., $\mathrm{Had}_{p(x)}$) are biased on the corresponding conditional distributions. This is provided by the following general lemma.

Lemma 4.4 *Let $f : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$ be any function, and let $X = (X_1, X_2)$ be any distribution on $\mathcal{A} \times \mathcal{B}$ of density δ. For every $a \in \mathcal{A}$ and $c \in \mathcal{C}$, let X^a denote distribution of X_2 conditioned on $X_1 = a$. Suppose that for some $c \in \mathcal{C}$, $\Pr\left[f(X) = c\right] \geq (1+\varepsilon)/|\mathcal{C}|$. Then, for at least a $\delta\varepsilon/3|\mathcal{C}|$ fraction of $a \in \mathcal{A}$, the following two conditions hold:*

1. $\Pr\left[f(a, X^a) = c\right] \geq (1+\varepsilon/3)/|\mathcal{C}|$.
2. *X^a has density at least $\delta\varepsilon/3|\mathcal{C}|$ in $\mathcal{B}$.*

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

Putting all the above tools together, we obtain the following theorem:

Theorem 4.5 *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_4-circuit*[9] *of size* $\mathrm{poly}(s, d, 1/\varepsilon)$ *which computes p' everywhere, provided that*

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

where c is a universal constant.

This immediately gives us a construction of deterministic extractors from Boolean functions that are worst-case hard for Σ_4-circuits.

Theorem 4.6 *There is a universal constant $\alpha > 0$ such that the following holds: Let $f : \{0,1\}^\ell \to \{0,1\}$ be such that no Σ_4-circuit of size s can compute f, where $\ell \leq s \leq 2^\ell$. Then for $s' = s^\alpha$ and any n satisfying $s' \geq n \geq \max\{\ell, (\ell/\log s')^2\}/\alpha$, there is a function $\mathrm{EXT}^f_{n,\ell,s} : \{0,1\}^n \to \{0,1\}$ such that*

1. *$\mathrm{EXT}^f_{n,\ell,s}$ is a $(n \cdot [1 - (\alpha \log s')/\ell], 1/s')$-deterministic extractor against circuit-size s'.*

2. *$\mathrm{EXT}^f_{n,\ell,s}$ is computable in time $\mathrm{poly}(n, 2^\ell)$ with oracle access to f.*

Corollary 4.7 *If there is a problem in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ which has Σ_4-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* $\mathrm{poly}(s)$.

5 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 taken from the "hard-core function" construction described in [Gol95]. Consider the function $\mathbf{C} : \{0,1\}^n \times \{0,1\}^{n+m} \to \{0,1\}^m$, defined as follows: $\mathbf{C}(x,y) = \mathbf{C}_1(x,y), \cdots, \mathbf{C}_m(x,y)$ where, for inputs $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_{n+m})$ we have

$$\mathbf{C}_i(x,y) = \langle (x_1, \ldots, x_n), (y_i, \ldots, y_{i+n-1}) \rangle$$

Notice that $\mathbf{C}(x,y)$ is independent of y_{n+m}. We could have defined $\mathbf{C}$ as a function $\mathbf{C} : \{0,1\}^n \times \{0,1\}^{n+m-1} \to \{0,1\}^m$, but it would have been annoying to carry the $(n+m-1)$ expression everywhere.

Lemma 5.1 *Let X be a distribution over $\{0,1\}^n$ of density δ and let $a \in \{0,1\}^m$. Then the number of strings y such that $\Pr[\mathbf{C}(X,y) = a] > 2^{-m} + \epsilon$. is at most $2^{2m}/\delta\epsilon^2$.*

Lemma 5.2 *For every fixed i, there is a probabilistic Σ_{i+2}-algorithm $\mathtt{CDecode}^{(i)}$ 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 there is an $a \in \{0,1\}^m$ such that $\Pr[\mathbf{C}(X,w) = a] > 2^{-m} + \epsilon$. Then $\mathtt{CDecode}^{(i)}(C, \varepsilon)$ runs in time $\mathrm{poly}(|C|, 1/\varepsilon, m)$ and outputs w with probability $\Omega(\delta \cdot \varepsilon^2 \cdot 2^{-2m})$.*

[9] 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.

Proof: Essentially identical to the proof of Lemma 4.3. ■

Theorem 5.3 *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 $p' : \mathbb{F}^t \times \{0,1\}^{q+m} \to \{0,1\}^m$ be its $\mathbf{C}$-encoding. Suppose there is a distribution X on $\mathbb{F}^k \times \{0,1\}^q$ which is of density δ and is samplable by size s, and an element $a \in \{0,1\}^m$ such that $\Pr[p'(X) = a] > 2^{-m} + \epsilon$. Then there is a Σ_4-circuit of size $\mathrm{poly}(s, d, 1/\varepsilon, m)$ which computes p' everywhere, provided that*

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

where c is a universal constant (not the same one of Theorem 4.5).

Proof: Essentially identical to the proof of Theorem 4.5. ■

Theorem 5.4 *There is a universal constant $\alpha > 0$ such that the following holds: Let $f : \{0,1\}^\ell \to \{0,1\}$ be such that no Σ_4-circuit of size s can compute f, where $\ell \leq s \leq 2^\ell$. Then for $s' = s^\alpha$ and any n satisfying $s' \geq n \geq \max\{\ell, (\ell/\log s')^2\}/\alpha$, there is a function $\mathrm{EXT}^f_{n,\ell,s} : \{0,1\}^n \to \{0,1\}^m$ such that*

1. $m = \frac{1}{2}\log s'$.
2. $\mathrm{EXT}^f_{n,\ell,s}$ *is a $(n \cdot [1 - (\alpha \log s')/\ell], 1/\sqrt{s'})$-deterministic extractor against circuit-size s'.*
3. $\mathrm{EXT}^f_{n,\ell,s}$ *is computable in time $\mathrm{poly}(n, 2^\ell)$ with oracle access to f.*

Corollary 5.5 *If there is a problem in $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ which has Σ_4-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 $\mathrm{poly}(s)$.*

Lemma 5.6 *There is a constant $\alpha > 0$ such the following holds. Let X be a distribution of min-entropy $n_1 + n_2 - \Delta$ ranging over $\{0,1\}^{n_1+n_2}$, and let us view X as a pair (X_1, X_2) where X_1 ranges over $\{0,1\}^{n_1}$ and X_2 ranges over $\{0,1\}^{n_2}$. Let X be samplable by a circuit of size s, let $\mathrm{EXT}_1 : \{0,1\}^{n_1} \times \{0,1\}^t \to \{0,1\}^{m_1}$ be a $(n_1 - \Delta, \epsilon)$-extractor, and let $\mathrm{EXT}_2 : \{0,1\}^{n_2} \to \{0,1\}^{m_2}$ be a $(n_2 - \Delta - \log(1/\epsilon), \epsilon)$-deterministic extractor against Σ_1-circuit-size s^α. Then $\mathrm{EXT}(X_1, X_2) = \mathrm{EXT}_1(X_1, \mathrm{EXT}_2(X_2))$ is 3ϵ-close to uniform.*

Theorem 5.7 ([Zuc97]) *For every $\gamma > 0$ there is a constant c_γ and an explicit construction of a $((1 - 2\gamma)n, 1/6n)$-extractor $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ where $t = c_\gamma \log n$ and $m = (1 - 3\gamma)n$.*

Theorem 5.8 *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 for every sufficiently small constant δ and for every s there is a $(1 - \delta, 1/n)$-extractor $\mathrm{EXT} : \{0,1\}^n \to \{0,1\}^m$ against circuit size s where $m = (1 - O(\delta))n$. EXT is computable in time $\mathrm{poly}(s)$, where the exponent of the polynomial depends on δ.*

Acknowledgments

We thank Avi Wigderson, Oded Goldreich, and Yevgeniy Dodis for helpful discussions.

References

[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*.

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287, Santa Fe, New Mexico, 20–22 November 1994. IEEE.

[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).

[Blu84] Manuel Blum. Independent unbiased coin flips from a correlated biased source: a finite state Markov chain. In *25th Annual Symposium on Foundations of Computer Science*, pages 425–433, Singer Island, Florida, 24–26 October 1984. IEEE.

[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.

[Dod00] Yevgeniy Dodis. Impossibility of black-box reduction from non-adaptively to adaptively secure coin-flipping. Unpublished manuscript, 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.

[Fri92] Joel Friedman. On the bit extraction problem. In *33rd Annual Symposium on Foundations of Computer Science*, pages 314–319, Pittsburgh, Pennsylvania, 24–27 October 1992. IEEE.

[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.

[Gol95] Oded Goldreich. *Foundations of Cryptography (Fragments of a Book)*. Weizmann Institute of Science, 1995. Available, along with revised version 1/98, from `http://www.wisdom.weizmann.ac.il/~oded`.

[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.

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

[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.

[ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of 32nd ACM Symposium on Theory of Computing*, 2000.

[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.

[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.

[KKL88] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science*, pages 68–80, White Plains, New York, 24–26 October 1988. IEEE.

[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.

[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.

[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.

A Appendix

A.1 Nonuniform Extractors & Negative Results

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

Proof: Let $N_s = s^{O(s)}$ be the number of circuits of size s, $t = 2\log(k + N_s)$ and $m = k - 2\log(1/\varepsilon) - \log t - 2$. We choose $\textsc{Ext}$ randomly from a family of t-wise independent functions from $\{0,1\}^n \to \{0,1\}^m$, and argue that it is a (k, ε)-deterministic extractor against circuit size s with high probability. Consider any fixed distribution X of min-entropy k that is samplable by size s. A standard application of the t-moment method (to be given in more detail shortly) yields:

$$\Pr_{\textsc{Ext}}[\textsc{Ext}(X) \text{ is not } \varepsilon\text{-close to uniform}] < 1/N_s \tag{1}$$

Taking a union bound over distributions samplable by size s shows that there exists an (s, k, ε)-deterministic extractor from this family. We note there are families of t-wise independent functions computable by circuits of size $\mathrm{poly}(t, n)$.

We now justify Inequality (1). Consider any fixed $y \in \{0,1\}^m$. The probability mass that y gets under $\textsc{Ext}(X)$ is

$$\mathrm{Mass}_y = \sum_{x \in \{0,1\}^n} p_x \cdot \chi_{x,y},$$

where p_x is the probability mass of x under X and $\chi_{x,y}$ is the indicator variable for the event $[\textsc{Ext}(x) = y]$. For a fixed y, the variables $\{\chi_{x,y}\}$ are t-wise independent and have expectation $\mu = 1/2^m$ (over the choice of $\textsc{Ext}$). Since X has min-entropy k, we have $p_x \cdot \chi_{x,y} \in [0, 2^{-k}]$. Applying a tail inequality for sums of t-wise independent variables from [BR94], we have

$$\Pr[|\mathrm{Mass}_y - \mu| \ge \varepsilon \cdot \mu] \le 8 \cdot \left[\frac{t \cdot 2^k \cdot \mu + t^2}{(2^k \cdot \varepsilon \cdot \mu)^2}\right]^{t/2} \le \frac{1}{2^{t/2}} < \frac{1}{2^k} \cdot \frac{1}{N_s}.$$

Hence, with probability greater than $1 - 1/N_s$, $\mathrm{Mass}_y \le (1+\varepsilon)/2^m$ for all y, which implies that $\textsc{Ext}(X)$ is ε-close to uniform. ■

A similar argument gives nonuniform extractors for *uniform* samplers.

Proposition A.2 *For all functions $t(n)$, $k(n) \le n$, and $\varepsilon(n)$ there exists a $(k(n), \varepsilon(n))$-extractor* $\{\textsc{Ext}_n : \{0,1\}^n \to \{0,1\}^{m(n)}\}$ *against time $t(n)$, with $m(n) = k(n) - 2\log(1/\varepsilon(n)) - O(\mathrm{loglog} n)$. Moreover,* $\textsc{Ext}_n$ *can be computed by a circuit of size* $\mathrm{poly}(n)$.

Note that in Proposition A.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 A.3 *There is a constant c such that no function* $\textsc{Ext} : \{0,1\}^n \to \{0,1\}$ *computable by a circuit of size s is a $(n-1, 1/5)$-deterministic extractor against circuit size $c \cdot s$.*[10]

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

Proof: Without loss of generality, we may assume that $\textsc{Ext}(x) = 1$ 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 $\textsc{Ext}(x) = 1$, 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, $\textsc{Ext}(X) = 1$ with probability at least $3/4$. ■

A similar argument applies to uniform deterministic extractors for uniform samplers (but not to nonuniform deterministic extractors for uniform samplers, as demonstrated by Proposition A.2).

Proposition A.4 *There is a constant c such that no family of functions* $\{\textsc{Ext}_n : \{0,1\}^n \to \{0,1\}\}$ *computable in time* $t(n)$ *is a* $(n-1, 1/5)$*-deterministic extractor against time* $t(n)$.

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 A.5 *Suppose* $\{\textsc{Ext}_n : \{0,1\}^n \to \{0,1\}\}$ *is a family of functions computable in time* $t(n)$ *such that, for every* n, $\textsc{Ext}_n$ *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 $L = \{x \in \{0,1\}^* : \textsc{Ext}_{|x|}(x) = 1\}$. Proposition A.3 implies that this language has circuit complexity at least $s(n)/c$. ■

A similar proof, noting that Proposition A.4 holds even if the extractor is computable by a randomized algorithm, yields:

Corollary A.6 *Suppose* $\{\textsc{Ext}_n : \{0,1\}^n \to \{0,1\}\}$ *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* $\mathbf{DTIME}(t(n)) \setminus \mathbf{BPTIME}(\Omega(t'(n))$.

A.2 Proofs Omitted From Section 3

Proof: [Of Lemma 3.1] Let $X(\cdot)$ be a circuit of size s' that samples a flat distribution of min-entropy $n - \Delta$ such that $\Pr_a[f(X(a)) = 1] > 1/2 + \epsilon'/2$ (the proof would be analogous in case $\Pr_a[f(X(a)) = 0] > 1/2 + \epsilon'/2$), where $\epsilon' = 2^{\Delta} \cdot \epsilon$. Consider the following algorithm A (that tries to approximate f): on input x, if x is in the range of X then output 1, otherwise output a random bit. A can be implemented by a nondeterministic circuit of size $s' + O(n)$. It follows from the definition of A that

$$\begin{aligned}
\Pr_{x \in \{0,1\}^n}[A(x) = f(x)] &= \Pr_x[A(x) = f(x) | x \text{ in the range of } X] \cdot 2^{-\Delta} \\
&\quad + \Pr_x[A(x) = f(x) | x \text{ not in the range of } X] \cdot (1 - 2^{-\Delta}) \\
&> \left(\frac{1}{2} + \frac{\epsilon'}{2}\right) \cdot 2^{-\Delta} + \frac{1}{2} \cdot \left(1 - 2^{-\Delta}\right) \\
&= \frac{1}{2} + \frac{\epsilon'}{2} 2^{-\Delta} == \frac{1}{2} + \frac{\epsilon}{2}
\end{aligned}$$

that contradicts our assumption on the hardness of f, if $s' = s - O(n)$. ■

Proof: [Of Lemma 3.2] Let X be a sampler of size s' such that

$$\Pr_a[f(X(a)) = 1] > 1/2 + \epsilon'/2$$

We now describe a Σ_1 circuit A of size $\mathrm{poly}(s', 1/\epsilon)$ such that A approximates f on a fraction $1/2 + 2^{-\Delta} \cdot \epsilon'/2$ of the inputs.

We first describe A as a randomized circuit' the randomness can be nonuniformly fixed at the end of the construction. For every $x \in \{0,1\}^n$, set $p_x = \Pr_a[X(a) = x]$. On input x, A computes a value q_x such that $q_x(1 - \epsilon') \leq p_x \leq q_x(1 + \epsilon')$. After that, A outputs 1 with probability $2^{n-\Delta} q_x$, and it outputs a random bit with probability $1 - 2^{n-\Delta} q_x$. By approximate counting (Theorem 2.4), A can be implemented as a probabilistic Σ_1-circuit of size $\mathrm{poly}(s', 1/\epsilon)$.

We have

$$\Pr[f(X) = 1] = \sum_{x:f(x)=1} p_x > \frac{1}{2} + \epsilon'$$

$$\Pr[f(X) = 0] = \sum_{x:f(x)=0} p_x < \frac{1}{2} - \epsilon'$$

and

$$\begin{aligned}
\Pr_x[A(x) = f(x)] &= \Pr_x[A(x) = f(x) = 1] + \Pr_x[A(x) = f(x) = 0] \\
&= 2^{-n} \sum_{x:f(x)=1} \left(\frac{1}{2} + \frac{2^{n-\Delta} q_x}{2}\right) + 2^{-n} \sum_{x:f(x)=0} \left(\frac{1}{2} - \frac{2^{n-\Delta} q_x}{2}\right) \\
&= \frac{1}{2} + \frac{2^{-\Delta}}{2} \cdot \left[\sum_{x:f(x)=1} q_x - \sum_{x:f(x)=0} q_x\right] \\
&\geq \frac{1}{2} + \frac{2^{-\Delta}}{2} \cdot \left[(1 - \epsilon') \sum_{x:f(x)=1} p_x - (1 + \epsilon) \sum_{x:f(x)=0} p_x\right] \\
&= \frac{1}{2} + \frac{2^{-\Delta}}{2} \cdot \left[(1 - \epsilon') \Pr[f(X) = 1] - (1 + \epsilon') \Pr[f(X) = 0]\right] \\
&= \frac{1}{2} + \frac{2^{-\Delta}}{2} \cdot (2 \Pr[f(X) = 1] - \epsilon' - 1) \\
&\geq \frac{1}{2} + \frac{2^{-\Delta} \epsilon'}{2}
\end{aligned}$$

■

A.3 Proofs Omitted From Section 4

Proof: [Of Lemma 4.1] For $x, y \in \mathbb{F}^t$, the *line through x and y* is the parametrized set of points $\{\ell_{x,y}(t) \stackrel{\text{def}}{=} (1-t)x + ty | t \in \mathbb{F}\}$. For a function $f : \mathbb{F}^m \to \mathbb{F}$, *$f$ restricted to the line $\ell_{x,y}$* is the function $f|_{\ell_{x,y}} : \mathbb{F} \to \mathbb{F}$ defined by $f|_{\ell_{x,y}}(t) = f(\ell_{x,y}(t))$. Note that $p(\ell_{x,y}(t))$ is a univariate polynomial of degree at most d. It is shown in [STV99, Lemma 28] that there exists a point $z \in \mathbb{F}^m$ such that for at least a $15/16$ fraction of points $x \in \mathbb{F}^m$, we have:

1. $p|_{\ell_{z,x}}$ and $C|_{\ell_{z,x}}$ agree on at least a $\delta/2$ fraction of $\mathbb{F}$.

2. There does not exist any degree d polynomial $h : \mathbb{F} \to \mathbb{F}$ other than $p|_{\ell_{z,x}}$ which agrees with $C|_{\ell_{z,x}}$ in at least a $\delta/4$ fraction of $\mathbb{F}$ and satisfies $h(0) = p(z)$.

Fix such a z; z and $p(z)$ will be nonuniformly hardwired into all the circuits we construct. By approximate counting (Theorem 2.4), there is a probabilistic Σ_{i+1}-circuit C' which, on input (x, h) (where $x \in \mathbb{F}^t$ and $h : \mathbb{F} \to \mathbb{F}$ is a degree d polynomial) (a) outputs 1 with high probability if h agrees with $C|_{\ell_{z,x}}$ in at least a $\delta/2$ fraction of $\mathbb{F}$ and $h(0) = p(z)$, and (b) outputs 0 with high probability if h agrees with $C|_{\ell_{z,x}}$ in less than a $\delta/4$ fraction of $\mathbb{F}$ or $h(0) \neq p(z)$. Moreover the size of C' is $\mathrm{poly}(|C|, d)$. After sufficient error reduction, the coin tosses of C' can be nonuniformly fixed so that it correctly distinguishes these two cases for all x and h. This yields the following Σ_{i+2}-circuit C'' for computing p almost everywhere:

$C''(x)$**:**

1. Use nondeterminism to find an h such that $C'(x, h) = 1$ (if one exists).
2. Output $h(1)$.

C'' is of size $\mathrm{poly}(|C|, d)$ and computes p in at least a $15/16$ fraction of points. The "self-corrector" for polynomials given in [GLR$^+$91] converts C'' into a circuit C''' that computes p everywhere. ■

Proof: [Of Lemma 4.2] The proof is based on the finite Fourier transform. For two real valued functions $f, g : \{0,1\}^n \to \mathbb{R}$, define their inner product to be

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) g(x).$$

For $w \in \{0,1\}^n$, define $L_w(x) = (-1)^{w \cdot x}$, where $w \cdot x$ denotes inner product mod 2. It is well-known that $\{L_w\}_{w \in \{0,1\}^n}$ form an orthnormal basis (called the Fourier basis) for the 2^n-dimensional vector space of real-valued functions on $\{0,1\}^n$. Now let $\mu : \{0,1\}^n \to \mathbb{R}$ be the probability mass function of X, i.e. $\mu(x) = Pr[X = x]$. For $w \in \{0,1\}^n$, the bias of $\mathrm{Had}_w(X)$ is exactly $|2^n \cdot \langle \mu, L_w \rangle|$. By Parseval's inequality,

$$\sum_{w \in \{0,1\}^n} |2^n \cdot \langle \mu, L_w \rangle|^2 = 2^{2n} \cdot \langle \mu, \mu \rangle = 2^n \cdot \sum_{x \in \{0,1\}^n} \mu(x)^2 \leq 2^n \cdot \sum_{x \in \{0,1\}^n} \mu(x) \cdot \frac{1}{\delta \cdot 2^n} = \frac{1}{\delta}.$$

Hence there are at most $1/(\delta \cdot \varepsilon^2)$ values of w such that $\mathrm{Had}_w(X)$ has bias at least ε. ■

Proof: [Of Lemma 4.3]

By approximate counting (Theorem 2.4), there is a probabilistic Σ_{i+1}-algorithm $\mathtt{Test}_i(C, \varepsilon, v)$ running in time $\mathrm{poly}(|C|, 1/\varepsilon)$ that (a) outputs 1 with probability at least $1 - 2^{-n-1}$ if $\mathrm{Had}_v(X)$ has bias at least ε and (b) outputs 0 probability at least $1 - 2^{-n-1}$ if $\mathrm{Had}_v(X)$ has bias at most $\varepsilon/2$. Thus, with probability least $1/2$ over the choice of the random coins r of $\mathtt{Test}_i$, $C'(v) \stackrel{\mathrm{def}}{=} \mathtt{Test}_i(C, \varepsilon, v; r)$ is a Σ_{i+1}-circuit which distinguishes these two cases correctly for all v. In particular, $C'(w) = 1$ and, by Lemma 4.2, $|\{v : C'(v) = 1\}| \leq (\varepsilon/2)^2/2^{n-k}$. Hence applying uniform sampling (Theorem 2.5) to this circuit gives the desired result. More formally, the procedure $\mathtt{HadDecode}_i$ does the following:

$\texttt{HadDecode}_i(C, \varepsilon)$:

1. Uniformly select coins r for $\texttt{Test}_i$.
2. Let C' be the Σ_{i+1}-circuit defined by $C'(v) = \texttt{Test}_i(C, \varepsilon, v; r)$.
3. Run $\texttt{Sample}_{i+1}(C')$.

■

Proof: [Of Lemma 4.4] Note that, for any $b \in \mathcal{B}$,

$$\Pr[X^a = b] = \frac{\Pr[X = (a,b)]}{\Pr[X_1 = a]} \leq \frac{1}{\delta \cdot |\mathcal{A}| \cdot |\mathcal{B}| \cdot \Pr[X_1 = a]},$$

so to achieve Condition 2, it suffices to have $\Pr[X_1 = a] \geq \varepsilon/(3|\mathcal{A}||\mathcal{C}|)$. Now suppose that the conclusion of the lemma does not hold. Then for greater than a $1 - \varepsilon\delta/3|\mathcal{C}|$ fraction of $a \in \mathcal{A}$, we have

$$\begin{aligned} \Pr[X_1 = a \text{ and } f(X) = c] &= \Pr[X_1 = a] \cdot \Pr[f(a, X^a) = c] \\ &< \Pr[X_1 = a] \cdot \left(\frac{(1+\varepsilon/3)}{|\mathcal{C}|}\right) + \frac{\varepsilon}{3|\mathcal{A}||\mathcal{C}|} \end{aligned}$$

For the remaining $a \in \mathcal{A}$, we certainly have

$$\Pr[X_1 = a \text{ and } f(X) = c] \leq \Pr[X_1 = a] \leq \frac{1}{\delta|\mathcal{A}|},$$

since X has density δ. Putting everything together, we have

$$\begin{aligned} \Pr[f(X) = c] &= \sum_{a \in \mathcal{A}} \Pr[X_1 = a \text{ and } f(X) = c] \\ &< \sum_{a \in \mathcal{A}} \Pr[X_1 = a] \cdot \frac{(1+\varepsilon/3)}{|\mathcal{C}|} + |\mathcal{A}| \cdot \left(\frac{\varepsilon}{3|\mathcal{C}|}\right) + \frac{\varepsilon\delta|\mathcal{A}|}{3|\mathcal{C}|} \cdot \left(\frac{1}{\delta|\mathcal{A}|}\right) \\ &= \frac{1+\varepsilon}{|\mathcal{C}|}, \end{aligned}$$

which is a contradication. ■

Proof: [Of Theorem 4.5] For every $x \in \mathbb{F}^t$, define the conditional distribution X^x on $\{0,1\}^q$ as in Lemma 4.4. By uniform sampling (Theorem 2.5), each X^x is samplable by a Σ_1-circuit C_x of size $\mathrm{poly}(s)$. By Lemma 4.4, for at least a $\varepsilon\delta/6$ fraction of $x \in \mathbb{F}^t$, the following two conditions hold:

1. $\mathrm{Had}_{p(x)}(X^x) = p'(x, X^x)$ has bias at least $\varepsilon/3$.
2. X^x has density at least $\delta\varepsilon/6$ in $\{0,1\}^q$.

By Lemma 4.3, for every x for which these two conditions hold, $\texttt{HadDecode}_1(C_x, \varepsilon/3)$ outputs $p(x)$ with probability at least $\Omega((\varepsilon/3)^2 \cdot (\delta\varepsilon/6)) = \Omega(\delta \cdot \varepsilon^3)$. By averaging, there exists a setting r of the random coins of this procedure such that $C'(x) = \texttt{HadDecode}_1(C_x, \varepsilon/3; r)$ outputs $p(x)$ for at least a $(\delta\varepsilon/6) \cdot \Omega(\delta \cdot \varepsilon^3) = \Omega(\delta^2 \cdot \varepsilon^4)$ fraction of x's. C' is a Σ_3-circuit of size $\mathrm{poly}(|C|, 1/\varepsilon)$. By Lemma 4.1, there is a Σ_4-circuit of size $\mathrm{poly}(|C|, d, 1/\varepsilon)$ computing p everywhere. ■

Proof: [Of Theorem 4.6]

Let $t = \lceil \ell / \log s' \rceil \geq 1/\alpha$, $q = \lceil n/(t+1) \rceil$, and $\mathbb{F} = \mathrm{GF}(2^q)$. Let H be a subset of $\mathbb{F}$ of size s', and fix some injective map $\tau : \{0,1\}^\ell \to H^t$. There exists a polynomial $p : \mathbb{F}^t \to \mathbb{F}$ of degree at most s' in each variable such that for all $x \in \{0,1\}^\ell$, $f(x) = p(\tau(x))$; moreover such a polynomial can be evaluated at any point of $\mathbb{F}^t$ in time $\mathrm{poly}(n, 2^\ell)$. p has total degree at most $d = s't$. Let $p' : \mathbb{F}^t \times \{0,1\}^q \to \{0,1\}$ be the Hadamard encoding of p, and, for $x \in \{0,1\}^n$, define $\textsc{Ext}^f_{n,\ell,s}(x) = p'(x0^j)$, where $j = (t+1)q - n \leq t$.

Now suppose that there is distribution X on $\{0,1\}^n$ such that X has min-entropy $n \cdot [1 - (\alpha \log s')/\ell)]$, X is samplable by size s', and $\textsc{Ext}^f_{n,\ell,s}(X)$ has bias at least $1/s'$. Then the distribution $p'(X')$ has bias at least $1/s'$, where $X' = X0^j$. X' has density at least

$$\delta = \frac{2^{n \cdot [1 - (\alpha \log s')/\ell]}}{2^{(t+1)q}} \geq \frac{1}{2^{t + (\alpha n \log s')/\ell}}. \tag{2}$$

In order to apply Theorem 4.5, we need

$$\delta^2 \cdot \left(\frac{1}{s'}\right)^4 \geq c\sqrt{\frac{d}{|F|}} = c\sqrt{\frac{s't}{2^q}},$$

i.e.

$$t^2 \cdot (s')^9 / \delta^4 \leq 2^q / c^2.$$

By Inequality (2), we have

$$\begin{aligned}
4\log(1/\delta) + 9\log s' + 2\log t &\leq 4 \cdot \left[t + \frac{\alpha n \log s'}{\ell}\right] + 9\log s' + 2\log t \\
&\leq 6t + \frac{13\alpha n \log s'}{\ell} \\
&\leq \frac{6\ell}{\log s'} + 1 + \frac{13\alpha n \log s'}{\ell} \\
&\leq \frac{19\alpha n \log s'}{\ell} + 1 \\
&\leq \frac{19\alpha n}{t-1} + 1 \\
&\leq \frac{20\alpha n}{t+1} + 1 \\
&\leq 20\alpha q + 1.
\end{aligned}$$

So,

$$t^2 \cdot (s')^9/\delta^4 \leq 2^{20\alpha q + 1} \leq 2^q / c^2,$$

for sufficiently small α. Hence Theorem 4.5 applies, and we conclude that p' (and hence also f) can be computed by a Σ_4-circuit of size $\mathrm{poly}(s', d, 1/s') = \mathrm{poly}(s^\alpha) \leq s$ for sufficiently small α. This contradicts the hardness of f. ■

A.4 Proofs Omitted From Section 5

Proof: [Of Lemma 5.1]

The Vazirani XOR Lemma [Vaz84] says that if $c_1, \ldots, c_m$ and $a_1, \ldots, a_m$ are arbitrary 0/1 random variables with arbitrary dependencies, then

$$\Pr[(a_1, \ldots, a_m) = (c_1, \ldots, c_m)] = \frac{1}{2^m} + \sum_{I \subseteq \{1,\ldots,m\}, I \neq \emptyset} \frac{1}{2^m}(2\Pr[\oplus_{i \in I} a_i = \oplus_{i \in I} c_i] - 1)$$

A proof of the above statement can be found in, e.g., [Gol95, Pf of Lemma 2.5.6]. So if there is an $a \in \{0,1\}^l$ such that

$$\Pr[\mathbf{C}(X,y) = a] > 2^{-m} + \epsilon$$

then there is also a non-empty subset $I \subseteq \{1,\dots,m\}$ and a bit $b = \bigoplus_{i\in I} a_i$ such that

$$\Pr\left[\bigoplus_{i\in I} \mathbf{C}_i(X,y) = b\right] > \frac{1}{2} + \frac{\epsilon}{2}$$

Using the definition of $\mathbf{C}$ and the linearity of the inner product operator, this is the same as

$$\Pr[\langle X, \bigoplus_{i\in I}(y_i,\dots,y_{i+n-1})\rangle = b] > \frac{1}{2} + \frac{\epsilon}{2}$$

We know from Lemma 4.2 that there can be at most $1/\epsilon^2\delta$ strings $z \in \{0,1\}^n$ such that

$$\Pr[\langle X, z\rangle = b] > \frac{1}{2} + \frac{\epsilon}{2}$$

so there are only so many possible values for $\bigoplus_{i\in I}(y_i,\dots,y_{i+n-1})$. On the other hand, the function mapping y into $\bigoplus_{i\in I}(y_i,\dots,y_{i+n-1})$ is a full-rank linear map, and so it is a regular 2^m-to-1 function; furthermore this map is totally specified by giving the set I (and there are $2^m - 1$ choices for it). It follows that B cannot contain more than $(2^m - 1)\cdot 2^m/\epsilon^2\delta$ elements of $\{0,1\}^{n+m}$. ■

Proof: [Of Lemma 5.8] From the assumption of the theorem, using Corollary 5.5, it follows that there is a constant γ such that for every constant c, every n_2, and every $n_2^c \le s \le 2^{\gamma n}$, there is a $((1-\gamma)n_2, 1/s)$ extractor $\mathrm{EXT} : \{0,1\}^{n_2} \to \{0,1\}^{c\log n_2}$ for Σ_1-circuit size s.

Let δ be a fixed constant such that $\delta < \gamma/2$. Let X be a distribution ranging over $\{0,1\}^n$, of min-entropy $(1-\delta)$, and samplable with a circuit of size s. We view X as a pair (X_1, X_2), where X_1 ranges over $\{0,1\}^{n_1}$ and X_2 ranges over $\{0,1\}^{n_2}$, with $n_1 = (1-\delta/\gamma)n$ and $n_2 = \delta n/\gamma$. Notice that $n_1 > n/2$.

Let c_δ be such that the construction of [Zuc97] cited in Theorem 5.7 gives a $((1-2\delta)n_1, 1/6n_1)$-extractor $\mathrm{EXT}_1 : \{0,1\}^{n_1} \times \{0,1\}^t \to \{0,1\}^{m_1}$ with $m_1 = (1-3\delta)n_1$ and $t = c_\delta \log n_1$. We will worsen the parameters of EXT_1 a bit, to simplify subsequent calculations, and we will see it as $(n_1 - \delta n, 1/3n)$-extractor $\mathrm{EXT}_1 : \{0,1\}^{n_1} \times \{0,1\}^t \to \{0,1\}^{m_1}$ with $m_1 = (1-3\delta-\delta/\gamma)n > (1-3\delta/\gamma)n$.

We also have a $(((1-\gamma)n_2 - \log 3n, 1/3n)$ deterministic extractor $\mathrm{EXT}_2 : \{0,1\}^{n_2} \to \{0,1\}^{m_2}$ for Σ_1-circuit size s, where $m_2 = c_\delta \log n_2$.

By combining these two extractor using Lemma 5.6, we are done. ■