comput. complex. **13** (2004), 147 – 188
1016-3328/04/030147–42
DOI 10.1007/s00037-004-0187-1

© Birkhäuser Verlag, Basel 2004

computational complexity

THE COMPLEXITY OF CONSTRUCTING PSEUDORANDOM GENERATORS FROM HARD FUNCTIONS

Emanuele Viola

Abstract. We study the complexity of constructing pseudorandom generators (PRGs) from hard functions, focussing on constant-depth circuits. We show that, starting from a function $f : \{0,1\}^l \to \{0,1\}$ computable in alternating time $O(l)$ with $O(1)$ alternations that is hard *on average* (i.e. there is a constant $\epsilon > 0$ such that every circuit of size $2^{\epsilon l}$ fails to compute f on at least a $1/\text{poly}(l)$ fraction of inputs) we can construct a PRG : $\{0,1\}^{O(\log n)} \to \{0,1\}^n$ computable by *DLOGTIME*-uniform constant-depth circuits of size polynomial in n. Such a PRG implies $BP \cdot AC^0 = AC^0$ under *DLOGTIME*-uniformity.

On the negative side, we prove that starting from a *worst-case* hard function $f : \{0,1\}^l \to \{0,1\}$ (i.e. there is a constant $\epsilon > 0$ such that every circuit of size $2^{\epsilon l}$ fails to compute f on some input) for every positive constant $\delta < 1$ there is no *black-box* construction of a PRG : $\{0,1\}^{\delta n} \to \{0,1\}^n$ computable by constant-depth circuits of size polynomial in n.

We also study worst-case hardness amplification, which is the related problem of producing an average-case hard function starting from a worst-case hard one. In particular, we deduce that there is no black-box worst-case hardness amplification within the polynomial time hierarchy. These negative results are obtained by showing that polynomial-size constant-depth circuits cannot compute good extractors and list-decodable codes.

Keywords. Pseudorandom generator, hardness, constant-depth circuit, *DLOGTIME*-uniformity, noise sensitivity.

Subject classification. 68Q01.

1. Introduction

A rigorous notion of *pseudorandom generators* (PRGs) was introduced in the seminal works of Blum & Micali (1984) and Yao (1982), and has since found a striking variety of applications in cryptography and complexity theory. A PRG

$G : \{0,1\}^u \to \{0,1\}^n$ is an efficient procedure that stretches u input bits into $n \gg u$ output bits such that the output of the PRG is indistinguishable from random to small circuits. That is, for every circuit A of size n we have

$$\left| \Pr_{x \in \{0,1\}^u}[A(G(x)) = 1] - \Pr_{x \in \{0,1\}^n}[A(x) = 1] \right| \leq 1/n.^1$$

Throughout this work the complexity of a PRG is measured in terms of its *output length* (denoted n above).

While the existence of PRGs is a major open problem, there has been a series of fascinating works constructing PRGs from weaker and weaker assumptions.

Nisan & Wigderson (1994) show how to construct PRGs from strong *average-case* hardness assumptions, namely the existence of a Boolean function $f : \{0,1\}^l \to \{0,1\}$ in $E := TIME(2^{O(l)})$ that is hard *on average* for circuits. In particular, they show that starting from a function $f : \{0,1\}^l \to \{0,1\}$ in E such that (for some constant $\epsilon > 0$) every circuit of size $2^{\epsilon l}$ fails to compute f on at least a $1/2 - 1/2^{\epsilon l}$ fraction of inputs, it is possible to construct a PRG with logarithmic input length, i.e. $G : \{0,1\}^{O(\log n)} \to \{0,1\}^n$, computable in time polynomial in n. Efficient PRGs with logarithmic seed length are of particular interest because, as we discuss below, they imply $BP \cdot P = P$. Note that the PRG G is computable in time polynomial in n even though it is constructed from a function f that is computable in exponential time (i.e. in E). This is possible because G only evaluates f on inputs of length $l = O(\log n)$.

Because the *average-case* hardness assumption in Nisan & Wigderson (1994) seemed very strong, much research was devoted to constructing PRGs with logarithmic seed length under a weaker *worst-case* hardness assumption, namely the existence of a function $f : \{0,1\}^l \to \{0,1\}$ in E such that (for some constant $\epsilon > 0$) every circuit of size $2^{\epsilon l}$ fails to compute f on *some* input (e.g. Babai *et al.* 1993; Impagliazzo 1995). This research culminated in Impagliazzo & Wigderson (1997) where it is shown how to amplify the worst-case hardness of functions in E to the average-case hardness required in Nisan & Wigderson (1994). We survey these results in Section 3. More direct proofs and improved results were obtained in Sudan *et al.* (2001); Impagliazzo *et al.* (2000); Shaltiel & Umans (2001); Umans (2002).

[1] The original definition of Blum & Micali (1984) and Yao (1982) is different in that it requires that every polynomial-size circuit has negligible advantage in distinguishing the output of the PRG from random. For derandomization purposes it is enough to fix a universal constant c and require that every circuit of size n^c has advantage at most $1/n^c$ in distinguishing the output of the PRG from random. Also, since the circuit can ignore part of the input, we can set $c = 1$. In this paper we adopt this latter definition of PRG.

1.1. The problem we study. In this paper we address the following problem: What is the complexity of constructing a PRG from a hard function? There are at least two reasons for studying this problem. First, we want to understand the computational relationship between two fundamental quantities in theoretical computer science: hardness and randomness. Second, PRGs are a basic tool whose variety of applications justifies the quest for more and more efficient constructions.

Derandomization. An important application that demands more and more efficient PRGs is the *high-end derandomization* of a complexity class C, that is, proving $BP \cdot C = C$. For such application we need PRGs with logarithmic seed length, which as we said above can be constructed starting from a function having exponential circuit complexity. For example, Impagliazzo & Wigderson (1997) show that $BP \cdot P = P$ if there is a function $f : \{0,1\}^l \to \{0,1\}$ in E that requires circuits of size $2^{\Omega(l)}$. This derandomization works as follows. We run the algorithm we want to derandomize using all the possible outputs of the PRG in place of true random bits. Then we decide according to majority vote. Since the seed length is logarithmic, this process is efficient, i.e. only gives a polynomial slow-down.

It is then clear that if we aim to derandomize a probabilistic complexity class $BP \cdot C$ using a PRG, then the PRG must be computable in C. Therefore, the lower the complexity class we want to derandomize, the more efficient the PRG must be. For example, already to derandomize $BP \cdot L$ (where $L := SPACE(\log n)$), one needs a more efficient PRG construction than the one given in Nisan & Wigderson (1994) and used in Impagliazzo & Wigderson (1997) to derandomize $BP \cdot P$. This problem is solved by Klivans & van Melkebeek (1999) who obtain $BP \cdot L = L$ under the assumption that there is a function $f : \{0,1\}^l \to \{0,1\}$ computable in linear space that requires circuits of size $2^{\Omega(l)}$.

In this paper we study more efficient PRG constructions that could be used to derandomize probabilistic classes below $BP \cdot L$. In particular, we aim to prove $BP \cdot AC^0 = AC^0$ subject to *DLOGTIME*-uniformity, where AC^0 denotes the class of functions computable by polynomial-size constant-depth circuits, and *DLOGTIME*-uniformity is a strong uniformity condition discussed below (at the end of Section 1.2).

Note that the high-end derandomization $BP \cdot AC^0 = AC^0$ subject to *DLOGTIME*-uniformity is not known to hold unconditionally. Indeed, it is still open whether the weaker inclusion P-uniform $BP \cdot AC^0 \subseteq P$ is true: The most efficient unconditional derandomization of P-uniform $BP \cdot AC^0$ is the one obtained by Nisan (1991) that runs in quasipolynomial time. Such de-

randomization is based on the fact that the *parity* function on l bits cannot be computed (on average) by constant-depth circuits of size 2^{l^ϵ} (where ϵ is a constant depending on the depth of the circuit; see e.g. Håstad 1987). On the other hand, proving P-uniform $BP \cdot AC^0 \subseteq P$ through a PRG would require exhibiting a function $f : \{0,1\}^l \to \{0,1\}$ in E that (for some constant $\epsilon > 0$) cannot be computed by constant-depth circuits of size $2^{\epsilon l}$. But it is consistent with the current state of knowledge that every function $f : \{0,1\}^l \to \{0,1\}$ in E is computable by circuits of size $2^{o(l)}$ and depth 3.

It should be noted that a result by Valiant (1977) states that if a function $f : \{0,1\}^l \to \{0,1\}$ (for some constant $\epsilon > 0$) cannot be computed by circuits of size $2^{\epsilon l}$ and depth 3, then it cannot be computed by circuits of linear size and logarithmic depth. Exhibiting a "natural function" that cannot be computed in the latter circuit class (i.e. linear size and logarithmic depth) is a long-standing open problem proposed in Valiant (1977).

1.2. Black-box PRG construction. As we explained in Section 1.1 in this work we study the complexity of constructing PRGs from hard functions. We now discuss what we mean by "constructing". An oracle procedure $G^f : \{0,1\}^u \to \{0,1\}^n$ is a *black-box PRG construction from worst-case hard functions* if for every function $f : \{0,1\}^l \to \{0,1\}$ and for every function $A : \{0,1\}^n \to \{0,1\}$, if A distinguishes the output of the PRG from random, i.e.

$$\left| \Pr_{x \in \{0,1\}^u}[A(G(x)) = 1] - \Pr_{x \in \{0,1\}^n}[A(x) = 1] \right| > 1/n,$$

then there is an oracle circuit C of size s that, given oracle access to A, computes f everywhere, i.e. $C^A(x) = f(x)$ for every x.

The idea is that if we start with a function f such that no circuit of size $s \cdot n$ can compute f everywhere (i.e. f is worst-case hard for size $s \cdot n$) then G is a PRG. This is because if a circuit A of size n distinguishes the output of G from random then C^A is a circuit of size $s \cdot n$ computing f everywhere, and this contradicts the hardness of f.

Note that in a black-box PRG construction we need the input length l of the hard function f to be $\Omega(\log n)$. Otherwise it can be shown that every function $f : \{0,1\}^l \to \{0,1\}$ can be computed by circuits of size $2^l \leq n$, and so our assumption that f is worst-case hard for size $s \cdot n$ is immediately false.

The PRG construction defined above is called *black-box* because it works for *every* function f and *every* potential distinguisher A, regardless of their complexity. While it is conceivable that non-black-box PRG constructions are more powerful than black-box ones, we note that all known PRG constructions are black-box, e.g. Blum & Micali (1984); Yao (1982); Impagliazzo & Wigderson

(1997); Klivans & van Melkebeek (1999); Sudan *et al.* (2001); Impagliazzo *et al.* (2000); Shaltiel & Umans (2001); Umans (2002), and all those in this paper. However, non-black-box PRG constructions are known to be more powerful than black-box ones in the *uniform* setting, i.e. when we only require that the output of the PRG be indistinguishable from random to *uniform* machines (as opposed to *nonuniform* circuits, required by our definition). We refer the reader to Trevisan & Vadhan (2002) for a discussion of this issue.

So far we have discussed black-box PRG constructions from *worst-case* hard functions. Black-box PRG constructions from *average-case* hard functions are analogous except that we only require that C^A computes f *on average* (as opposed to everywhere). By the same reasoning this suffices to ensure that G^f is a PRG whenever f is hard on average.

Black-box PRG constructions in AC^0. The main technical question addressed in this paper is: is there a black-box PRG construction in AC^0? (Recall AC^0 denotes the class of functions computable by polynomial-size constant-depth circuits.) As we explain below (Section 1.3), in this paper we exhibit both positive and negative results on this question. Our negative results apply regardless of the uniformity of AC^0, and our positive results hold even under a strict uniformity condition, namely *DLOGTIME*-uniformity, which we now discuss. Informally, a family of circuits is *DLOGTIME*-uniform if given indices to two gates one can decide their type and whether they are connected in linear time in the length of the indices (which is logarithmic time in the size of the circuit). There is a consensus that this is the "right" uniformity condition for AC^0, and the evidence for this is that *DLOGTIME*-uniform AC^0 has several different and elegant characterizations; see Barrington *et al.* (1990). A characterization that we will often use in this work is the following: *DLOGTIME*-uniform AC^0 is equivalent to $ATIME(O(1), \log n)$, where $ATIME(O(1), \log n)$ denotes alternating time $O(\log n)$ with $O(1)$ alternations (cf. Theorem 2.2). The class $ATIME(O(1), \log n)$, introduced by Sipser (1983), is the logarithmic analogue of the polynomial time hierarchy ($ATIME(O(1), n^{O(1)})$), and it is strictly contained in $SPACE(\log n)$.

1.3. Our results. Some of our results are summarized and compared to previous ones in Table 1. Our main finding is that there are black-box PRG constructions G^f from (mild) *average-case* hard functions such that G^f is computable in $ATIME(O(1), \log n)^f$. But there is no black-box PRG construction G^f from *worst-case* hard functions such that G^f is computable by small nonuniform oracle constant-depth circuits.

Table 1.1: A comparison of some of our results with previous ones.

(Recall $ATIME(O(1), \log n) = DLOGTIME$-uniform AC^0.)

Hardness amplification for functions : $\{0,1\}^l \to \{0,1\}$		**PRG construction from strong hardness** for PRG : $\{0,1\}^{O(\log n)} \to \{0,1\}^n$	**Derandomization given by PRG construction**
Previous results			
Worst-case hard $\Downarrow$ strongly hard		Complexity of PRG	Worst-case hardness assumption implies
Possible both in $TIME(2^{O(l)})$ [Theorems 3.3, 3.4, 3.5] and in $SPACE(O(l))$ [Theorem 3.7]		$TIME(n^{O(1)})$ [Theorem 3.2] $SPACE(O(\log n))$ [Theorem 3.7]	$BP \cdot P = P$ [Theorem 3.5] $BP \cdot L = L$ [Theorem 3.7]
Our results			
Worst-case hard $\Downarrow$ mildly hard	Mildly hard $\Downarrow$ strongly hard	Complexity of PRG	Mild average-case hardness assumption implies
Impossible in $ATIME(O(1), 2^{o(l)})$ if black-box [Corollary 7.5]	Possible in $ATIME(O(1), l)$ [Theorem 4.2]	$ATIME(O(1), \log n)$ [Theorem 4.1]	$BP \cdot ATIME(O(1), \log n)$ $\|$ $ATIME(O(1), \log n)$ [Theorems 4.3, 4.7]

The positive side. We show that there is a black-box PRG construction $G^f : \{0,1\}^{O(\log n)} \to \{0,1\}^n$ from mild *average-case* hard functions such that G^f is computable in $ATIME(O(1), \log n)^f$, where a function $f : \{0,1\}^l \to \{0,1\}$ is mildly hard on average if there is a constant $\epsilon > 0$ such that every circuit of size $2^{\epsilon l}$ fails to compute f on at least a $1/\text{poly}(l)$ fraction of inputs. In particular we deduce that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is mildly hard on average then there exists a PRG $G : \{0,1\}^{O(\log n)} \to \{0,1\}^n$ computable in $ATIME(O(1), \log n)$, and $BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n)$. (When we say that G is computable in $ATIME(O(1), \log n)$ we mean that given x and $i \leq n$ we can compute the i-th output bit of $G(x)$ in $ATIME(O(1), \log n)$.) The main new technical tool to achieve this result is a construction of combinatorial designs that is computable in $ATIME(O(1), \log n)$. We also show that it is possible to amplify mild average-case hardness up to strong average-case hardness within $ATIME(O(1), l)$, where l is the input size of the mild average-case hard function.

In addition, using results by Agrawal (2001), we show that our PRG construction can be based on the weaker hardness assumption that there exists a function that is hard for constant-depth circuits (whereas the discussion above refers to hardness against general circuits).

The negative side. We show that, for every positive constant $\delta < 1$, any black-box PRG construction $G^f : \{0,1\}^{\delta n} \to \{0,1\}^n$ from *worst-case* hard functions such that G^f is computable by an oracle circuit of constant depth and size g must essentially satisfy (we omit here some low order terms)

$$\log^{O(1)} g \geq 2^l / s, \tag{1.1}$$

where we start with a function $f : \{0,1\}^l \to \{0,1\}$ that is worst-case hard for circuits of size s. We stress that inequality (1.1) holds even when the input length of G is as big as $u = \delta n$ (recall our goal should be $u = O(\log n)$). To understand inequality (1.1) we must recall (from Section 1.2) that $l \geq \Omega(\log n)$. So when we start with a function that cannot be computed by circuits of size $s = 2^{\epsilon l}$, inequality (1.1) gives $\log^{O(1)} g \geq 2^{(1-\epsilon)l} \geq n^{\Omega(1)}$, and in particular the black-box PRG construction cannot be computed by a constant-depth circuit of size polynomial in n. We also show that this bound on g is tight.

On the other hand, if one insists on g polynomial in n then inequality (1.1) implies that we must start with a function $f : \{0,1\}^l \to \{0,1\}$ that is worst-case hard for circuits of size $s \geq 2^l / \log^{O(1)} n \geq 2^l / l^{O(1)}$. However, we show that this bound is so strong that worst-case and mild average-case hardness

are equivalent, in the sense that every function $f : \{0,1\}^l \to \{0,1\}$ that is worst-case hard for circuits of size $2^l/l^{O(1)}$ is actually mildly average-case hard for circuits of roughly the same size. In such a case one can construct a PRG using our construction from mild average-case hard functions.

We obtain analogous negative results for black-box constructions of average-case hard functions starting from worst-case hard ones. This gives the entry in the bottom left corner in Table 1. In particular we show that, starting from a worst-case hard function f, there is no black-box construction of a mild average-case hard function computable in the polynomial time hierarchy. Again, note that all known approaches are black-box, e.g. Babai *et al.* (1993); Sudan *et al.* (2001). However, similarly to PRG constructions (cf. Section 1.2), non-black-box hardness amplification is known to be more powerful than black-box hardness amplification in the *uniform* setting, i.e. when the function is hard for *uniform* machines (as opposed to *nonuniform* circuits). We refer the reader to Trevisan & Vadhan (2002) for a discussion of this issue.

It should be noted that a certain negative result for black-box worst-case hardness amplification already follows from our previous results. Namely, if there is a black-box worst-case hardness amplification then combining this with our black-box PRG construction from mild average-case hardness one gets a black-box PRG construction from worst-case hardness, and we have already given a negative result on this. However, we get a more general negative result through a direct proof.

Discussion. Since Impagliazzo & Wigderson (1997), PRG constructions from worst-case hard functions have been simplified and strengthened; see Sudan *et al.* (2001); Impagliazzo *et al.* (2000); Shaltiel & Umans (2001); Umans (2002). In particular, the latest constructions do not fall in the twofold paradigm of "hardness amplification + Nisan-Wigderson PRG", but directly transform worst-case hardness into randomness. However, our results suggest that the process of transforming worst-case hardness into randomness *is* twofold: black-box worst-case hardness amplification is harder than black-box PRG constructions from mild average-case hardness.

Our techniques. We now sketch the ideas behind our negative results. Our negative result for black-box PRG constructions employs the following ideas. First we use the fact, discovered by Trevisan (2001) (see also Trevisan & Vadhan 2002; Shaltiel 2002), that black-box PRG constructions give rise to "good" *extractors*, an object introduced in Nisan & Zuckerman (1996). Then we show that constant-depth circuits cannot compute "good" extractors. For this last

point we use the notion of *noise sensitivity*, which is a measure of how likely the output of a function is to change when the input is perturbed with random noise. On the one hand we show that extractors are very sensitive to noise, while on the other hand we know that constant-depth circuits are not (see Linial *et al.* 1993; Boppana 1997). This dichotomy establishes our negative result.

Our negative result for black-box worst-case hardness amplifications proceeds along similar lines: First, following Sudan *et al.* (2001) and Trevisan & Vadhan (2002), we show that black-box worst-case hardness amplifications give rise to "good" list-decodable codes. Then we show that "good" list-decodable codes are very sensitive to noise. Again, the negative result follows from the fact that constant-depth circuits are not very sensitive to noise.

1.4. Additional related work. There exist several other works addressing the complexity of PRGs (other than those we have already mentioned). We discuss known negative results first. Kharitonov *et al.* (1989) and Yu & Yung (1994) prove strong negative results about the ability of various automata and other space-restricted devices to compute PRGs. Linial *et al.* (1993) prove that small constant-depth circuits cannot compute pseudorandom functions (an object related to PRGs). Cryan & Miltersen (2001) consider the question of whether there are PRGs in NC^0. However, none of the above works study the complexity of constructing PRGs from hard functions. It should also be noted that space lower bounds for on-line computation of extractors and list-decodable codes are proved in Bar-Yossef *et al.* (2002). However, these lower bounds hold only in the on-line model of computation and therefore are incomparable with our results.

We now discuss known positive results on PRG constructions. There has been a series of works on the construction of PRGs from hard functions: Impagliazzo & Wigderson (1997); Klivans & van Melkebeek (1999); Sudan *et al.* (2001); Impagliazzo *et al.* (2000); Shaltiel & Umans (2001); Umans (2002). Previous to our paper, the most efficient construction is the one given in Klivans & van Melkebeek (1999) that constructs a PRG : $\{0,1\}^{O(\log n)} \to \{0,1\}^n$ computable in space $O(\log n)$. Note that in this paper we consider constructions in $ATIME(O(1), \log n)$, a class strictly contained in space $O(\log n)$. Other works give PRG constructions under the assumption that some *specific* problem is hard: Impagliazzo & Naor (1996) show how to construct PRGs based on the assumed intractability of the subset sum problem. In particular, they show how to construct a PRG : $\{0,1\}^{n-\Theta(\log n)} \to \{0,1\}^n$ in AC^0. Naor & Reingold (1997) give PRG constructions based on number-theoretic hardness assump-

tions. Their PRGs are computable by polynomial-size constant-depth circuits with MAJORITY gates (TC^0).

1.5. Organization. In Section 2 we give some preliminaries. In Section 3 we survey previous constructions of PRGs from hard functions. In Section 4 we describe our main results. In Section 5 we show how to construct a PRG computable in $ATIME(O(1), \log n)$ from a mild average-case hardness assumption. In Section 6 we prove our negative result for black-box PRG constructions from worst-case hardness assumptions. We also discuss in which sense our results are tight. In Section 7 we prove our negative result for black-box worst-case hardness amplification. In Section 8 we relax the hardness assumptions to the existence of functions hard for constant-depth circuits. In Section 9 we prove a lemma about noise sensitivity of constant-depth circuits which is used in our negative results. Finally, Section 10 discusses some open problems.

2. Preliminaries

Complexity. We denote by $ATIME(O(1), l)$ the class of functions computable in time $O(l)$ with constant number of alternations by a multitape Turing machine. (The Turing machine has a special address tape. On a given time step the machine has access to the bit of the input denoted by the contents of the address tape. This is to handle running times smaller than the input length.) We sometimes make use of the following result, usually attributed to Nepomnjaščiĭ (1970):

THEOREM 2.1 (Nepomnjaščiĭ 1970). *For any $\epsilon > 0$, if $f : \{0,1\}^l \to \{0,1\}$ is computable by an algorithm running in time* $\mathrm{poly}(l)$ *and using space $l^{1-\epsilon}$ then f is in $ATIME(O(1), l)$.*

A non-boolean function $f : \{0,1\}^n \to \{0,1\}^{n'}$ is in $ATIME(O(1), l)$ if for every i the i-th output bit $f(x)_i$ of f is in $ATIME(O(1), l)$. Note that if $g : \{0,1\}^{n'} \to \{0,1\}$ is in $ATIME(O(1), l)$ then the function $g \circ f$ is also in $ATIME(O(1), l)$.

We will occasionally consider the following complexity classes: We denote by $CTIME(O(1), l)$ the extension of $ATIME(O(1), l)$ where we also allow for *counting* quantifiers (see, e.g., Wagner 1986; Torán 1988). Along the same lines we denote by $A{\oplus}TIME(O(1), l)$ the extension of $ATIME(O(1), l)$ where we also allow for *parity* quantifiers.

We now define some complexity classes defined in terms of circuits. In this paper all gates in circuits have *unbounded fan-in*, with the only exception

of NOT gates that have fan-in one. The *size* of a circuit is the number of *edges* in the circuit. We denote by AC^0 the class of functions computable by polynomial-size constant-depth circuits of AND, OR and NOT gates. We denote by TC^0 the class of functions computable by polynomial-size constant-depth circuits of AND, OR, NOT and MAJORITY gates.

When we say *uniform* we always mean *DLOGTIME*-uniform (see, e.g., Barrington *et al.* 1990; Vollmer 1999). Other types of uniformity (e.g. P-uniformity) are always explicitly stated. Barrington *et al.* (1990) showed that uniform AC^0 is equivalent to $ATIME(O(1), \log n)$ (see also Vollmer 1999, Corollary 4.32). The same techniques give analogous equivalences for the classes we defined above:

THEOREM 2.2 (Barrington *et al.* 1990). *Let* $f : \{0,1\}^n \to \{0,1\}$.

- f *is in uniform* AC^0 *if and only if it is in* $ATIME(O(1), \log n)$.
- f *is in uniform* AC^0 *with PARITY gates if and only if it is in* $A{\oplus}TIME(O(1), \log n)$.
- f *is in uniform* TC^0 *if and only if it is in* $CTIME(O(1), \log n)$.

The following inclusions hold (see, e.g., Vollmer 1999, p. 161): uniform $AC^0 \subsetneq$ uniform AC^0 with PARITY gates $\subsetneq$ uniform $TC^0 \subsetneq L$. The first two inclusions also hold for non-uniform circuits.

For background on circuit complexity and uniformity the reader may consult the survey by Allender & Wagner (1990) and the excellent textbook by Vollmer (1999).

Finally, for a complexity class C, the class $BP \cdot C$ consists of the languages L for which there is $V \in C$ and a polynomial p such that $x \in L \Rightarrow \Pr_{y:\,|y|=p(|x|)}[V(x,y) = 1] \geq 2/3$ and $x \notin L \Rightarrow \Pr_{y:\,|y|=p(|x|)}[V(x,y) = 1] \leq 1/3$.

Hardness and pseudorandomness. We denote by U_l a random variable uniform on $\{0,1\}^l$.

We denote by CKT the class of circuits (with no depth restriction) made of AND, OR and NOT gates. We denote by $AC^0[d]$ the class of circuits of depth d made of AND, OR and NOT gates. We denote by $TC^0[d]$ the class of circuits of depth d made of AND, OR, NOT and MAJORITY gates.

Let $\mathcal{C}$ be a circuit class (e.g. $CKT, AC^0[17], \ldots$). A function $f : \{0,1\}^l \to \{0,1\}$ is (g,δ)-*hard* for $\mathcal{C}$ if for every circuit $C \in \mathcal{C}$ of size at most g we have

$$\Pr[C(U_l) = f(U_l)] < \delta.$$

Worst-case hardness corresponds to $\delta = 1$. Our threshold for *average-case* hardness is *mild* average-case hardness, corresponding to δ at most $1 - 1/l^c$ for some c. When we say that a function $f : \{0,1\}^l \to \{0,1\}$ is $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$-hard for $\mathcal{C}$ we mean that there is a constant $\epsilon > 0$ such that for every l the function $f : \{0,1\}^l \to \{0,1\}$ is $(2^{\epsilon l}, 1/2 + 2^{-\epsilon l})$-hard for $\mathcal{C}$.

A function $G : \{0,1\}^u \to \{0,1\}^n$ is an (n, ϵ)-*pseudorandom generator* (PRG) against $\mathcal{C}$ if for all $C \in \mathcal{C}$ of size at most n we have

$$\Big|\Pr[C(G(U_u)) = 1] - \Pr[C(U_n) = 1]\Big| \leq \epsilon.$$

An n-PRG is an $(n, 1/n)$-PRG. We refer to u as the *seed length* of G.

3. Previous PRG constructions from hard functions

In this section we survey previous PRG constructions from hard functions. This survey is only needed to understand our more efficient PRG constructions and our new derandomization results. The reader who is only interested in our negative results can safely skip this section.

We start with PRGs against *CKT* (recall *CKT* simply denotes standard circuits with no depth restrictions). Then we focus on PRGs against constant-depth circuits.

3.1. PRGs against *CKT*.

Nisan & Wigderson (1994) show how to construct PRGs from strong average-case hardness assumptions. We recall the definition of their PRG and state their result.

Definition 3.1 (Nisan & Wigderson 1994). An (m, l) *design of size* n *over a universe of size* u is a collection $(S_1, \ldots, S_n)$ of subsets of $\{1, \ldots, u\}$, each of size l, such that for any $1 \leq i < j \leq n$, the intersection $S_i \cap S_j$ has size at most m.

For a function $f : \{0,1\}^l \to \{0,1\}$, and a $(\log n, l)$ design of size n over a universe of size u, the Nisan–Wigderson PRG NW_f is defined as

$$NW_f : \{0,1\}^u \to \{0,1\}^n, \quad NW_f(x) = f(x|_{S_1}) \circ \cdots \circ f(x|_{S_n}),$$

where $x|_S$ is the string obtained from x by selecting the bits indexed by S.

Theorem 3.2 (Nisan & Wigderson 1994). *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$*-hard for CKT then there is an* n*-PRG against CKT with seed length* $O(\log n)$ *and computable in time* $\mathrm{poly}(n)$*, and in particular* $BP \cdot P = P$.

PROOF IDEA. The PRG is NW_f for a family of $(\log n, c \log n)$ designs of size n over a universe of size $d \log n$, for some constants c, d. Specifically, one needs such a family for every given c and some d. Nisan and Wigderson show that these families are computable in time poly(n), and that NW_f is an n-PRG. The "in particular" part is proved as follows: We run the algorithm we want to derandomize using all the possible outputs of the PRG in place of true random bits. Then we decide according to majority vote. □

An important point to keep in mind is that, although we are assuming that f is in E, the PRG is computable in time poly(n). This comes from the fact that f is evaluated on inputs of length $O(\log n)$.

A major line of research in the last ten years has focussed on relaxing the average-case hardness assumption in Theorem 3.2 to a worst-case one, that is, the existence of a function in E that is $(2^{\Omega(l)}, 1)$-hard for CKT. This was first achieved through the following *hardness amplifications within E*.

First, in Babai *et al.* (1993), random self-reducibility of EXP-complete problems is used to convert a worst-case hard function to one with mild average-case hardness.

THEOREM 3.3 (Babai *et al.* 1993). *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1)$*-hard for* CKT*, then there is a function* $f' \in E$ *that is* $(2^{\Omega(l)}, 1 - 1/\text{poly}(l))$*-hard for* CKT*.*

PROOF IDEA. f' is a small degree, multi-variate polynomial extension of f. For a suitable choice of parameters, the random self-reducibility of low-degree polynomials implies that f' has the required hardness. □

Then in Impagliazzo (1995) mild average-case hardness is amplified to constant hardness.

THEOREM 3.4 (Impagliazzo 1995). *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1 - 1/\text{poly}(l))$*-hard for* CKT*, then there is a function* $f' \in E$ *that is* $(2^{\Omega(l)}, 2/3)$*-hard for* CKT*.*

PROOF IDEA. Let $\tilde{f} : \{0,1\}^{O(l)} \to \{0,1\}$ be

$$\tilde{f}(a, r) := \langle f(x_1) \circ \cdots \circ f(x_l), r \rangle,$$

where $|a| = O(l)$, $|r| = l$ and $x_1, \ldots, x_l$ are pairwise independent samples in $\{0,1\}^l$ obtained from seed a, and $\langle \cdot, \cdot \rangle$ denotes inner product mod 2. In other words, $\tilde{f}$ is the inner product of the random string r with l evaluations of f on pairwise independent inputs $x_1, \ldots, x_l$.

It is shown in Impagliazzo (1995) that, if we apply this transformation a constant number of times to f, then we obtain a function with constant hardness. □

Finally, in Impagliazzo & Wigderson (1997) it is shown how to amplify constant hardness to the kind of hardness required in Theorem 3.2.

THEOREM 3.5 (Impagliazzo & Wigderson 1997). *If there exists a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 2/3)$*-hard for CKT, then there is a function* $f' \in E$ *which is* $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$*-hard for CKT.*

PROOF IDEA. Consider $f' : \{0,1\}^{O(l)} \to \{0,1\}$ defined as

$$f'(x, r, v_1, p) := \langle f(x|_{S_1} \oplus v_1) \circ \cdots \circ f(x|_{S_l} \oplus v_l), r \rangle,$$

where $\oplus$ denotes bitwise XOR, $(S_1, \ldots, S_l)$ is an (l, cl) design of size l over a universe of size dl, for some c, d as in Theorem 3.2, and $(v_1, \ldots, v_l)$ is a walk in an expander graph over $\{0,1\}^{O(l)}$ with constant degree and bounded second largest eigenvalue. This walk is obtained by starting at v_1 and walking according to p. □

Combining all these results, one gets:

THEOREM 3.6 (Impagliazzo & Wigderson 1997). *If there exists a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1)$*-hard for CKT then there is an* n*-PRG against CKT with seed length* $O(\log n)$ *and computable in time* $\mathrm{poly}(n)$*, and in particular* $BP \cdot P = P$*.*

After Impagliazzo & Wigderson (1997) PRGs constructions from worst-case hard functions have been simplified and strengthened (see Sudan *et al.* 2001; Impagliazzo *et al.* 2000; Shaltiel & Umans 2001; Umans 2002). In particular, last constructions do not fall in the twofold paradigm "hardness amplification + NW PRG", but directly transform worst-case hardness into pseudorandomness. However, our results show that transforming worst-case hardness into pseudorandomness is a substantially harder task than transforming mild average-case hardness into pseudorandomness. Therefore we use the earlier constructions that allow us to investigate the fine structure of hardness amplification.

Klivans & van Melkebeek (1999) prove a space-bounded analogue of Theorem 3.6. They show how to amplify hardness within linear space, then they

give a more efficient implementation of the NW PRG. We summarize their final result in the following theorem.

THEOREM 3.7 (Klivans & van Melkebeek 1999). *If there exists a function* $f : \{0,1\}^l \to \{0,1\}$ *in* $SPACE(l)$ *that is* $(2^{\Omega(l)}, 1)$*-hard for* CKT*, then there is an* n*-PRG against* CKT *with seed length* $O(\log n)$ *and computable in* $SPACE(\log n)$*, and in particular* $BP \cdot L = L$.

3.2. PRGs against constant-depth circuits. A natural question, addressed by Agrawal (2001), is: What are the hardness assumptions needed for constructing PRGs against more restricted classes of circuits? As pointed out in Agrawal (2001), in all the proofs of correctness of the above constructions the depth only increases by a constant amount, *provided that the circuits have MAJORITY gates*. This gives the following result (recall that $TC^0[d]$ denotes the class of circuits of depth d made of AND, OR, NOT and MAJORITY gates):

THEOREM 3.8 (Agrawal 2001). *There is a constant* c *such that if there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1)$*-hard for* $TC^0[d+c]$*, then there is an* n*-PRG against* $TC^0[d]$ *with seed length* $O(\log n)$ *and computable in time* $\mathrm{poly}(n)$*. In particular, if for every* d *there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* E *that is* $(2^{\Omega(l)}, 1)$*-hard for* $TC^0[d]$*, then* $BP \cdot TC^0 = TC^0$ *subject to P-uniformity.*

Now we focus on PRGs against constant-depth circuits *without* MAJORITY gates. Note that Theorem 3.8 does *not* immediately translate to constant-depth circuits because it is known that small constant-depth circuits cannot compute majority (see Furst *et al.* 1984; Håstad 1987).

Nisan (1991) constructs an *unconditional* PRG against constant-depth circuits, using the results by Håstad (1987) on the average-case hardness of the function $\mathrm{parity}(x_1 \dots x_l) := (\sum_i x_i) \bmod 2$.

Recall that $AC^0[d]$ denotes the class of circuits of depth d made of AND, OR and NOT gates.

THEOREM 3.9 (Nisan 1991). *For every* d *there is an* n*-PRG against* $AC^0[d]$ *with seed length* $\log^{O(1)} n$*, and computable in time* $\mathrm{poly}(n)$.

PROOF IDEA. The PRG is NW_{parity} for a family of $(\log n, \log^c n)$ designs of size n over a universe of size $\log^e n$, for some constants c, e. Specifically, one needs such a family for every given c and some e. Nisan shows how to construct such families in time $\mathrm{poly}(n)$. □

Although Nisan's PRG does not rely on any complexity assumption, it has polylogarithmic seed length, and therefore it cannot be used directly to obtain $BP \cdot AC^0 = AC^0$ subject to P-uniformity.

One can try to construct, under the assumption that some function is hard for constant-depth circuits, a PRG with logarithmic seed length against constant-depth circuits, following the construction in Section 3.1. The difficulty in this approach is that the proof of correctness of the construction in Theorem 3.5 (and other approaches like Sudan *et al.* 2001) does not carry through in small constant-depth circuits. This problem is discussed and then solved by Agrawal (2001):

THEOREM 3.10 (Agrawal 2001). *There is a constant c such that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in E that is $(2^{\Omega(l)}, 1)$-hard for $AC^0[c \cdot d]$, then there is an $(n, 1/\log^{O(1)} n)$-PRG against $AC^0[d]$ with seed length $O(\log n)$ and computable in time* $\mathrm{poly}(n)$.

PROOF IDEA. Agrawal's PRG is obtained by combining a conditional PRG G with Nisan's unconditional PRG from Theorem 3.9. Since Nisan's PRG has polylogarithmic seed length, we can get a combined PRG with logarithmic seed length if G has only polynomial stretch (i.e. $G : \{0,1\}^l \to \{0,1\}^{l^{O(1)}}$). Now, to construct such a G we can use exactly the same construction in Section 3.1: Agrawal shows that, since the stretch of G is only polynomial, all the proofs of correctness carry through in small constant-depth circuits. □

Note that Theorem 3.10 gives an $(n, 1/\log^{O(1)} n)$-PRG instead of an n-PRG. However, this is sufficient for derandomization purposes.

As we already mentioned, a PRG with logarithmic seed length allows us to derandomize an algorithm *provided that we can compute majority*. While AC^0 cannot compute majority, Klivans (2001) notices that one can use a construction by Ajtai (1993) to *approximately* compute majority in AC^0, which is enough for the derandomization to go through. This gives the following corollary.

COROLLARY 3.11 (Agrawal 2001; Klivans 2001). *If for every d there is a function $f : \{0,1\}^l \to \{0,1\}$ in E that is $(2^{\Omega(l)}, 1)$-hard for $AC^0[d]$, then $BP \cdot AC^0 = AC^0$ subject to P-uniformity.*

The P-uniformity in Theorem 3.8 and Corollary 3.11 can be lowered to L-uniformity (i.e. circuit families described by a Turing machine running in logarithmic space) using techniques in Klivans & van Melkebeek (1999); Klivans (2001).

4. Our results

In this section we describe our results. The main ones are summarized and compared to previous results in Table 1. Our main finding is that constructions from *worst-case* hard functions have higher complexity than constructions from *mildly hard on average* functions.

Improving on the complexity of the design construction in the NW PRG, we obtain the following:

THEOREM 4.1. *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* $ATIME(O(1), l)$ *that is* $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$*-hard for CKT then there is an* n*-PRG against CKT with seed length* $O(\log n)$ *and computable in* $ATIME(O(1), \log n)$*, and*

$$BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n).$$

In analogy with the results discussed in Section 3.1, to relax the average-case hardness assumption in Theorem 4.1 we study hardness amplification in the linear exponential analogue of $ATIME(O(1), \log n)$, that is, linear alternating time with $O(1)$ alternations.

Combining our design construction with a result by Ajtai (1993) on the complexity of certain expander graphs we show that average-case hardness can be amplified from mild to strong within linear alternating time.

THEOREM 4.2. *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* $ATIME(O(1), l)$ *that is* $(2^{\Omega(l)}, 1 - 1/\mathrm{poly}(l))$*-hard for CKT, then there is a function* $f' \in ATIME(O(1), l)$ *which is* $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$*-hard for CKT.*

Combining Theorems 4.1 and 4.2 we can construct a PRG computable in $ATIME(O(1), \log n)$ from a function of mild average-case hardness.

THEOREM 4.3. *If there is a function* $f : \{0,1\}^l \to \{0,1\}$ *in* $ATIME(O(1), l)$ *that is* $(2^{\Omega(l)}, 1 - 1/\mathrm{poly}(l))$*-hard for CKT then there is an* n*-PRG against CKT with seed length* $O(\log n)$ *and computable in* $ATIME(O(1), \log n)$*, and*

$$BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n).$$

Theorems 4.1, 4.2 and 4.3 are proved in Section 5.

On the negative side, we show a negative result for black-box PRG constructions starting from worst-case hard functions (cf. Section 1.2).

THEOREM 4.4 (informal). *Starting from a worst-case hard function, for every positive constant $\delta < 1$ there is no black-box construction of a PRG $: \{0,1\}^{\delta n} \to \{0,1\}^n$ computable by a constant-depth circuit of size $2^{n^{o(1)}}$.*

It is interesting to note that the bottleneck is indeed worst-case hardness amplification:

THEOREM 4.5 (informal). *Starting from a worst-case hard function $f : \{0,1\}^l \to \{0,1\}$, there is no black-box construction of a mildly hard on average function $f' : \{0,1\}^{l'} \to \{0,1\}$ computable in $ATIME(O(1), 2^{o(l)})$.*

In particular, there is no black-box worst-case hardness amplification within the polynomial time hierarchy.

Theorems 4.4 and 4.5 are tight in the following sense: The only settings of parameters which are not ruled out correspond either to computational resources that allow for the worst-case hardness amplification in Theorem 3.3, which combined with Theorem 4.3 gives a PRG construction from worst-case hard functions, or else they correspond to hardness assumptions so strong that worst-case hardness and mild average-case hardness *collapse*, in which case no worst-case hardness amplification is needed, and to get a PRG one can apply Theorem 4.3 directly.

Theorems 4.4 and 4.5 are obtained by showing that constant-depth circuits cannot compute good extractors and list-decodable codes. These theorems are formally stated and proved in Sections 6 and 7, respectively.

We note that worst-case to average-case hardness amplification becomes feasible if one allows PARITY gates. This allows us to construct a PRG computable in $A \oplus TIME(O(1), \log n)$ from a worst-case hardness assumption.

THEOREM 4.6. *If there is a function $f : \{0,1\}^l \to \{0,1\}$ in $A \oplus TIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for CKT then there is an n-PRG against CKT with seed length $O(\log n)$ and computable in $A \oplus TIME(O(1), \log n)$, and*

$$BP \cdot A \oplus TIME(O(1), \log n) = A \oplus TIME(O(1), \log n).$$

Theorem 4.6 is proved in Section 6.1.

What is not completely satisfactory in the above derandomization results is that our hardness assumptions are qualitatively stronger than the corresponding derandomizations. For example, consider Theorem 4.3. The nonuniform analogue of $ATIME(O(1), \log n)$ is AC^0, so one wants the same conclusions under the weaker assumption of a hard function for small constant-depth circuits. Using Agrawal's construction presented in Theorem 3.10, we obtain the following theorem.

THEOREM 4.7. *There exists a constant c such that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/l^b)$-hard for $AC^0[c \cdot \max(b, d)]$, then there is an $(n, 1/\log^{O(1)} n)$-PRG against $AC^0[d]$ with logarithmic seed length and computable in $ATIME(O(1), \log n)$.*

In particular, if there is a constant b such that for every d there is a function in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/l^b)$-hard for $AC^0[d]$, then

$$BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n).$$

Finally, we point out the following derandomization of $BP \cdot CTIME(O(1), \log n)$ under worst-case hardness assumptions for small constant-depth circuits with MAJORITY gates.

THEOREM 4.8. *There exists a constant c such that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in $CTIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for $TC^0[c + d]$, then there is an n-PRG against $TC^0[d]$ with seed length $O(\log n)$ and computable in $CTIME(O(1), \log n)$.*

In particular, if for every d there is a function $f : \{0,1\}^l \to \{0,1\}$ in $CTIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for $TC^0[d]$, then

$$BP \cdot CTIME(O(1), \log n) = CTIME(O(1), \log n).$$

Theorems 4.7 and 4.8 are proved in Section 8.

Note that our results could be equivalently stated in terms of *DLOGTIME*-uniform circuit classes because of Theorem 2.2.

5. Average-case hardness vs. randomness

In this section we show how to construct an n-PRG $G : \{0,1\}^{O(\log n)} \to \{0,1\}^n$ against CKT computable in $ATIME(O(1), \log n)$ starting from a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/\text{poly}(l))$-hard for CKT. In particular, we prove Theorems 4.1, 4.2 and 4.3.

Here our main new technical contribution is the construction of the family of designs to be used in the NW PRG, which we now discuss.

First we show how to compute pairwise independent samples over $\{0,1\}^l$ in $ATIME(O(1), l)$. A matrix T with entries in $\{0,1\}$ is *Toeplitz* if it is constant on diagonals. It is well known (cf. Goldreich 1997) that if we choose a random $l \times l$ Toeplitz matrix T and a random vector $U \in \{0,1\}^l$, then the 2^l random variables $\{Tx + U : x \in \{0,1\}^l\}$ are pairwise independent over $\{0,1\}^l$.

Clearly, an $l \times l$ Toeplitz matrix T is uniquely determined by the string $t \in \{0,1\}^{2l-1}$ of its values on the first row and on the first column. The

following lemma states that we can compute pairwise independent samples over $\{0,1\}^l$ in $ATIME(O(1), l)$.

LEMMA 5.1. *There exists a machine $A(t, x, u)$ which computes $Tx + u$ in $ATIME(O(1), l)$ for $|x| = |u| = l$, $|t| = 2l - 1$ and T the Toeplitz matrix determined by t.*

PROOF. Recall that what we need to show is that, given t, x, u and i, we can compute the i-th bit of $Tx + u$ in $ATIME(O(1), l)$. We actually show that it can be computed in deterministic time $O(l)$. It is easy to see that the i-th bit of $Tx + u$ is

$$\langle t_i \dots t_{i+l-1}, x \rangle + u_i.$$

Note that the inner product is over l bits, and therefore can be computed in time $O(l)$. □

We now show our design construction.

LEMMA 5.2. *For every constant c there is a constant d such that there is a family $\{D_n\}$ of $(\log n, c \log n)$ designs of size n over a universe of size $d \log n$ with the following property: There is a machine in $ATIME(O(1), \log n)$ which, given n and $k \leq n$, computes the characteristic vector of the k-th set in D_n.*

PROOF. Let $l := \log n$. First we show the existence with a probabilistic argument. Then we show how to derandomize the argument. Finally, we show how the derandomization is implementable in $ATIME(O(1), l)$.

Existence: We view the universe as cl blocks of b elements each, i.e. let $d := cb$, for some b we specify later.

Let us choose $S_1, \dots, S_n$ at random from the sets which have exactly one element in each block. Notice the size of these sets is cl, as required.

For every $i \neq j$, by a union bound,

$$\Pr[|S_i \cap S_j| \geq l] \leq \binom{cl}{l} \left(\frac{1}{b}\right)^l \leq \left(\frac{ecl}{l}\right)^l \left(\frac{1}{b}\right)^l \leq \left(\frac{ec}{b}\right)^l.$$

If we take $b := 4ec$, the latter equals $1/n^2$. So, by a union bound,

$$\Pr[\exists i < j : |S_i \cap S_j| \geq l] \leq \sum_{i<j} \Pr[|S_i \cap S_j| \geq l] \leq \binom{n}{2} \frac{1}{n^2} < \frac{1}{2} < 1.$$

Therefore such designs exist.

Derandomization: Note that the analysis goes through even if the sets are just pairwise independent. We use this below to show that we can compute in $ATIME(O(1), \log n)$ the characteristic vectors of the sets in D_n.

$ATIME(O(1), l)$: Each string $s \in \{0,1\}^{(\log b)\cdot cl}$ represents a set S with one element in each block in the following natural way: View s as cl blocks of $\log b$ bits each; the i-th block of s is an index to an element in the i-th block of b elements in our universe. We can easily construct a machine T running in time $O(l)$ computing this transformation, i.e. $T(s) \in \{0,1\}^{b\cdot cl}$ is the characteristic vector of the set with one element in each block which $s \in \{0,1\}^{(\log b)\cdot cl}$ represents.

Let $A \in ATIME(O(1), l)$ be the machine given by Lemma 5.1 that, given a and i, computes the i-th pairwise independent sample over $\{0,1\}^{(\log b)\cdot cl}$ according to a. Note we can check in $ATIME(O(1), l)$ if the samples corresponding to some a form a design:

$$\forall i \neq j \in \{0,1\}^l \qquad \Big|T(A(a,i)) \cap T(A(a,j))\Big| \leq l.$$

We already know that A and T are in $ATIME(O(1), l)$. Note that computing the intersection size is feasible since we are dealing with strings of length $O(l)$.

To put our hands on some particular design, we can existentially guess a string a^* and universally verify that it is the lexicographically first string whose samples correspond to a design. The characteristic vector of the k-th set in D_n is then $T(A(a^*, k))$. □

REMARK 5.3. Our construction of designs is a mix of the constructions in Raz *et al.* (1999) and Klivans & van Melkebeek (1999): We choose the sets with one element in each block, as in Raz *et al.* (1999), and we derandomize the argument through pairwise independence, as in Klivans & van Melkebeek (1999). Neither the construction in Raz *et al.* (1999) nor the one in Klivans & van Melkebeek (1999) seems to be easily implementable in $ATIME(O(1), \log n)$: The construction in Raz *et al.* (1999) seems to require polynomial space in the size of the design because of the method of conditional probabilities. The construction in Klivans & van Melkebeek (1999) needs to associate to a number $x \leq \binom{c\log n}{d\log n}$ the x-th subset of $\{1, \ldots, c\log n\}$ of size $d\log n$. This latter operation can be easily computed in $SPACE(\log n)$, going through all the subsets, but we do not know if it can be computed in $ATIME(O(1), \log n)$. Moreover, the analysis of our construction is simpler than the analysis in Klivans & van Melkebeek (1999).

Other constructions of designs are obtained by Hartman & Raz (2003) and Luca Trevisan & Hoeteck Wee (personal communication, Sept. 2002). These constructions either do not achieve the parameters of interest here, or, to the best of our knowledge, are not known to be computable in $ATIME(O(1), \log n)$.

Plugging this design construction into the NW PRG we obtain the following:

THEOREM 4.1, RESTATED. *If there exists a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$-hard for CKT then there exists an n-PRG against CKT with seed length $O(\log n)$ and computable in $ATIME(O(1), \log n)$, and $BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n)$.*

PROOF. The PRG is NW_f, with the design construction from Lemma 5.2. The correctness of this construction has been proved in Nisan & Wigderson (1994). The fact that $NW_f \in ATIME(O(1), \log n)$ follows from Lemma 5.2 and the fact that $f \in ATIME(O(1), \log n)$. In analogy with Corollary 3.11, to obtain $BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n)$ we use the construction for approximate majority by Ajtai (1993). (In Ajtai 1993 the construction is given in terms of first-order definability, but this coincides with $ATIME(O(1), \log n)$, see Barrington *et al.* 1990.) □

Along the lines of the previous results discussed in Section 3.1, we now want to relax the strong average-case hardness assumption. Therefore we now prove some results about hardness amplification within $ATIME(O(1), l)$. These hardness amplifications will allow us to start from a function with mild average-case hardness. See Sections 7 and 6.1 for a discussion of worst-case hardness assumptions.

LEMMA 5.4. *If there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/\text{poly}(l))$-hard for CKT, then there is a function $f' \in ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 2/3)$-hard for CKT.*

PROOF. We use the construction in Theorem 3.4. The correctness of this construction has already been proved in Impagliazzo (1995), so it is only left to see that $f' \in ATIME(O(1), l)$. This follows from the construction of a pairwise independent sample space given in Lemma 5.1. □

Combining our design construction with a result by Ajtai (1993) on the complexity of certain expander graphs we can amplify from constant hardness to exponential hardness.

LEMMA 5.5. *If there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 2/3)$-hard for CKT, then there is a function $f' \in ATIME(O(1), l)$ which is $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$-hard for CKT.*

PROOF. We use the construction in Theorem 3.5. The correctness of this construction has already been proved in Impagliazzo & Wigderson (1997), so it is only left to see that $f' \in ATIME(O(1), l)$. Lemma 5.2 shows how to compute the required designs in $ATIME(O(1), l)$.

It remains to show how to compute walks on expanders in $ATIME(O(1), l)$. This problem, for the parameters of interest here, has already been solved by Ajtai (1993), using the expander construction by Lubotzky *et al.* (1988).

LEMMA 5.6 (Ajtai 1993). *There is a constant α, $0 < \alpha < 1$, such that for every prime n congruent to 1 modulo 4 there is a 6-regular graph G_n on n vertices with second largest eigenvalue at most α. Moreover, there is a machine in $ATIME(O(1), \log n)$ which, given a prime n congruent to 1 modulo 4, $x \in G_n$ and p, with $|p| \leq O(\log n)$, computes the node in G_n reached starting from x and following the path specified by p.*

□

Combining the above two hardness amplifications we get the following theorem.

THEOREM 4.2, RESTATED. *If there exists a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/\mathrm{poly}(l))$-hard for CKT, then there is a function $f' \in ATIME(O(1), l)$ which is $(2^{\Omega(l)}, 1/2 + 2^{-\Omega(l)})$-hard for CKT.*

This allows us to construct a PRG computable in $ATIME(O(1), \log n)$ from a function of mild average-case hardness.

THEOREM 4.3, RESTATED. *If there exists a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/\mathrm{poly}(l))$-hard for CKT then there exists an n-PRG against CKT with seed length $O(\log n)$ and computable in $ATIME(O(1), \log n)$, and $BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n)$.*

6. PRGs from worst-case hardness

In this section we discuss PRG constructions from worst-case hardness assumptions, and in particular we prove a formal version of Theorem 4.4, establishing a negative result for black-box PRG constructions from worst-case hardness

assumptions. In Section 6.1 we discuss the tightness of our negative result and we also prove Theorem 4.6.

To show our negative result for black-box PRG constructions we proceed in two steps: First we use the fact, discovered by Trevisan (2001) (see also Trevisan & Vadhan 2002; Shaltiel 2002), that black-box PRG constructions give rise to "good" *extractors*. Then we show that "good" extractors are not computable by small constant-depth circuits. To explain the intuition behind this last step we need the notion of *noise sensitivity*. Roughly speaking, the noise sensitivity of a function is a measure of how likely the output of the function is to change when one perturbs the input with random noise. We show that "good" extractors are very sensitive to noise. Since constant-depth circuits are not (see Section 9), we obtain our negative result.

We now proceed to turn the above sketch into a formal proof.

DEFINITION 6.1. An oracle algorithm $G^f : \{0,1\}^u \to \{0,1\}^n$ is an (l, s, ϵ)-*black-box PRG construction* if for every $f : \{0,1\}^l \to \{0,1\}$ and for every $A : \{0,1\}^n \to \{0,1\}$ such that

$$\left|\Pr[A(G^f(U_u)) = 1] - \Pr[A(U_n) = 1]\right| \geq \epsilon$$

there is an oracle circuit C of size at most s such that $C^A(x) = f(x)$ for every x.

Note that in the above definition we did not specify the type of the circuit C (e.g. $CKT, AC^0[17], \ldots$) because it does not play a role in this section. Also note that if $G^f : \{0,1\}^u \to \{0,1\}^n$ is an (l, s, ϵ)-black-box PRG construction then for every function $f : \{0,1\}^l \to \{0,1\}$, if f is $(ns, 1)$-hard then G^f is an n-PRG.

We note that in this notation the PRG construction in Theorem 3.6 is an $(O(\log n), n^\gamma, 1/n)$-black-box PRG construction for some $0 < \gamma < 1$ (see, e.g., Trevisan 2001). This PRG construction also gives $u = O(\log n)$, which is what one needs for high-end derandomization. However, our negative result applies regardless of this.

We now define extractors. The *min-entropy* of a random variable X is defined as $H_\infty(X) := \min_x \log(1/\Pr[X = x])$.

DEFINITION 6.2 (Nisan & Zuckerman 1996). $E : \{0,1\}^h \times \{0,1\}^u \to \{0,1\}^n$ is a (k, ϵ) *extractor* if for every random variable X of min-entropy at least k, and for every $T \subseteq \{0,1\}^n$,

$$\left|\Pr[E(X, U_u) \in T] - \Pr[U_n \in T]\right| \leq \epsilon.$$

We call $T \subseteq \{0,1\}^n$ a *test* and $y \in \{0,1\}^u$ a *seed*.

Trevisan (2001) shows that black-box PRG constructions are extractors (see also Trevisan & Vadhan 2002; Shaltiel 2002). For completeness, we now state and prove this result.

THEOREM 6.3 (Trevisan 2001). *Let $G^f : \{0,1\}^u \to \{0,1\}^n$ be an (l,s,ϵ)-black-box PRG construction. Then $E : \{0,1\}^{2^l} \times \{0,1\}^u \to \{0,1\}^n$ defined as $E(x,y) := G^x(y)$ is an $(O(s \log s) + \log(1/\epsilon), 2\epsilon)$ extractor.*

PROOF. Let X be a random variable and $T \subseteq \{0,1\}^n$ such that

$$\left| \Pr_{X,U_u}[E(X,U_u) \in T] - \Pr_{U_n}[U_n \in T] \right| > 2\epsilon.$$

Then, using the triangle inequality we get

$$\Pr_X \left[\left| \Pr_{U_u}[E(X,U_u) \in T] - \Pr_{U_n}[U_n \in T] \right| \geq \epsilon \right] > \epsilon.$$

Since for every x such that $|\Pr_{U_u}[E(x,U_u) \in T] - \Pr_{U_n}[U_n \in T]| \geq \epsilon$ there must exist an oracle circuit of size at most s such that $C^T = x$, the number of such x is bounded by the number of oracle circuits of size at most s. There are at most $2^{O(s \log s)}$ such circuits. Therefore X lands in a set of size at most $2^{O(s \log s)}$ with probability greater than ϵ, and so $H_\infty(X) < O(s \log s) + \log(1/\epsilon)$. □

The following theorem states that, for every positive constant $\delta < 1$, constant-depth circuits cannot compute good extractors for min-entropy $k \leq n^\delta$ and seed length $u \leq \delta n$. In a subsequent work ("On constructing parallel pseudorandom generators from one-way functions", 2004, Electronic Colloquim on Computational Complexity, Technical Report 04-074) we show that small constant-depth circuits *can* compute extractors when the seed length u is greater than n.

THEOREM 6.4. *Fix positive constants $\epsilon < 1, \delta < 1$. Let $E : \{0,1\}^h \times \{0,1\}^u \to \{0,1\}^n$ be a (k,ϵ) extractor, with $u \leq \delta n$, and let E be computable by a circuit of size g and depth d. Then*

$$\log^{d-1} g \geq \Omega\left(\frac{h}{k}\right).$$

Before proving Theorem 6.4 note that, in combination with Theorem 6.3, it yields the following negative result for black-box PRG constructions. In the following corollary the reader is invited to ignore the term $O(l)$ that will be negligible for most settings of parameters.

COROLLARY 6.5 (formal version of Theorem 4.4). *Let $\epsilon := 1/4$ and fix a positive a constant $\delta < 1$. Let $G : \{0,1\}^{\delta n} \to \{0,1\}^n$ be an (l, s, ϵ)-black-box PRG construction, and let G^f be computable by an oracle circuit of size g and depth d. Then*

$$\log g \geq \left(\frac{2^l}{s \log s}\right)^{1/O(d)} - O(l).$$

In particular, for any fixed constants $\gamma < 1, c \geq 1$, there is no $(c \log n, n^\gamma, \epsilon)$-black-box PRG construction computable by a constant-depth circuit of size $2^{n^{o(1)}}$.

PROOF. A circuit C^f of size g and depth d with oracle access to $f : \{0,1\}^l \to \{0,1\}$ can be transformed into another equivalent circuit C' of size $g \cdot \text{poly}(l) \cdot 2^l$ and depth $O(d)$ such that $C^f(x) = C'(x, f)$ for every x, f. Note that C' does not have an oracle but instead takes the truth-table of f as part of the input. This transformation is simply obtained by replacing every oracle gate of C with a constant-depth circuit of size $\text{poly}(l) \cdot 2^l$ that answers the query by looking at the truth-table of f.

The result then follows from Theorems 6.3 and 6.4. □

To prove Theorem 6.4 we make use of the following fact about low noise sensitivity of constant-depth circuits, which we deduce by combining results in Kahn *et al.* (1988); Boppana (1997); O'Donnell (2002). We denote bitwise XOR by $\oplus$.

LEMMA 6.6. *Let $C : \{0,1\}^n \to \{0,1\}$ be a circuit of size g and depth d. Let $X \in \{0,1\}^n$ be a random input and let $\Psi \in \{0,1\}^n$ be a random noise vector where each bit is 1 independently with probability $\delta < 1/2$. Then*

$$\Pr_{X,\Psi}[C(X) \neq C(X \oplus \Psi)] \leq O(\delta \log^{d-1} g).$$

The proof of Lemma 6.6 requires a detour into Fourier analysis and therefore we defer it to Section 9.

The following easy lemma states that a random vector of noise has high min-entropy.

LEMMA 6.7. *Fix any $x \in \{0,1\}^h$, and let $\Psi \in \{0,1\}^n$ be a random noise vector where each bit is 1 independently with probability $k/h \leq 1/2$. Then $H_\infty(x \oplus \Psi) \geq \Omega(k)$.*

PROOF. We have

$$H_\infty(x \oplus \Psi) \geq \log(1/\Pr[x \oplus \Psi = x]) = \log(1/(1-k/h)^h) \geq \Omega(k),$$

where the first inequality holds because $k/h \leq 1/2$. □

PROOF OF THEOREM 6.4. For $z, z' \in \{0,1\}^n$ let $\Delta(z, z')$ denote the *relative* Hamming distance, i.e. $\Pr_i[z_i \neq z'_i]$. Let $E(x,y)_i$ denote the i-th bit of $E(x,y)$. Let $\Psi \in \{0,1\}^h$ be a random noise vector where each bit is 1 independently with probability $O(k/h)$ so that for every *fixed* $x \in \{0,1\}^h$ we have $H_\infty(x \oplus \Psi) \geq k$ by Lemma 6.7. Let X be chosen at random in $\{0,1\}^h$.

The main ideas are the following: For every seed y, we expect $\Delta(E(X,y), E(X \oplus \Psi, y))$ to be "small" by the low average sensitivity of constant-depth circuits (Lemma 6.6). We can fix $X = x$ maintaining this property. Now we can tell whether a sample z comes from $E(x \oplus \Psi, U_u)$, rather than being truly random, checking whether there is a seed y such that $\Delta(E(x,y), z)$ is "small". This contradicts the fact that E is an extractor since for every fixed $x \in \{0,1\}^h$ the distribution $x \oplus \Psi$ has high min-entropy by Lemma 6.7.

Fix a seed y and a position $i \in \{1, \ldots, n\}$. By Lemma 6.6

$$\Pr_{X,\Psi}[E(X,y)_i \neq E(X \oplus \Psi, y)_i] \leq \frac{O(k \log^{d-1} g)}{h}.$$

By linearity of expectation,

$$\mathop{\mathrm{E}}_{X,\Psi,Y}[\Delta(E(X,Y), E(X \oplus \Psi, Y))] \leq \frac{O(k \log^{d-1} g)}{h}.$$

By averaging there must exist a fixed x such that

$$\mathop{\mathrm{E}}_{\Psi,Y}[\Delta(E(x,Y), E(x \oplus \Psi, Y))] \leq \frac{O(k \log^{d-1} g)}{h}.$$

Pick a small constant $\xi > 0$. By the Markov inequality,

$$\Pr_{\Psi,Y}[\Delta(E(x,Y), E(x \oplus \Psi, Y)) \geq \xi] \leq \frac{O(k \log^{d-1} g)}{h \cdot \xi}. \tag{6.8}$$

We are now ready to define the test T that will distinguish the output of the extractor from U_n:

$$T := \{z \in \{0,1\}^n : \exists y \in \{0,1\}^u \ \Delta(E(x,y), z) \leq \xi\}.$$

By (6.8) we have

$$\Pr_{\Psi,Y}[E(x \oplus \Psi, Y) \in T] \geq 1 - \frac{O(k \log^{d-1} g)}{h \cdot \xi}. \tag{6.9}$$

We now show that a truly random sample will pass the test with very low probability. Fix a seed y. Then

$$\Pr_{U_n}[\Delta(E(x,y), U_n) \leq \xi] \leq \frac{V_n(\xi)}{2^n} \leq 2^{(H(\xi)-1)n},$$

where $V_n(\xi)$ is the size of a Hamming ball in $\{0,1\}^n$ of radius ξn, and $H(x) = -x \log x - (1-x)\log(1-x)$ is the binary entropy function.

Since there are 2^u seeds, using a union bound we obtain

$$\Pr_{U_n}[U_n \in T] \leq 2^u \cdot 2^{(H(\xi)-1)n} \leq 2^{\delta n + (H(\xi)-1)n} = o(1), \tag{6.10}$$

where the last equality holds for sufficiently small ξ.

Since $H_\infty(x \oplus \Psi) \geq k$, and E is a (k, ϵ) extractor, the probabilities in (6.9) and in (6.10) differ by at most ϵ. Thus we obtain

$$1 - \frac{O(k \log^{d-1} g)}{h \cdot \xi} \leq \epsilon + o(1),$$

which concludes the proof. □

6.1. Tightness. In this section we study in more detail the consequences of Corollary 6.5. Recall that it established the following tradeoff for an (l, s, ϵ) black-box PRG construction G such that G^f is computable by an oracle circuit of size g and depth d:

$$\log g \geq \left(\frac{2^l}{s \log s} \right)^{1/O(d)} - O(l).$$

We investigate what happens in the following two cases:

- The PRG construction is computable by a constant-depth circuit of size $g = \text{poly}(n)$.
- The PRG construction is based on the existence of a function $f : \{0,1\}^l \to \{0,1\}$ that is $(2^{\Omega(l)}, 1)$-hard for CKT.

Note that to obtain for $ATIME(O(1), \log n)$ a result analogous to Theorem 3.6 one needs both the above items.

PRG construction computable by a constant-depth circuit of size $g = \text{poly}(n)$. If one wants $g = \text{poly}(n)$ then $s \geq 2^l/l^{O(1)}$. In particular, the function $f : \{0,1\}^l \to \{0,1\}$ we start with must be hard for circuits of size at least $2^l/l^{O(1)}$. However, the next easy proposition shows that for such big sizes mild average-case and worst-case hardness *collapse*! Consequently, under such an assumption no worst-case to average-case hardness amplification is needed, and to get a PRG one could apply directly Theorem 4.3 or Theorem 4.7.

We state the next proposition for both CKT and $AC^0[d]$ since in Section 8 we discuss derandomization under hardness assumptions for constant-depth circuits.

PROPOSITION 6.11. *There is a constant k such that if $f : \{0,1\}^l \to \{0,1\}$ is $(k \cdot 2^l/l^{c-1}, 1)$-hard for CKT (respectively, $AC^0[d+k]$) then f is $(2^l/l^c, 1-1/l^c)$-hard for CKT (respectively, $AC^0[d]$).*

PROOF. Suppose not. Let C be a circuit of size at most $2^l/l^c$ (and depth d) such that

$$\Pr[C(U_l) \neq f(U_l)] < 1/l^c.$$

Then there are at most $2^l/l^c$ inputs x such that $C(x) \neq f(x)$. We can construct a circuit C' of size at most $2l \cdot 2^l/l^c$ which, given x, decides whether $C(x) \neq f(x)$ (recall our size measure is the number of edges). C' does a simple lookup table: For every x such that $C(x) \neq f(x)$ there is an AND gate with l connections to the corresponding input bits (or their negations). After this layer of AND gates we put an OR gate with $2^l/l^c$ connections. It is easy to see that such a circuit correctly decides whether $C(x) \neq f(x)$ and has size at most $2 \cdot 2^l/l^{c-1}$ (and depth 2).

Combining C and C' with a XOR we obtain a circuit of size at most $4 \cdot 2^l/l^{c-1}$ (and depth $d+3$) computing f everywhere. Contradiction (for $k = 4$). □

PRG construction based on $f : \{0,1\}^l \to \{0,1\}$ that is $(2^{\Omega(l)}, 1)$-hard. If one wants $s = 2^{\epsilon l}$ then $t \geq 2^{\Omega(l/d)}$. We now prove that these resources are also sufficient. Our approach shows that they allow for computing worst-case to mild average-case hardness amplification. One can then obtain a PRG construction from worst-case hard functions by combining this hardness amplification with the construction in Theorem 4.3.

To show that these resources are sufficient for computing worst-case to mild average-case hardness amplification we examine the construction in Theorem 3.3. First we show that *one* parity quantifier is sufficient for it, and then we note that this parity quantifier can be simulated in $ATIME(d, 2^{O(l/d)})$.

THEOREM 6.12. *If there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for CKT, then there is a function $f' \in \oplus \cdot ATIME(O(1), l)$ which is $(2^{\Omega(l)}, 1 - 1/\mathrm{poly}(l))$-hard for CKT.*

PROOF. We use the same construction in Theorem 3.3. Let us recall it. Fix a field F of size $4l^2$. Let H be the first (lexicographically) $\sqrt{|F|}$ elements of F.

Let $k := l/\log|H|$, so that f can be seen as mapping H^k to $\{0,1\}$. Let $\hat{f} : F^k \to F$ be

$$\hat{f}(x_1, \ldots, x_k) := \sum_{h_1,\ldots,h_k \in H} f(h_1, \ldots, h_k)\delta_{h_1}(x_1) \cdots \delta_{h_k}(x_k),$$

where for $h \in H$ and $x \in F$,

$$\delta_h(x) := \prod_{h' \in H, h' \neq h} \frac{x - h'}{h - h'}.$$

As pointed out in Babai *et al.* (1993) (see also Agrawal 2001), $\hat{f}$ has the required hardness, but $\hat{f}$ is not yet our final function since it is not boolean. Define $f'(x, i) := \hat{f}(x)_i$, note $|i| = O(\log l)$. It is easy to see that this final transformation preserves mild hardness.

Thus we only need to show that $f' \in \oplus \cdot ATIME(O(1), l)$. First we show that given $h \in H$ and $x \in F$ we can compute $\delta_h(x)$ in $ATIME(O(1), l)$.

To compute $\delta_h(x)$ we need to perform $\mathrm{poly}(l)$ field operations. Note that the field F can be found, and a single field operation computed, in time $\mathrm{poly}\log l^{O(1)} = \mathrm{poly}\log l$ (see Shoup 1990). Moreover, we can use the same space for all the field operations, for a total of $\mathrm{poly}\log l$ space. By Theorem 2.1 we can compute $\delta_h(x)$ in $ATIME(O(1), l)$.

Similarly, given $h_1, \ldots, h_k, x_1, \ldots, x_k$, we can compute $f(h_1, \ldots, h_k) \cdot \delta_{h_1}(x_1) \cdots \delta_{h_k}(x_k)$ in $ATIME(O(1), l)$. In fact, $f \in ATIME(O(1), l)$ by assumption and the total time to compute $\delta_{h_1}(x_1) \cdots \delta_{h_k}(x_k)$ is still $\mathrm{poly}(l)$, and moreover we can reuse the same space for the δ's. So by Theorem 2.1 this product can be computed in $ATIME(O(1), l)$.

What is left to do is to sum over all 2^l possible $h_1, \ldots, h_k$. Now note we can assume that the characteristic of F is 2, so addition equals XOR. In particular, the i-th output bit of f' is the parity of the i-th bit of $f(h_1, \ldots, h_k)\delta_{h_1}(x_1) \cdots \delta_{h_k}(x_k)$ over all 2^l possible $h_1, \ldots, h_k$. Thus $f' \in \oplus \cdot ATIME(O(1), l)$. □

Note that the parity quantifier in the above computation ranges over 2^l bits. The following easy lemma states that this parity can be computed in $ATIME(d, 2^{O(l/d)})$.

LEMMA 6.13. *For every integer $d \geq 1$, the parity of n bits can be computed in $ATIME(d, n^{O(1/d)})$.*

PROOF SKETCH. The idea is trading alternations for time taking advantage of the associativity of parity. Namely, we partition the input into $n^{O(1/d)}$ pieces, existentially guess the parity of each, and universally verify that each guess is correct (recursively with the same algorithm). □

Therefore $ATIME(d + O(1), 2^{O(l/d)})$ is necessary and sufficient for PRG constructions from worst-case hard functions.

Combining Theorems 4.3 and 6.12 we obtain Theorem 4.6.

THEOREM 4.6, RESTATED. *If there exists a function $f : \{0,1\}^l \to \{0,1\}$ in $A \oplus TIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for CKT then there is an n-PRG against CKT with seed length $O(\log n)$ and computable in $A \oplus TIME(O(1), \log n)$, and $BP \cdot A \oplus TIME(O(1), \log n) = A \oplus TIME(O(1), \log n)$.*

7. Worst-case hardness amplification

In this section we discuss worst-case hardness amplification. In particular, we prove a formal version of Theorem 4.5, establishing a negative result for black-box worst-case hardness amplifications. As mentioned in the introduction, a certain negative result for black-box worst-case hardness amplifications already follows from our previous results. Namely, if there is a black-box worst-case hardness amplification then combining this with our black-box PRG construction from mild average-case hardness (Theorem 4.3) one gets a black-box PRG construction from worst-case hardness, and the negative result in Corollary 6.5 applies. In this section we give a direct proof of a negative result for black-box worst-case hardness amplification. This direct proof yields a more general negative result than what one can get using the above approach.

The general ideas in our negative result are the same we employed in our negative result for black-box PRG constructions in Section 6, with the exception that "extractors" will be replaced with *list-decodable codes*: First we show that every black-box hardness amplification gives rise to a "good" list-decodable code. Then we show that "good" list-decodable codes are very sensitive to noise. Since constant-depth circuits are not, we get our negative result.

We now proceed to turn the above sketch into a formal proof.

DEFINITION 7.1. An oracle algorithm $\mathrm{Amp} : \{0,1\}^{l'} \to \{0,1\}$ is an (l, δ, s)-*black-box worst-case hardness amplification* if for every $f : \{0,1\}^l \to \{0,1\}$ and

for every $A : \{0,1\}^{l'} \to \{0,1\}$ such that

$$\Pr[A(U_{l'}) = \mathrm{Amp}^f(U_{l'})] \geq \delta,$$

there is an oracle circuit C of size at most s such that $C^A(x) = f(x)$.

Note in the above definition we have not specified the type of the circuit C (e.g. $CKT, AC^0[17], \ldots$) because it does not play a role in this section. Also note that, if Amp is an (l, δ, s)-black-box worst-case hardness amplification, then for every function $f : \{0,1\}^l \to \{0,1\}$, if f is $(s', 1)$-hard then Amp^f is $(s'/s, \delta)$-hard.

We note that in this notation the hardness amplification in Theorem 3.3 is an $(l, 1 - 1/\mathrm{poly}(l), \mathrm{poly}(l))$-black-box hardness amplification. It should also be noted that in this hardness amplification the input length increases only by a constant factor, i.e. $\mathrm{Amp}^f : \{0,1\}^{O(l)} \to \{0,1\}$. While this is what one needs for high-end derandomization (see Impagliazzo & Wigderson 1997), our negative result applies regardless of this.

We give the definition of list-decodable codes:

Definition 7.2. A code $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ is (δ, ρ)-*list-decodable* if for every $\bar{x} \in \{0,1\}^{\bar{n}}$,

$$\left|\{y \in \{0,1\}^n : \Delta(\bar{x}, C(y)) \leq \delta\}\right| \leq \rho,$$

where Δ is the *relative Hamming distance*: $\Delta(\bar{x}, \bar{y}) := \Pr_i[\bar{x}_i \neq \bar{y}_i]$. We refer to $x \in \{0,1\}^n$ as *messages* and to $C(x), x \in \{0,1\}^n$, as *codewords*.

Let Amp be a black-box worst-case hardness amplification. The following lemma, implicit in Sudan *et al.* (2001) and Trevisan & Vadhan (2002), states that if we consider the truth table of a function f as a message and the truth table of Amp^f as a codeword, then Amp can be seen as an encoding algorithm.

Lemma 7.3. *Let* Amp *be an* (l, δ, s)*-black-box worst-case hardness amplification. Then* $\mathrm{Enc} : \{0,1\}^{2^l} \to \{0,1\}^{\bar{n}}$ *defined as* $\mathrm{Enc}(f) := \mathrm{Amp}^f$ *is* $(1 - \delta, 2^{O(s \cdot \log s)})$*-list-decodable.*

Proof. Consider $A \in \{0,1\}^{\bar{n}}$. By definition of hardness amplification, for every f such that $\Pr_{x \in \{0,1\}^n}[A(x) = \mathrm{Amp}^f(x)] \geq \delta$, there is an oracle circuit C of size at most s such that $C^A(x) = f(x)$. Therefore the number of such codewords is bounded by the number of oracle circuits. Noting that there are at most $2^{O(s \cdot \log s)}$ oracle circuits of size at most s, and that $\Pr_{x \in \{0,1\}^n}[A(x) = \mathrm{Amp}^f(x)] = 1 - \Delta(A, \mathrm{Amp}^f)$, completes the proof. □

The following theorem states that constant-depth circuits cannot compute list-decodable codes even for very weak parameters.

THEOREM 7.4. *There is a constant γ, $0 < \gamma < 1$, such that the following holds. Let $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ be a $(\delta, 2^m)$-list-decodable code, with $m \leq \gamma n$. If C can be computed by a circuit of size g and depth d, then*

$$\log^{d-1} g \geq \Omega\left(\frac{n\delta}{m}\right).$$

Before proving Theorem 7.4 note that, in combination with Lemma 7.3, it yields the following negative result for black-box hardness amplification. While the following corollary holds even for *non*uniform circuits (like Corollary 6.5), we state a uniform version to point out the connection with the polynomial time hierarchy.

COROLLARY 7.5 (formal version of Theorem 4.5). *Suppose that* Amp *is an $(l, 1-\delta, s)$ black-box hardness amplification, and that* Amp^f *is in $ATIME(d,t)^f$. Then*

$$t \geq \Omega\left(\frac{2^l \delta}{s \log s}\right)^{1/(d+O(1))}.$$

In particular, for any constants $c > 0, \epsilon < 1$, there is no $(l, 1 - 1/l^c, 2^{l^\epsilon})$-black-box worst-case hardness amplification computable in $ATIME(O(1), 2^{o(l)})$. In particular, there is no black-box worst-case hardness amplification in the polynomial time hierarchy.

PROOF. By standard techniques (see e.g. Furst *et al.* 1984; Håstad 1987), the $ATIME(d,t)^f$ computation can be carried out by a circuit of depth $d + O(1)$ and size $2^{O(t)}$, where we view the oracle as part of the input. The result then follows from Lemma 7.3 and Theorem 7.4. □

REMARK 7.6. Corollary 7.5 is tight in the same way as Corollary 6.5: The only settings of parameters that are not ruled out either allow for the construction in Theorem 6.12, or else correspond to hardness assumptions so strong that worst-case hardness and average-case hardness collapse, and therefore worst-case hardness amplification is vacuous (see Section 6.1).

We now prove Theorem 7.4. The proof is very similar to the proof of Theorem 6.4, and again makes use of Lemmas 6.6 and 6.7.

PROOF OF THEOREM 7.4. Let $C_i(x)$ denote the i-th bit of $C(x)$. Let $\Psi \in \{0,1\}^n$ be a random noise vector where each bit is 1 independently with probability $O(m/n)$ so that for every fixed $x \in \{0,1\}^n$ we have $H_\infty(x \oplus \Psi) \geq m+1$ by Lemma 6.7. Let X be chosen at random in $\{0,1\}^n$.

The idea of the proof is to consider the quantity

$$\Pr_{i,X,\Psi}[C_i(X) \neq C_i(X \oplus \Psi)]$$

and to bound it using (1) the assumption that C is $(\delta, 2^m)$-list-decodable and (2) the low average sensitivity of constant-depth circuits (Lemma 6.6).

For every fixed $x \in \{0,1\}^n$, the list-decodability assumption tells us that there are at most 2^m messages whose codewords are at distance at most δ from $C(x)$. Fix any such message. Since $H_\infty(x \oplus \Psi) \geq m+1$, the probability that $x \oplus \Psi$ is equal to this message is at most $2^{-(m+1)}$. Therefore, by a union bound,

$$\Pr_{X,\Psi}[\Delta(C(X), C(X \oplus \Psi)) \leq \delta] \leq 2^m \cdot 2^{-(m+1)} = 1/2.$$

Hence

$$\begin{aligned} \Pr_{i,X,\Psi}[C_i(X) \neq C_i(X \oplus \Psi)] &\geq \Pr_{i,X,\Psi}[C_i(X) \neq C_i(X \oplus \Psi) \,|\, \Delta(C(X), C(X \oplus \Psi)) > \delta] \\ &\quad \cdot \Pr_{X,\Psi}[\Delta(C(X), C(X \oplus \Psi)) > \delta] \\ &\geq \delta \cdot \frac{1}{2}. \end{aligned} \tag{7.7}$$

On the other hand, by Lemma 6.6 we have

$$\Pr_{i,X,\Psi}[C_i(X) \neq C_i(X \oplus \Psi)] \leq m \cdot \frac{O(\log^{d-1} g)}{n}. \tag{7.8}$$

The theorem follows on putting together bounds (7.7) and (7.8). □

8. Derandomization from weaker assumptions

In this section we work on relaxing the hardness assumptions needed in our derandomization results. In particular, we prove Theorems 4.7 and 4.8.

We start with the latter. As explained in Section 3.2, Agrawal (2001) notices that all the proofs of correctness of the constructions described in Section 3.1 carry through against small constant-depth circuits with MAJORITY gates. Combining this with Theorem 4.6, and recalling that counting quantifiers can simulate parity quantifiers, one gets the following theorem.

THEOREM 4.8, RESTATED. *There is a constant c such that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in $CTIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for $TC^0[c+d]$, then there is an n-PRG against $TC^0[d]$ with seed length $O(\log n)$ and computable in $CTIME(O(1), \log n)$.*

In particular, if for every d there is a function $f : \{0,1\}^l \to \{0,1\}$ in $CTIME(O(1), l)$ that is $(2^{\Omega(l)}, 1)$-hard for $TC^0[d]$, then

$$BP \cdot CTIME(O(1), \log n) = CTIME(O(1), \log n).$$

Now we focus on Theorem 4.7. For proving it we use the same construction in Theorem 3.10, but without the worst-case to average-case hardness amplification step in Theorem 3.3. The correctness of this construction has already been proved in Agrawal (2001), so we only need to show that it has an implementation in $ATIME(O(1), \log n)$.

Recall that this construction combined the conditional PRG from Section 3.1 with Nisan's unconditional PRG (Theorem 3.9). We have already shown in Section 5 how to compute the conditional PRG. So it is only left to discuss Nisan's unconditional PRG. In particular, we need to show that the following items are computable in $ATIME(O(1), \log n)$:

- A family of $(\log n, \log^c n)$ designs of size n over a universe of size $\log^d n$, for every given value of $c \geq 1$ and some $d \geq c$.
- parity over $\log^c n$ bits, for every given value of c.

The result about parity has been proved in Lemma 6.13. We now show that the design construction in Nisan (1991) is computable in $ATIME(O(1), \log n)$.

LEMMA 8.1. *For every constant c there is a constant d such that there is a family $\{D_n\}$ of $(\log n, \log^c n)$ designs of size n over a universe of size $\log^d n$ with the following property: There is a machine in $ATIME(O(1), \log n)$ which, given n and $k \leq n$, computes the characteristic vector of the k-th set in D_n.*

PROOF. Let $l := \log n$. Let us first recall the construction in Nisan (1991). Let l^c be the cardinality of a field F. Let $d := 2c$, i.e. the universe size is $|F|^2 = l^d$. Given a string i of length l, we view the string as the coefficients of a univariate polynomial $\hat{i}$ with coefficients in F. The corresponding set is

$$S_i := \{a \circ \hat{i}(a) : a \in F\}.$$

It is pointed out by Nisan (1991) that $S_1, \ldots, S_n$ is an (l, l^c) design. Thus we only need to show that it is computable in $ATIME(O(1), l)$.

Our task is, given k and an element j of the universe, to decide whether $j \in S_k$ in $ATIME(O(1), l)$. Let $j|_{c \log l}$ be the first $c \log l$ bits of j. Now, $j \in S_k$ if and only if $j = j|_{c \log l} \circ \hat{k}(j|_{c \log l})$. To compute $\hat{k}(j|_{c \log l})$ we need to perform $\mathrm{poly}(l)$ field operations. Note that the field F can be found, and a single field operation computed, in time $\mathrm{poly} \log l^{O(1)} = \mathrm{poly} \log l$ (see Shoup 1990). Moreover, we can use the same space for all the field operations, for a total of $\mathrm{poly} \log l$ space. Consequently, we can decide whether $j \in S_k$ in $ATIME(O(1), l)$ by Theorem 2.1. □

This completes the proof of Theorem 4.7.

THEOREM 4.7, RESTATED. *There is a constant c such that if there is a function $f : \{0,1\}^l \to \{0,1\}$ in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/l^b)$-hard for $AC^0[c \cdot \max(b, d)]$, then there is an $(n, 1/\log^{O(1)} n)$-PRG against $AC^0[d]$ with logarithmic seed length and computable in $ATIME(O(1), \log n)$.*

In particular, if there exists a constant b such that for every d there is a function in $ATIME(O(1), l)$ that is $(2^{\Omega(l)}, 1 - 1/l^b)$-hard for $AC^0[d]$, then $BP \cdot ATIME(O(1), \log n) = ATIME(O(1), \log n)$.

9. Noise sensitivity of constant-depth circuits

In this section we prove Lemma 6.6. Recall that $\oplus$ denotes bitwise XOR.

LEMMA 6.6, RESTATED. *Let $C : \{0,1\}^n \to \{0,1\}$ be a circuit of size g and depth d. Let $X \in \{0,1\}^n$ be a random input and let $\Psi \in \{0,1\}^n$ be a random noise vector where each bit is 1 independently with probability $\delta < 1/2$. Then*

$$\Pr_{X,\Psi}[C(X) \neq C(X \oplus \Psi)] \leq O(\delta \log^{d-1} g).$$

Although it is well known that constant-depth circuits have small noise sensitivity, the bound we need is not stated anywhere, and to prove it we need to introduce the Fourier machinery and then combine several results.

We now set up the usual Fourier machinery (see e.g. Linial *et al.* 1993 for details). Whenever we discuss Fourier coefficients, we will use $\{+1, -1\}$ instead of $\{0,1\}$. When working over $\{+1, -1\}$ we denote also by $\oplus$ bitwise multiplication. Let $f : \{+1, -1\}^n \to \{+1, -1\}$ be any boolean function. Then f has a unique representation as a multilinear polynomial in $x_1, \ldots, x_n$ of total degree at most n. The S-th Fourier coefficient of f, denoted $\hat{f}(S)$, is the coefficient of the monomial $\prod_{i \in S} x_i$ in this polynomial. We also have, by Parseval's identity,

$$\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1, \quad \text{where } [n] := \{1, \ldots, n\}.$$

The first result we need is a characterization of the noise sensitivity of a function in terms of its Fourier coefficients. Such a characterization is given by O'Donnell (2002).

LEMMA 9.1 (O'Donnell 2002). *Let $f : \{+1,-1\}^n \to \{+1,-1\}$ be any boolean function. Let $X \in \{+1,-1\}^n$ be a random input and let $\Psi \in \{+1,-1\}^n$ be a random noise vector where each bit is 1 independently with probability $\delta < 1/2$. Then*

$$\Pr_{X,\Psi}[f(X) \neq f(X \oplus \Psi)] = \frac{1}{2} - \frac{1}{2} \sum_{S \subseteq [n]} (1-2\delta)^{|S|} \hat{f}(S)^2.$$

The second result we need is a bound on the Fourier coefficients of functions computed by constant-depth circuits. Boppana (1997) gives a tight bound on the *average sensitivity* of constant-depth circuits. Combining this bound with the characterization of average sensitivity in terms of Fourier coefficients given by Kahn *et al.* (1988) (see also Linial *et al.* 1993), we obtain the following bound.

LEMMA 9.2 (Kahn *et al.* 1988; Boppana 1997). *Let f be computable by a circuit of size g and depth d. Then*

$$\sum_{S \subseteq [n]} |S| \hat{f}(S)^2 \leq O(\log^{d-1} g).$$

We can now prove Lemma 6.6.

PROOF OF LEMMA 6.6. Let $f : \{+1,-1\}^n \to \{+1,-1\}$ be the function computed by C. We have

$$\begin{aligned}
&\Pr_{X,\Psi}[C(X) \neq C(X \oplus \Psi)] \\
&\qquad = \frac{1}{2} - \frac{1}{2} \sum_{S \subseteq [n]} (1-2\delta)^{|S|} \hat{f}(S)^2 && \text{(by Lemma 9.1)} \\
&\qquad \leq \frac{1}{2} - \frac{1}{2} \sum_{S \subseteq [n]} (1-2\delta|S|) \hat{f}(S)^2 && \text{(by Bernoulli's inequality)} \\
&\qquad = \frac{1}{2} \sum_{S \subseteq [n]} 2\delta|S| \hat{f}(S)^2 && \text{(by Parseval's identity)} \\
&\qquad \leq O(\delta \log^{d-1} g) && \text{(by Lemma 9.2),}
\end{aligned}$$

which proves the lemma. □

10. Open problems

Notice the gap between Theorems 4.6 and 4.8: While parity quantifiers are sufficient for a PRG construction from a worst-case hard function, we seem to need hardness against circuits with MAJORITY gates for proving its correctness. (Recall that, for constant-depth computation, MAJORITY gates are strictly more powerful than PARITY gates.) This is what prevents us from using the known exponential lower bounds for constant-depth circuits with PARITY gates (Razborov 1987; Smolensky 1987) to construct a PRG against small constant-depth circuits with PARITY gates. (The related question of the existence of hardcore sets for such circuits was independently raised by Eric Allender and Sambuddha Roy, personal communication, Nov. 2002.) This is also what prevents us from constructing, under some complexity assumption for small constant-depth circuits, an $(n, 1/n)$-PRG against $AC^0[d]$ with seed length $O(\log n)$ (Theorems 3.10 and 4.7 only give an $(n, 1/\log^{O(1)} n)$-PRG.)

The bottleneck is *not* the NW PRG, whose proof of correctness does not rely on the use of MAJORITY gates, rather the bottleneck is hardness amplification. As we have seen in Section 7, black-box hardness amplification gives rise to "good" list-decodable codes. Moreover, these codes are *locally* list-decodable, i.e. for every corrupted codeword there is a small circuit which, given oracle access to the corrupted codeword, computes the associated message everywhere (see, e.g., Sudan *et al.* 2001). The difficulty is that, while parity quantifiers are sufficient for computing "good" error correcting codes (e.g. linear codes), it seems that MAJORITY gates are needed for locally list-decoding of "good" error correcting codes. We believe that MAJORITY gates are indeed necessary. In particular we conjecture that no locally list-decodable code with the parameters of the code in Sudan *et al.* (2001) can be locally list-decoded by small constant-depth circuits with PARITY gates.

Acknowledgements

Research supported in part by NSF grant CCR-0133096 and US-Israel BSF grant 2002246. A preliminary version of this paper was published in *Proceedings of the 18th Annual Conference on Computational Complexity*, IEEE, Aarhus, 2003, pp. 53–69. Many thanks to Salil Vadhan for encouragement, illuminating discussions and suggesting the problem of the derandomization of circuit classes under *DLOGTIME*-uniformity. Madhu Sudan suggested the sensitivity approach for proving negative results for AC^0. We are grateful to Oded Goldreich for discussions and many useful suggestions that helped improve the

presentation. Adam Klivans pointed out Agrawal (2001) to us. Thanks to Ronen Shaltiel and the anonymous referees for useful comments.

References

M. Agrawal (2001). Hard sets and pseudo-random generators for constant depth circuits. In *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science* (Bangalore), Lecture Notes in Comput. Sci. 2245, Springer, 58–69.

M. Ajtai (1993). Approximate counting with uniform constant-depth circuits. In *Advances in Computational Complexity Theory* (New Brunswick, NJ, 1990), Amer. Math. Soc., Providence, RI, 1–20.

E. W. Allender & K. W. Wagner (1990). Counting hierarchies: polynomial time and constant depth circuits. *Bull. Eur. Assoc. Theoret. Comput. Sci.* **40**, 182–194. URL `citeseer.nj.nec.com/allender90counting.html`.

L. Babai, L. Fortnow, N. Nisan & A. Wigderson (1993). BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complexity* **3**, 307–318.

Z. Bar-Yossef, O. Reingold, R. Shaltiel & L. Trevisan (2002). Streaming through combinatorial objects. In *17th Annual IEEE Conference on Computational Complexity*, IEEE Computer Soc., Los Alamitos, CA.

D. A. M. Barrington, N. Immerman & H. Straubing (1990). On uniformity within NC^1. *J. Comput. System Sci.* **41**, 274–306.

M. Blum & S. Micali (1984). How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.* **13**, 850–864.

R. B. Boppana (1997). The average sensitivity of bounded-depth circuits. *Inform. Process. Lett.* **63**, 257–261.

M. Cryan & P. B. Miltersen (2001). On pseudorandom generators in NC^0. In *26th International Symposium on Mathematical Foundations of Computer Science (MFCS 01)*, Springer, 272–284.

M. L. Furst, J. B. Saxe & M. Sipser (1984). Parity, circuits, and the polynomial-time hierarchy. *Math. Systems Theory* **17**, 13–27.

O. Goldreich (1997). A sample of samplers—a computational perspective on sampling (survey). *Electronic Colloq. Comput. Complexity (ECCC)* **4**(020). URL `citeseer.nj.nec.com/goldreich97sample.html`.

T. Hartman & R. Raz (2003). On the distribution of the number of roots of polynomials and explicit weak designs. *Random Structures Algorithms* **23**, 235–263.

J. Håstad (1987). *Computational Limitations of Small-Depth Circuits*. MIT Press.

R. Impagliazzo (1995). Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science* (Milwaukee, WI), IEEE, 538–545.

R. Impagliazzo & M. Naor (1996). Efficient cryptographic schemes provably as secure as subset sum. *J. Cryptology* **9**, 199–216. URL `citeseer.nj.nec.com/impagliazzo96efficient.html`.

R. Impagliazzo, R. Shaltiel & A. Wigderson (2000). Extractors and pseudo-random generators with optimal seed length. In *Proc. 32nd Annual ACM Symposium on Theory of Computing* (Portland, OR), 1–10. See also ECCC TR00-009.

R. Impagliazzo & A. Wigderson (1997). $P = BPP$ if E requires exponential circuits: derandomizing the XOR lemma. In *Proc. 29th Annual ACM Symposium on Theory of Computing* (El Paso, TX), 220–229.

J. Kahn, G. Kalai & N. Linial (1988). The influence of variables on boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science* (White Plains, NY), IEEE, 68–80.

M. Kharitonov, A. V. Goldberg & M. Yung (1989). Lower bounds for pseudo-random number generators. In *30th Annual Symposium on Foundations of Computer Science* (Research Triangle Park, NC), IEEE, 242–247.

A. R. Klivans (2001). On the derandomization of constant depth circuits. In *Proc. 5th International Workshop on Randomization and Approximation Techniques in Computer Science*.

A. R. Klivans & D. van Melkebeek (1999). Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Annual ACM Symposium on Theory of Computing* (Atlanta, GA, 1999), ACM, New York, 659–667.

N. Linial, Y. Mansour & N. Nisan (1993). Constant depth circuits, Fourier transform, and learnability. *J. ACM* **40**, 607–620.

A. Lubotzky, R. Phillips & P. Sarnak (1988). Ramanujan graphs. *Combinatorica* **8**, 261–277.

M. Naor & O. Reingold (1997). Number-theoretic constructions of efficient pseudo-random functions. In *38th Annual Symposium on Foundations of Computer Science* (Miami Beach, FL), IEEE, 458–467.

V. A. Nepomnjaščiĭ (1970). Rudimentary predicates and Turing calculations. *Soviet Math. Dokl.* **11**, 1462–1465.

N. Nisan (1991). Pseudorandom bits for constant depth circuits. *Combinatorica* **11**, 63–70.

N. Nisan & A. Wigderson (1994). Hardness vs randomness. *J. Comput. System Sci.* **49**, 149–167.

N. Nisan & D. Zuckerman (1996). Randomness is linear in space. *J. Comput. System Sci.* **52**, 43–52.

R. O'Donnell (2002). Hardness amplification within NP. In *Proc. 34th Annual ACM Symposium on Theory of Computing*, ACM, 751–760.

R. Raz, O. Reingold & S. Vadhan (1999). Extracting all the randomness and reducing the error in Trevisan's extractors. In *Annual ACM Symposium on Theory of Computing* (Atlanta, GA, 1999), ACM, New York, 149–158.

A. A. Razborov (1987). Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Mat. Zametki* **41**, 598–607, 623 (in Russian).

R. Shaltiel (2002). Recent developments in explicit constructions of extractors. *Bull. Eur. Assoc. Theoret. Comput. Sci.* **77**, 182–194.

R. Shaltiel & C. Umans (2001). Simple extractors for all min-entropies and a new pseudo-random generator. In *42nd Annual Symposium on Foundations of Computer Science*, IEEE.

V. Shoup (1990). New algorithms for finding irreducible polynomials over finite fields. *Math. Comp.* **54**, 435–447.

M. Sipser (1983). Borel sets and circuit complexity. In *Proc. 5th Annual ACM Symposium on Theory of Computing* (Boston, MA), 61–69.

R. Smolensky (1987). Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 9th Annual ACM Symposium on Theory of Computing* (New York), 77–82.

M. Sudan, L. Trevisan & S. Vadhan (2001). Pseudorandom generators without the XOR lemma. *J. Comput. System Sci.* **62**, 236–266.

J. Torán (1988). *Structural properties of the counting hierarchies*. Ph.D. thesis, Facultat d'Informatica de Barcelona, Barcelona.

L. Trevisan (2001). Extractors and pseudorandom generators. *J. ACM* **48**, 860–879. URL `citeseer.nj.nec.com/trevisan99extractors.html`.

L. Trevisan & S. Vadhan (2002). Pseudorandomness and average-case complexity via uniform reductions. In *Proc. 17th Annual IEEE Conference on Computational Complexity* (Montréal), IEEE, 129–138.

C. Umans (2002). Pseudo-random generators for all hardnesses. In *Proc. 34th ACM Symposium on Theory of Computing*, ACM Press, 627–634.

L.. Valiant (1977). Graph-theoretic arguments in low-level complexity. In *Mathematical Foundations of Computer Science* (Tatranská Lomnica), Lecture Notes in Comput. Sci. 53, Springer, Berlin, 162–176.

H. Vollmer (1999). *Introduction to Circuit Complexity*. Springer, Berlin.

K.. Wagner (1986). The complexity of combinatorial problems with succinct input representation. *Acta Inform.* **23**, 325–356.

A. C. Yao (1982). Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science* (Chicago, IL), IEEE, 80–91.

X. Yu & M. Yung (1994). Space lower-bounds for pseudorandom-generators. In *9th Annual Structure in Complexity Theory Conference*, IEEE Computer Soc., Los Alamitos, CA, 186–197.

Manuscript received 7 September 2003

Emanuele Viola
Division of Engineering
and Applied Sciences
Harvard University
Cambridge, MA 02138, U.S.A.
`viola@eecs.harvard.edu`