comput. complex. **14** (2005), 256 – 279
1016-3328/05/030256–24
DOI 10.1007/s00037-005-0197-7

computational complexity

DERANDOMIZING ARTHUR–MERLIN GAMES USING HITTING SETS

Peter Bro Miltersen and N. V. Vinodchandran

Abstract. We prove that **AM** (and hence Graph Nonisomorphism) is in **NP** if for some $\epsilon > 0$, some language in **NE** ∩ **coNE** requires nondeterministic circuits of size $2^{\epsilon n}$. This improves results of Arvind and Köbler and of Klivans and van Melkebeek who proved the same conclusion, but under stronger hardness assumptions.

The previous results on derandomizing **AM** were based on pseudorandom generators. In contrast, our approach is based on a strengthening of Andreev, Clementi and Rolim's hitting set approach to derandomization. As a spin-off, we show that this approach is strong enough to give an easy proof of the following implication: for some $\epsilon > 0$, if there is a language in **E** which requires nondeterministic circuits of size $2^{\epsilon n}$, then **P** = **BPP**. This differs from Impagliazzo and Wigderson's theorem "only" by replacing deterministic circuits with nondeterministic ones.

Keywords. Derandomization, interactive proof systems, complexity classes, graph nonisomorphism.

Subject classification. 68Q15, 68Q17.

1. Introduction

Using hardness for simulating randomness has been a fundamental idea in complexity theory. The main objective is to find nontrivial deterministic simulations of an entire class of randomized algorithms (rather than just a specific one) under certain complexity-theoretic hardness assumptions. Typically, the assumptions are in the form of the existence of functions in a uniform complexity class (for example **EXP**) that cannot be computed or approximated by a certain nonuniform class (for example polynomial size circuits). An early seminal result is the following result of Nisan and Wigderson that was proved by constructing a *pseudorandom generator*.

Theorem 1.1 (Nisan–Wigderson). *Let $\epsilon > 0$ be any constant. If there exists a language L in* **E** *so that for all but finitely many n, any circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $1/2 + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, then* **P** = **BPP**.

The hardness assumption in Theorem 1.1 is "average-case" rather than worst case. Substantial research has been done in order to remedy this and arguably the most remarkable result is a theorem due to Impagliazzo & Wigderson (1997). They showed the following improvement of Theorem 1.1.

THEOREM 1.2 (Impagliazzo–Wigderson). *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{E}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ has circuit complexity at least $2^{\epsilon n}$, then* $\mathbf{P} = \mathbf{BPP}$.

The proof of this theorem is technical and is built on the results of many earlier papers, including Babai *et al.* (1993); Blum & Micali (1984); Goldreich & Levin (1989); Impagliazzo (1995); Nisan & Wigderson (1994); Yao (1982). This result has been subsequently simplified and extended to get derandomization results for a range of parameters (Impagliazzo *et al.* 1999, 2000; Shaltiel & Umans 2001; Sudan *et al.* 2001; Umans 2003).

Although much research has gone into derandomizing **BPP** and **RP**, derandomization of classes like **AM** has received attention only recently. The class **AM** was defined, by Babai (1985) and Babai & Moran (1988), as a natural randomized (and interactive) version of the class **NP**. A number of natural computational problems have been shown to be in **AM** but are not known to be in **NP** (Babai 1985, 1992; Babai & Moran 1988; Goldreich *et al.* 1991; Goldwasser & Sipser 1989). Most have a group-theoretic flavor. The most celebrated one among them is the Graph Nonisomorphism problem. A complete derandomization of **AM** (that is, a proof of the statement $\mathbf{AM} = \mathbf{NP}$) would immediately give polynomial size membership proofs for positive instances of Graph Nonisomorphism. In contrast, the lengths of the shortest proofs known, without any assumptions, are exponential in the sizes of the graphs (Babai *et al.* 1983; Babai & Luks 1983).

Arvind & Köbler (2001) showed that the construction of Nisan & Wigderson (1994) can be extended to the nondeterministic setting to get pseudorandom generators which can be used to completely derandomize **AM**. As in the case of Nisan & Wigderson (1994), they needed an average-case hardness assumption in order to construct the generator. To be precise, Arvind & Köbler (2001) showed the following theorem.

THEOREM 1.3 (Arvind–Köbler). *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{NE} \cap \mathbf{coNE}$[1] *so that for all but finitely many n any nondeter-*

[1] Arvind and Köbler only state the theorem under the assumption $L \in \mathbf{E}$, but their proof easily generalizes.

ministic circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $1/2 + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, then $\mathbf{AM} = \mathbf{NP}$.

Klivans & van Melkebeek (2002) constructed generators for derandomizing **AM** under a worst-case hardness assumption. The main observation they make is that the proof of Impagliazzo & Wigderson (1997) *relativizes*. This leads to the following theorem.

THEOREM 1.4 (Klivans–van Melkebeek). *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{NE} \cap \mathbf{coNE}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ has oracle circuit complexity at least $2^{\epsilon n}$ with oracle gates for* SAT, *then* $\mathbf{AM} = \mathbf{NP}$.

Here, *oracle circuits* are Boolean circuits which contain special gates called *oracle gates*. These oracle gates are of unbounded fan-in (but a gate of fan-in r contributes size r to the circuit) and can be used for oracle access to a language, in this case SAT. The output of the gate on a string x is 1 if $x \in$ SAT. Otherwise the output is 0.

Arvind & Köbler (2001) and van Melkebeek (1998) asked whether $\mathbf{AM} = \mathbf{NP}$ follows from the existence of a language in $\mathbf{NE} \cap \mathbf{coNE}$ which does not have subexponential nondeterministic circuit complexity. In this paper, we answer this question affirmatively, proving the following theorem which improves Theorem 1.3 as well as Theorem 1.4.

THEOREM 1.5. *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NE** $\cap$ **coNE** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then* $\mathbf{AM} = \mathbf{NP}$.

Here an SV (single-valued) nondeterministic circuit is a restriction of the notion of a nondeterministic circuit: in an SV-nondeterministic circuit, there are two output bits, the real output bit, and a flag, indicating whether the computation has been correctly performed. On both positive and negative instances, if the flag is on, the output bit should be correct. Additionally, for all instances, there should be some setting of the nondeterministic choice bits that make the flag turn on.

To see the difference between our result and the result of Klivans and van Melkebeek, we can informally say that SV-nondeterministic circuits of the stated size form a nonuniform and exponential analogue of $\mathbf{NP} \cap \mathbf{coNP}$, while oracle circuits with SAT as an oracle form a nonuniform and exponential analogue of $\mathbf{P}^{\mathbf{NP}}$ (see Section 7 for a more detailed discussion on this issue).

Our approach to proving Theorem 1.5 is completely different from the techniques of Arvind & Köbler (2001) and of Klivans & van Melkebeek (2002). Instead of using pseudorandom generators, we use a strengthened version of the *hitting set generator* approach to derandomization, due to Andreev, Clementi & Rolim (1997a). They gave, independently and almost simultaneously to Impagliazzo and Wigderson's work, two different conditions, each implying $\mathbf{P} = \mathbf{BPP}$. The conditions were much stronger than the hardness assumption in the Impagliazzo–Wigderson theorem; one of them essentially stating that there should be an algorithm operating in time polynomial in the size of its output, which on input n, m outputs the truth table of a Boolean function f from $\{0,1\}^n$ to $\{0,1\}^m$ with circuit complexity within a certain additive low order term of the maximum possible.

Their proof had two parts. First it is shown that the stated condition implies the existence of a certain hitting set generator (for definition of hitting set, see Section 2). Then it is shown that the existence of such a generator implies $\mathbf{P} = \mathbf{BPP}$ (it is easy to show that it implies $\mathbf{P} = \mathbf{RP}$). The latter part of the proof, i.e., the fact that the existence of the hitting set generator is enough to show $\mathbf{P} = \mathbf{BPP}$ was proved already by Andreev, Clementi & Rolim (1996b, 1998). Since then, the proof of this implication was simplified enormously (Andreev *et al.* 1997b; Buhrman & Fortnow 1999; Goldreich *et al.* 2000).

It was not (and is still not) clear if the hitting set approach to derandomization can be pushed to yield the Impagliazzo–Wigderson theorem. However, in this paper, we show, by strengthening the first part of their proof, that it can be pushed to yield the following statement.

Theorem 1.6. *Let $\epsilon > 0$ be any constant. If there exists a language L in $\mathbf{E}$ so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then $\mathbf{P} = \mathbf{BPP}$.*

Note that this differs from the Impagliazzo–Wigderson theorem "only" in the assumption being about SV-nondeterministic circuits, rather than about deterministic ones.

Our main technical result is the following theorem, describing a procedure for turning the truth table of a Boolean function with big circuit complexity into a hitting set for circuits with very high acceptance probability (for precise definitions of the terms in the theorem, we refer the reader to Section 2).

Theorem 1.7. *For any $\epsilon > 0$ and $q \geq 1$, there is a polynomial time procedure P with the following properties. Let $f : \{0,1\}^m \to \{0,1\}$ be a function that* *cannot be computed by SV-nondeterministic circuits of size less than* $2^{\epsilon m}$ *for almost all* m. *Then there are constants* $\delta = \delta(\epsilon) < \epsilon$ *and* $k > q$ *such that, given the truth table of* $f : \{0,1\}^{kl} \to \{0,1\}$ *as input,* P *outputs a hitting set* $H_f \subseteq \{0,1\}^n$ *for co-nondeterministic circuits of size* n^q *with threshold* $1 - 2^{-n+n^{\delta}}$, *where* $n = (2l)2^{2l}$.

The main ingredient we add to the techniques of Andreev, Clementi & Rolim (1997a) to prove Theorem 1.7 is to first replace the truth table of f by a *multidimensional* encoding of f using an appropriate error-correcting code. An intuitive reason why this turns out to be useful is as follows. The technique of Andreev *et al.* (1997a) is based on *compression* in the form of hashing. As was previously noted by Miltersen (1998), hashing becomes a much easier and cleaner operation when applied to data encoded in an error-correcting code. This essentially enables us to compress a *multidimensional* object along *all* dimensions, rather than just compressing a *two-dimensional* object along *one* dimension, as done in Andreev *et al.* (1997a). We use *low degree extension* (Babai *et al.* 1991) of f for encoding purposes.

While the above intuition was useful for coming up with the proof of Theorem 1.7, the self-contained proof we present in Section 3 is quite short and the above intuition should not be necessary for understanding it.

Having proven Theorem 1.7, we combine it with a variation of a lemma from Andreev *et al.* (1996a) (Lemma 2.4 of the present paper), and prove the following.

COROLLARY 1.8. *For any constant* $\epsilon > 0$, *there is a constant* $\tau > 0$ *so that the following holds. There is a deterministic polynomial time procedure which, given as input the truth table of a Boolean function* $f : \{0,1\}^m \to \{0,1\}$ *(i.e.,* 2^m *bits) with SV-nondeterministic circuit complexity at least* $2^{\epsilon m}$, *outputs a hitting set in* $\{0,1\}^n$ *with threshold* $1/2$ *for co-nondeterministic circuits of size* n, *where* $n = \lceil 2^{\tau m} \rceil$.

This corollary then immediately implies Theorem 1.5. To prove Theorem 1.6, we use the result of Andreev *et al.* (1998), stating that the hitting set generator of Corollary 1.8 derandomizes **BPP**.

The existence of explicit dispersers is the main technical tool that we have to employ in order to prove Corollary 1.8. But we would like to point out that any *relativizable* proof of Corollary 1.8 (such as ours) *has* to use the existence of explicit dispersers. More precisely, we note that any relativizable worst case hardness-based hitting set generator defines a disperser. The truth of this statement can be seen by arguing along the lines given by Trevisan (2001) where an analogous statement for extractors is implicitly established. We formally state this hitting set/disperser correspondence as Theorem 4.2 in Section 4.

In Section 5, we note that the techniques used in this paper to prove our main derandomization result can be used to show other hardness-randomness tradeoffs for **AM**. More precisely, we show the following theorem.

THEOREM 1.9. *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{NEXP} \cap \mathbf{coNEXP}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{n^{1/2+\epsilon}}$, then* $\mathbf{AM} \subseteq \mathbf{NQuasiP}$.

Most of the results in this paper were first published in Miltersen & Vinodchandran (1999). Since then there has been significant progress in derandomizing **AM**. In Section 7, we briefly discuss these results.

2. Terminology and preliminary results

Lower case Greek letters denote rational constants between 0 and 1. The symbol log denotes $\log_2$.

Complexity classes. We assume standard complexity-theoretic notations and definitions such as the definitions of standard complexity classes **P**, **NP**, **E**, **NE**, **NEXP** and **BPP**. Please refer to the textbooks Balcázar *et al.* (1995); Papadimitriou (1994) for these. Here we only give the definition of the class **AM**.

A language L is defined[2] to be in **AM** if there is a language $L' \in \mathbf{P}$ and a polynomial p so that for all $x \in \{0,1\}^n$,

$$\begin{aligned} x \in L &\Rightarrow \Pr_{y \in \{0,1\}^{p(n)}} (\exists z \in \{0,1\}^{p(n)}\ (x,y,z) \in L') = 1, \\ x \notin L &\Rightarrow \Pr_{y \in \{0,1\}^{p(n)}} (\exists z \in \{0,1\}^{p(n)}\ (x,y,z) \in L') \le \frac{1}{2}. \end{aligned}$$

An SVNP-procedure (SV meaning *Single-Valued* (Selman 1996)) computing a function f is a polynomial time nondeterministic procedure so that every computation path on input x either produces $f(x)$ or rejects. Furthermore, at least one computation path must produce $f(x)$.

[2] The original definition in Babai (1985) of **AM** is a two-sided error version. But it is shown in Fürer *et al.* (1989) that this definition is equivalent to the one-sided error version, which we give here.

Circuits. A *nondeterministic* Boolean circuit C contains, in addition to the standard AND, OR and NOT gates, *choice gates* of fan-in 0. The circuit evaluates to 1 on an input x, and we say that $C(x) = 1$, if there is some assignment of truth values to the choice gates that makes the circuit evaluate to 1. Otherwise $C(x) = 0$. A *co-nondeterministic* circuit C is defined similarly: the circuit evaluates to 0 on an input x, and we say that $C(x) = 0$, if there is some assignment of truth values to the choice gates that makes the circuit evaluate to 0. Otherwise $C(x) = 1$.

Similarly, an *SV-nondeterministic* circuit C computing a function f has, in addition to its usual output, an extra output bit called the *flag*. For any input x and any setting of the choice gates, if the flag is on, the circuit should output the correct value of $f(x)$. Furthermore, for any x, there should be some setting of the choice gates that turn the flag on. It is easy to see that a Boolean function f has an SV-nondeterministic circuit of size $O(s(n))$ if and only if f has a nondeterministic circuit of size $O(s(n))$ and a co-nondeterministic circuit of size $O(s(n))$.

Oracle circuits (Wilson 1985) are Boolean circuits with special gates called *oracle gates*. These oracle gates can be of arbitrary fan-in, though a gate of fan-in r contributes size r to the circuit and can be used for oracle access to a fixed language, say L. The output of the gate on a string x is 1 if $x \in L$, otherwise the output is 0. Nondeterministic and SV-nondeterministic oracle circuits are defined by combining the above definitions in the obvious way.

Dispersers. For the purposes of this paper (there are more parameters in the general definition), a *disperser* with threshold t is a bipartite graph $G = (U, V, E)$ such that, for any subset $S \subseteq U$ with $|S| \geq t$, more than half the vertices of V are adjacent to S.

Also for the purposes of this paper, for constants $\delta, \gamma > 0$ and $k \geq 1$, an *explicit* (n^δ, n^γ)*-disperser* is a family of dispersers $G_n = (U_n, V_n, E_n)$, $n = 1, 2, \ldots$, with $|U_n| = \{0,1\}^n$, $|V_n| = \{0,1\}^{\lceil n^\gamma \rceil}$, and threshold $t_n = 2^{n^\delta}$ so that there is a deterministic polynomial time algorithm which on input $x \in U_n$ enumerates the vertices in V_n adjacent to x (in particular, the outdegree of every $x \in U_n$ must be polynomial).

The first construction of explicit dispersers with these parameters was given by Saks, Srinivasan & Zhou (1998). Their construction was subsequently simplified and improved in several papers including Ta-Shma (2002); Ta-Shma *et al.* (2001); Trevisan (2001). For an excellent exposition on recent developments in the construction of explicit extractors and dispersers we refer the reader to the survey paper by Shaltiel (2002). For the theorem below, the original result by Saks, Srinivasan and Zhou suffices.

Theorem 2.1 (Saks–Srinivasan–Zhou). *For any $\delta > 0$, there is a $\gamma > 0$ so that an explicit (n^δ, n^γ)-disperser exists.*

Hitting sets. A *hitting set* with threshold $\delta(n)$ for co-nondeterministic circuits of size $s(n)$ is a subset H of $\{0,1\}^n$ so that for any co-nondeterministic circuit C of size $s(n)$, taking n inputs and producing one output, the following holds: if $\Pr_{x\in\{0,1\}^n}[C(x) = 1] \geq \delta(n)$, then $\exists x \in H, C(x) = 1$. (The more usual definition of hitting sets for deterministic circuits is analogous).

With this definition, the following proposition is easy to prove.

Proposition 2.2. *If there is an SVNP-procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ with threshold $1/2$ for co-nondeterministic circuits of size n, then $\mathbf{AM} = \mathbf{NP}$.*

Now we state a lemma from Andreev *et al.* (1996a)[3]. Actually, the lemma is already implicit in Sipser (1988). It shows that in fact it is sufficient to construct hitting sets with threshold much bigger than 1/2. This lemma is a consequence of the existence of explicit dispersers. Indeed, in Sipser (1988), it was Sipser's motivation for defining the notion of a disperser.

Lemma 2.3 (Sipser, Andreev–Clementi–Rolim). *For any constant $\delta > 0$, there are constants $q \geq 1$ and $\gamma > 0$ so that the following holds. There is a polynomial time procedure which, on input H where H is a hitting set in $\{0,1\}^n$ with threshold $1-2^{-n+n^\delta}$ for circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $1/2$ for circuits of size n', where $n' = \lceil n^\gamma \rceil$.*

What we actually need is the analogous lemma for co-nondeterministic circuits. This lemma is proved exactly as Lemma 2.3 using explicit dispersers. To make the paper self-contained, we give the proof. In the proof, for a circuit C, let $Z(C)$ denote the set of instances for which C evaluates to 0.

Lemma 2.4. *For any constant $\delta > 0$, there are constants $q \geq 1$ and $\gamma > 0$ so that the following holds. There is a polynomial time procedure which, on input H where H is a hitting set in $\{0,1\}^n$ with threshold $1 - 2^{-n+n^\delta}$ for co-nondeterministic circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $1/2$ for co-nondeterministic circuits of size n', where $n' = \lceil n^\gamma \rceil$.*

[3] The reader should note that the lemma can only be found in the *revised* version of the cited ECCC technical report.

Proof. Let $\delta > 0$ be fixed. According to Theorem 2.1, there is a γ so that an explicit (n^δ, n^γ)-disperser exists. Let $G_n = (U_n, V_n, E_n)$ be this disperser, i.e., with $n' = \lceil n^\gamma \rceil$, $U_n = \{0,1\}^n$, $V_n = \{0,1\}^{n'}$, and for all subsets S of U_n of size at least 2^{n^δ}, more than half the vertices of V are adjacent to S.

Let $H \subseteq \{0,1\}^n$ be a hitting set with threshold $1 - 2^{-n+n^\delta}$ for co-nondeterministic circuits of size n^q, where the constant q will be determined below.

Note that H is a subset of U_n. Let H' be the set of vertices in V_n adjacent to H. As the disperser is explicit, H' can be generated in polynomial time from H. We claim that it is a hitting set with threshold $1/2$ for co-nondeterministic circuits of size n'. Once we show this claim, we are done.

Indeed, take any co-nondeterministic circuit C' of size n' with n' inputs so that $|Z(C')| \le 2^{n'}/2$. We have to show that H' is not a subset of $Z(C')$. For this, construct a co-nondeterministic circuit C with n inputs as follows: $C(x) = 1$ iff $\exists y, (x,y) \in E_n \wedge C'(y)$. As the disperser is explicit, the size of this circuit can be made polynomial. We fix the constant q, so that n^q is an upper bound on its size. We claim $|Z(C)| < 2^{n^\delta}$: Otherwise, as G_n is a disperser, the neighbours in V_n of $Z(C)$ are more than half of V_n and thus the neighbours must intersect $V_n - Z(C')$, i.e., for some y adjacent to $x \in Z(C)$, $C'(y)$ is 1. But this implies $C(x) = 1$, contradicting $x \in Z(C)$. As $|Z(C)| < 2^{n^\delta}$, the acceptance probability of C is at least $1 - 2^{-n+n^\delta}$. Thus, as H is a hitting set, for some value $x \in H$, $C(x) = 1$. This means that for some $y \in V_n$, adjacent to some $x \in H$, $C'(y) = 1$. But such a y is by definition in H', i.e., H' hits C' as was to be shown. □

3. Simulating AM in NP

In this section we prove Theorem 1.5. We first prove Theorem 1.7 (stated in the Introduction) which shows how to construct a hitting set from the truth table of a hard function. We restate this theorem below.

Theorem 1.7. *For any $\epsilon > 0$ and $q \ge 1$, there is a polynomial time procedure P with the following properties. Let $f : \{0,1\}^m \to \{0,1\}$ be a function that cannot be computed by SV-nondeterministic circuits of size less than $2^{\epsilon m}$ for almost all m. Then there are constants $\delta = \delta(\epsilon) < \epsilon$ and $k > q$ such that, given the truth table of $f : \{0,1\}^{kl} \to \{0,1\}$ as input, P outputs a hitting set $H_f \subseteq \{0,1\}^n$ for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+n^\delta}$, where $n = (2l)2^{2l}$.*

Proof. We first show how to efficiently generate the set $H_f \subseteq \{0,1\}^n$ from the truth table for f and then argue that it has the right property. In the following, the values of k and δ, which depend on ϵ and q, will be fixed later. We assume, without loss of generality, that l is sufficiently large.

View f as a map $f : (\{0,1\}^l)^k \to \{0,1\}$. Now let $\mathbf{F}$ be the finite field with 2^{2l} elements. Identify $\mathbf{F}$ with $\{0,1\}^{2l}$ in any way that makes arithmetic efficient and embed $\{0,1\}^l$ in $\mathbf{F}$ by padding with zeros. Let the *low degree extension* (Babai *et al.* 1991) $\tilde{f} : \mathbf{F}^k \to \mathbf{F}$ of f be the unique polynomial with individual degree in each variable at most $2^l - 1$, agreeing with f on $(\{0,1\}^l)^k$.

Informally speaking, we define H_f as the set of tabulations of the restrictions of $\tilde{f}$ to every axis-parallel line in $\mathbf{F}^k$. More precisely, for $i \in \{1, \ldots, k\}$ and $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k \in \mathbf{F}$, let $v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ be the vector $(w_j)_{j \in \mathbf{F}}$ in $\mathbf{F}^{2^{2l}}$, with $w_j = \tilde{f}(a_1, \ldots, a_{i-1}, j, a_{i+1}, \ldots, a_k)$. As we have identified $\mathbf{F}$ with $\{0,1\}^{2l}$, we can also view $v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ as a bit string in $\{0,1\}^{(2l)2^{2l}} = \{0,1\}^n$.

With this in mind, now define $H_i \subseteq \{0,1\}^n$ as follows:

$$H_i = \{v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k) \mid a_1, \ldots, a_k \in \mathbf{F}\},$$

and let H_f be the union of all the H_is, that is,

$$H_f = \bigcup_{i=1}^{k} H_i.$$

First note that generating H_f from the truth table of f is a polynomial time procedure.

The structure of the proof that H_f is a hitting set with the desired properties is the following. We will suppose to the contrary that H_f is not such a hitting set, i.e., that it does not hit some circuit C. We will then show that f has a smaller SV-nondeterministic circuit than it is assumed to have. This will be done by making a *compressed* representation of $\tilde{f}$ which will have enough information to efficiently evaluate $\tilde{f}$ (and hence f) at any given point. The compressed representation is a table of the restriction of $\tilde{f}$ to S^k, for a small subset S of $\mathbf{F}$. The set S is carefully chosen, depending on the circuit C. Using the circuit C, we will be able to reconstruct $\tilde{f}$ at any desired point in $\mathbf{F}^k$ from its values in S^k.

Now we give a formal proof. Let $S \subseteq \mathbf{F}$ be a set of indices. Let π_S denote the projection function from any set of vectors in $\mathbf{F}^{|\mathbf{F}|}$ to $\mathbf{F}^S$.

We need to find a subset $S \subseteq \mathbf{F}$ so that the restriction of $\tilde{f}$ to S^k can be used to reconstruct $\tilde{f}$ at any desired point. The following lemma is used to find such a set. The proof of the lemma uses the *error-correcting* properties of low degree polynomials.

Lemma 3.1. *Let* $\mathbf{F}$ *be a finite field of size greater than* 2. *For any polynomial* p, *let* L_p *denote the vector* $(p(i))_{i\in\mathbf{F}}$ *in* $\mathbf{F}^{|\mathbf{F}|}$. *Let* $\mathcal{L} = \{L_p \mid p$ *is a polynomial of degree* $\leq |\mathbf{F}|^{1/2}\}$. *Then for any set* $Z \subset \mathbf{F}^{|\mathbf{F}|}$, *there is a set of indices* $S \subset \{1,\ldots,|\mathbf{F}|\}$ *with* $|S| \leq \lceil \log |Z| \rceil$ *such that the projection* $\pi_S : \mathcal{L} \cap Z \to \mathbf{F}^S$ *is* 1-1.

Proof. Any distinct $x, y \in \mathcal{L}$ are the same on less than $|\mathbf{F}|^{1/2}$ indices. Since $|\mathbf{F}| > 2$, this is less than 1/4 fraction of all the indices. Let $r = \lceil \log |Z| \rceil$. We can construct $S = \{j_1, \ldots, j_r\}$ in a greedy manner. Having chosen indices $S_i = \{j_1, \ldots j_i\}$, we construct $S_{i+1} = \{j_{i+1}\} \cup S_i$ as follows.

Let Y_i be the set of distinct pairs (x, y) such that x and y coincide on S_i. That is, $Y_i = \{(x, y) \mid x, y \in \mathcal{L} \cap Z,\ x \neq y$, and $\pi_{S_i}(x) = \pi_{S_i}(y)\}$. For any fixed $(x, y) \in Y_i$, the probability that x and y are different on a random index j in $\{1, \ldots, |\mathbf{F}|\} - S_i$ is at least 3/4. Hence, the expected number of pairs $(x, y) \in Y_i$ so that $x_j \neq y_j$ is at least $3|Y_i|/4$. So, by an averaging argument, we can find an index j_{i+1} such that $|Y_{i+1}| \leq |Y_i|/4$. Since $\binom{|Z|}{2}/4^r < 1$, the lemma follows. □

Now assume H_f is not a hitting set with the desired properties. Let C be a co-nondeterministic circuit establishing this, i.e., C maps $\{0,1\}^n$ to $\{0,1\}$, it has size n^q, and if we denote by Z the set $\{x \in \{0,1\}^n \mid C_i(x) = 0\}$, i.e., those x for which there is some setting of the nondeterministic choice gates making C evaluate to 0 on x, then $|Z| \leq 2^{n^\delta}$ and $H_f \subseteq Z$.

Let $\mathcal{L} \subseteq \mathbf{F}^{2^{2l}}$ be the vectors of the form $(p(i))_{i\in\mathbf{F}}$ for some univariate polynomial of degree less than 2^l. We can also view $\mathcal{L}$ as a subset of $\{0,1\}^n$. By construction, $H_f \subseteq \mathcal{L}$. Also by assumption $H_f \subseteq Z$. Hence by Lemma 3.1, there is a set S of indices of size $\leq n^\delta$ so that $\pi_S : \mathcal{L} \cap Z \to \mathbf{F}^S$ is 1-1. Fix this set S.

We will now construct a small SV-nondeterministic circuit for f. In fact, we will exhibit an efficient SV-nondeterministic procedure computing $\tilde{f}$ (and hence f) with the following nonuniform advice: the circuit C, the set S, and a table of the restriction of $\tilde{f}$ to S^k.

An overview of the procedure is the following. We need to compute $\tilde{f}$ on an arbitrary input $(a_1, \ldots, a_k)$. This is done in k stages. Let T_i denote the table of values $\tilde{f}(a_1, \ldots, a_i, S^{k-i})$. Then $T_k = \tilde{f}(a_1, \ldots, a_k)$ is the value we want to compute and $T_0 = \tilde{f}(S^k)$ is the table of the restriction of $\tilde{f}$ to S^k which is given as part of nonuniform advice. In the i^{th} stage, table $T_i = \tilde{f}(a_1, \ldots, a_i, S^{k-i})$ is reconstructed. This reconstruction procedure uses table T_{i-1}, the circuit C and the set of indices S as input. The procedure (Reconstruct) is described more formally below.

```
RECONSTRUCT($(a_1,\ldots,a_k), T_{i-1}, C, S$)

1 For all $u \in S^{k-i}$ do
2   Guess $v \in \mathbf{F}^{2l}$ /* A guess for the vector $(\tilde{f}(a_1,\ldots,a_{i-1},j,u))_{j\in\mathbf{F}}$*/
3   Verify the following
      a $v \in \mathcal{L}$ /*$v$ is the table of a polynomial of low degree */
      b $C(v) = 0$ /* By guessing a setting of the choice bits
                     of $C$ making the circuit evaluate to 0 on $v$ */
      c $\pi_S(v) = \pi_S(\tilde{f}(a_1,\ldots,a_{i-1},j,u))_{j\in\mathbf{F}}$ /* By looking up $T_{i-1}$ */
4   If all the verifications are correct, then include $v_{a_i}$ as the value of
    $\tilde{f}(a_1,\ldots,a_i,u)$ in $T_i$
```

Let v be a vector satisfying the conditions in Step 3 of the procedure. The verifications done in (a) and (b) guarantee that v is a vector in $\mathcal{L} \cap Z$. By Lemma 3.1, for all vectors $v' \in \mathcal{L} \cap Z$ other than v, their projection to the set of indices S is different. That is, $\pi_S(v) \neq \pi_S(v')$. Also, by the construction of the hitting set, $(\tilde{f}(a_1,\ldots,a_{i-1},j,u))_{j\in\mathbf{F}} \in \mathcal{L} \cap Z$. Hence verification done in (c) ensures that $v = (\tilde{f}(a_1,\ldots,a_{i-1},j,u))_{j\in\mathbf{F}}$ and hence $v_{a_i} = \tilde{f}(a_1,\ldots,a_i,u)$.

Let us estimate the size of the circuit for $\tilde{f}$ that can be built from the above procedure. In order to compute $\tilde{f}(a_1,\ldots,a_k)$, the procedure is called k times. For each such call, the time complexity of the above procedure is bounded by the time required to do less than $|S|^{k-1}$ verifications of a v-value. Each of these verifications takes the time of evaluating a circuit of size n^q, the time required to check that a table of size n is a low degree polynomial (which is bounded by n^2) and comparing $|S|$ values in $\mathbf{F}$ (which is bounded by n). Thus, building in the advice, we can convert the overall procedure into an SV-nondeterministic circuit. The size of the circuit is upper bounded by $O((n^\delta)^k n^q)$.

Now we can choose δ and k such that

$$O((n^\delta)^k n^q) < 2^{\epsilon k l}.$$

It is very easily seen that choosing $\delta = \epsilon/4$ allows us to choose a constant $k = 12q/\epsilon$, so that the above inequality is satisfied. □

Remark. It is not essential to use the low degree extension for our construction. What we essentially use is the fact that the error-correcting code we get from low degree multivariate polynomials is the *tensor product* of the low degree univariate polynomial codes. Therefore in our construction we can use the tensor product of any systematic code (codes which have the original message as part of the codeword) with parameters comparable to the low degree polynomial codes. See the survey paper by Miltersen (2001) for a construction based on tensor products of codes.

We can combine the above theorem with Lemma 2.4 to get Corollary 1.8.

Corollary 1.8. *For any constant $\epsilon > 0$, there is a constant $\tau > 0$ so that the following holds. There is a deterministic polynomial time procedure which, given as input the truth table of a Boolean function $f : \{0,1\}^m \to \{0,1\}$ (i.e., 2^m bits) with SV-nondeterministic circuit complexity at least $2^{\epsilon m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $1/2$ for co-nondeterministic circuits of size n, where $n = \lceil 2^{\tau m} \rceil$.*

Proof. Let $\epsilon > 0$ be given. Assume m is sufficiently large. Given a truth table on m inputs with SV-nondeterministic circuit complexity at least $2^{\epsilon m}$, pad this truth table with zeros to obtain a truth table on m' inputs so that m' is divisible by k. Since k is a constant (to be chosen later), m' is at most $2m$ for sufficiently large m. Hence the circuit complexity of the new truth table is at least $2^{(\epsilon/2)m'}$.

Applying Theorem 1.7 (with parameters $\delta = \epsilon/8$ and $k = 24q/\epsilon$), we have an efficient procedure for transforming this truth table into a hitting set in $\{0,1\}^{n'}$ with threshold $1 - 2^{-n' + n'^{\delta}}$ for co-nondeterministic circuits of size $(n')^q$, where $n' = (2m'/k)2^{2m'/k}$.

For the chosen value of δ, let γ be the constant in Lemma 2.4. Now choose $\tau = \gamma/k$ and apply Lemma 2.4 and deterministically convert this hitting set into a hitting set in $\{0,1\}^{n''}$ with threshold $1/2$ for co-nondeterministic circuits of size n'' with

$$n'' = \lceil ((2m'/k)2^{2m'/k})^{\gamma} \rceil \geq 2^{2m\gamma/k} \geq \lceil 2^{\tau m} \rceil.$$

Take this hitting set and remove the last $n'' - n$ bits in each string in it. This is the desired hitting set in $\{0,1\}^n$. □

Implications

Corollary 3.2. *For any constant $\epsilon > 0$, the following holds. If there exists a language L in $\mathbf{NE} \cap \mathbf{coNE}$ so that for all but finitely many n, $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\epsilon n}$, then there is an SVNP-procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $1/2$ for co-nondeterministic circuits of size n.*

Proof. Given ϵ, let τ be the corresponding constant of Corollary 1.8 and n be sufficiently large. On input 1^n, the SVNP-procedure computes $m =$

$\lceil \tau^{-1} \log n \rceil$ and enumerates the truth table of the characteristic function of L on $\{0,1\}^m$. Having found the truth table, it applies the procedure of Corollary 1.8 to it, yielding a hitting set in $\{0,1\}^{n'}$, where $n' = \lceil 2^{\tau m} \rceil = \lceil 2^{\tau \lceil \tau^{-1} \log n \rceil} \rceil$. Take this hitting set and remove the last $n' - n$ bits in each string in it. This is the desired hitting set in $\{0,1\}^n$. □

From Proposition 2.2 and Corollary 3.2 we have the derandomization result for **AM**.

Theorem 1.5. *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{NE} \cap \mathbf{coNE}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then* $\mathbf{AM} = \mathbf{NP}$.

As Graph Nonisomorphism is in **AM** (Goldreich *et al.* 1991) (and trivially in **coNP**), we have in particular the following corollary.

Corollary 3.3. *If for some $\epsilon > 0$, there exists a language $L \in \mathbf{NE} \cap \mathbf{coNE}$ so that for all but finitely many n, $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\epsilon n}$, then Graph Isomorphism is in* $\mathbf{NP} \cap \mathbf{coNP}$.

Corollary 1.8 also implies a derandomization of the class **BPP**, stated in Theorem 1.6. It is easy to see that, by a proof completely analogous to the proof of Corollary 3.2, we can get the following corollary.

Corollary 3.4. *For any constant $\epsilon > 0$, the following holds. If there is a language L in* **E** *so that the SV-nondeterministic circuit complexity of $L \cap \{0,1\}^n$ is at least $2^{\epsilon n}$ for all but finitely many n, then there is a polynomial time procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $1/2$ for circuits of size n.*

The following result of Andreev *et al.* (1998) (simplified by Andreev *et al.* 1997b; Buhrman & Fortnow 1999) gives a method to derandomize **BPP** using hitting sets.

Theorem 3.5 (Andreev–Clementi–Rolim). *If there is a polynomial time procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ with threshold $1/2$ for deterministic circuits of size n then* $\mathbf{BPP} = \mathbf{P}$.

Hence we obtain Theorem 1.6.

4. Relativizable hitting set generators are dispersers

The explicit construction of disperser with necessary parameters is a major technical tool that we employ in constructing our hitting set. In this section, we note that any *similar* construction of hitting sets has to appeal to the existence of explicit dispersers or itself provide such dispersers. More precisely, we note that any *relativizable, hardness-based* hitting sets are also dispersers with matching parameters.

We make it more formal for certain settings of parameters. Note that Corollary 1.8 and its proof relativize, i.e., the following statement has been proved.

COROLLARY 4.1. *For any $\epsilon > 0$, there is a $\delta > 0$ so that the following holds. There is a deterministic polynomial time procedure which, for any oracle A, has the following property. Given as input the truth table of a Boolean function $f : \{0,1\}^m \to \{0,1\}$ (i.e., 2^m bits) with SV-nondeterministic oracle circuit complexity with oracle gates for A at least $2^{\epsilon m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $1/2$ for co-nondeterministic oracle circuits of size n with oracle gates for A, where $n = \lceil 2^{\delta m} \rceil$.*

The hardest part of a self-contained proof of Corollary 4.1 is the existence of explicit dispersers. We now note that any proof of Corollary 4.1 *has* to appeal to the existence of explicit dispersers or itself provide such dispersers.

THEOREM 4.2. *Let a procedure with the property of Corollary 4.1 be given. Let $n = 2^m$ be sufficiently large. Define the bipartite graph $G_n = (U_n, V_n, E_n)$ with $U_n = \{0,1\}^n$, $V_n = \{0,1\}^{\lceil n^\delta \rceil}$, and an edge between x and y if and only if y is a member of the hitting set produced by the procedure on input x. Then G_n is a disperser with threshold $2^{n^{2\epsilon}}$.*

PROOF. We need to prove that, given any subset S of U_n with $|S| \geq 2^{n^{2\epsilon}}$, more than half the vertices of V_n are adjacent to S. Suppose not. Let S be a set for which this is not the case, and let A be the non-neighbours of S, so we have $|A| \geq |V_n|/2$. Viewed as a subset of $\{0,1\}^{\lceil n^\delta \rceil}$, we can use A as an oracle and consider circuit complexity relative to A. By Shannon's counting argument, viewed as truth tables for Boolean functions on n variables, at least one of the members of S must have SV-nondeterministic oracle circuit complexity with oracle gates for A at least $\frac{1}{2} \log |S| / \log \log |S| > n^\epsilon = 2^{\epsilon m}$. Let this element of S be denoted a. Thus, as the procedure has the property of Corollary 4.1, the vertices in V_n adjacent to a will intersect every set in V_n which

1. is the characteristic (accepted) set of an oracle circuit with oracle gates for A of size at most n and
2. has size at least $|V_n|/2$.

But then consider the oracle circuit defined by $x \to A(x)$. It has size n, its characteristic set has size at least $|V_n|/2$, and the neighbours of a do not intersect its characteristic set, as this set is the non-neighbours of S and $a \in S$. A contradiction.

□

5. Simulating AM in nondeterministic quasi-polynomial time

In this section we sketch how the techniques used to prove Theorem 1.5 can be directly applied to get other hardness-randomness tradeoffs for the class **AM**. In particular we show that if there are languages in **NEXP** $\cap$ **coNEXP** that require SV-nondeterministic circuit complexity $2^{n^{1/2+\epsilon}}$ for some $\epsilon > 0$, then **AM** is in nondeterministic quasi-polynomial time.

Let **NQuasiP** denote the class of problems that can be decided in nondeterministic quasi-polynomial time. That is, $L \in$ **NQuasiP** if there exists a nondeterministic machine accepting L which runs in time $2^{\log^c n}$ for some constant c. By SVNQuasiP-procedure we mean a single-valued nondeterministic procedure (defined formally in Section 2) running in time $2^{\log^c n}$ for some constant c.

We will show the following derandomization of **AM**.

Theorem 1.9. *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NEXP** $\cap$ **coNEXP** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{n^{1/2+\epsilon}}$, then* **AM** $\subseteq$ **NQuasiP**.

As in the proof of Theorem 1.5, we first construct a hitting set with very high threshold, and convert this into a hitting set of threshold 1/2 using explicitly constructed dispersers. The main difference from the proof of Theorem 1.5 is the choice of parameters for the dispersers. Here we need dispersers with much smaller threshold than those used in proving Lemma 2.4. In particular, we need, for any δ, explicit dispersers with threshold $2^{2^{\log^{\delta} n}}$. Since the proofs of most of the results required to prove Theorem 1.9 are very similar to the proofs of analogous statements that are given in previous sections, we only sketch them here.

Analogous to Proposition 2.2, we have the following proposition.

Proposition 5.1. *If there is an* SVNQuasiP*-procedure which on input* 1^n *outputs a hitting set in* $\{0,1\}^n$ *for co-nondeterministic circuits with threshold* $1/2$, *then* **AM** $\subseteq$ **NQuasiP**.

An explicit construction of dispersers with parameters necessary for us is given by Ta-Shma (2002).

Theorem 5.2 (Ta-Shma). *For any constant* $\delta > 0$ *there is a* $\gamma > 0$ *such that an explicit* $(2^{\log^\delta n}, 2^{\log^\gamma n})$*-disperser exists.*

Using the disperser with the above-mentioned parameters, it is easy to rework the proof of Lemma 2.3 to prove the following.

Lemma 5.3. *For any constant* $\delta > 0$, *there are constants* $\gamma > 0$ *and* $q \geq 1$ *such that the following holds. There is a polynomial time procedure which takes input* H, *where* $H \subseteq \{0,1\}^n$ *is a hitting set for co-nondeterministic circuits of size* n^q *with threshold* $1-2^{-n+2^{\log^\delta n}}$, *and outputs a hitting set* $H' \subseteq \{0,1\}^{n'}$ *with threshold* $1/2$ *for co-nondeterministic circuits of size* n', *where* $n' = \lceil 2^{\log^\gamma n} \rceil$.

Now we state the main theorem of this section which is analogous to Theorem 1.7.

Theorem 5.4. *For any* $\epsilon > 1/2$ *and* $q \geq 1$, *there is a polynomial time procedure* P *with the following properties. Let* $f : \{0,1\}^m \to \{0,1\}$ *be a function that cannot be computed by SV-nondeterministic circuits of size less than* 2^{m^ϵ} *for almost every* m. *Then there is a constant* $\delta = \delta(\epsilon) < \epsilon$ *and a* $k = k(l)$ *such that, given the truth table of* $f : \{0,1\}^{kl} \to \{0,1\}$ *as input,* P *outputs a hitting set* $H_f \subseteq \{0,1\}^n$ *for co-nondeterministic circuits of size* n^q *with threshold* $1 - 2^{-n+2^{\log^\delta n}}$, *where* $n = (2l)2^{2l}$.

Proof (sketch). The construction of the hitting set is identical to the construction given in Theorem 1.7. We have to choose k and δ appropriately. Now suppose the set H_f produced is not a hitting set for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+2^{\log^\delta n}}$. Then as in the proof of Theorem 1.7, we can construct an SV-nondeterministic circuit for f where the size of the circuit is $O((2^{\log^\delta n})^k(n^q))$. We will now choose δ and k such that $O((2^{\log^\delta n})^k(n^q)) < 2^{(kl)^\epsilon}$. An easy calculation shows that for large enough l, choosing $\delta = \frac{1}{2}(\epsilon - \frac{1}{2})$ allows us to choose $k = l$ so as to satisfy the inequality. □

The above result implies Theorem 1.9. We sketch the proof here.

Proof of Theorem 1.9 (sketch). We give an SVNQuasiP-procedure (say M) which, on input 1^m, outputs a hitting set in $\{0,1\}^m$ with threshold 1/2 for co-nondeterministic circuits of size m. Let f be the characteristic function of L.

For ϵ, let δ be the constant given by Theorem 5.4. For this δ, let γ be the constant given by Lemma 5.3. Let $l = \lceil \log^{1/\gamma} m \rceil$.

The SVQNP-procedure, on input 1^m, first enumerates the truth table of f on inputs of length l^2. Now M, viewing this as a truth table of $f : \{0,1\}^{kl} \to \{0,1\}$ (with $k = l$), simulates the deterministic procedure of Theorem 5.4 and produces a hitting set H_f in $\{0,1\}^{n'}$ for co-deterministic circuits of size $(n')^q$ with threshold $1 - 2^{-n'+2^{\log^\delta n'}}$, where $n' = (2l)2^{2l}$. Then M deterministically converts this hitting set H_f into a hitting set $H \subseteq \{0,1\}^{n''}$ with threshold 1/2 for co-nondeterministic circuits of size n'', where $n'' = \lceil 2^{\log^\gamma n'} \rceil \geq m$. Finally, M removes the last $n'' - m$ bits from each string in H to produce the required hitting set. Since $l^2 = \log^{O(1)} m$, and $L \in \mathbf{NEXP} \cap \mathbf{coNEXP}$, it follows that M is an SVQNP-procedure.

6. Final remarks

In addition to the derandomization of **AM**, Klivans & van Melkebeek (2002) gave several other applications of the fact that the Impagliazzo–Wigderson construction relativizes. Each of the applications showed that a hardness assumption involving oracle circuits implies a "derandomization" (in a loose sense).

For one of these extra applications we can combine their reasoning with Corollary 1.8 and obtain an improvement. Specifically, we can prove the following theorem relating two circuit lower bounds, which is identical to Theorem 5.15 in Klivans & van Melkebeek (2002), except that there the phrase, "SV-nondeterministic circuit complexity" is replaced with "oracle circuit complexity with oracle gates for SAT".

Theorem 6.1. *If there is a language L in $\mathbf{E}$ so that L has SV-nondeterministic circuit complexity at least $2^{\Omega(n)}$, then there exists a polynomially bounded function $p(n)$ and a polynomial time computable family of matrices M_n where M_n is an $n \times n$ matrix over $\mathbf{Z}_{p(n)}[x]$ such that the linear transformation defined by the family M_n cannot be computed by log-depth linear size circuits which have special gates that can compute binary linear operators over $\mathbf{Z}_{p(n)}[x]$.*

We omit the proof which is a straightforward combination of the proof of Theorem 5.15 in Klivans & van Melkebeek (2002) and Corollary 1.8 of the present paper.

7. Recent progress

Techniques in this paper work for functions in $\mathbf{NEXP} \cap \mathbf{coNEXP}$ with SV-nondeterministic circuit complexity at least 2^{n^δ} where $\delta > 1/2$. Derandomizing **AM** under weaker hardness assumptions was open. Shaltiel & Umans (2001) gave a construction which works for all ranges of parameters. In particular, they showed the "low-end" derandomization of **AM**: under the assumption that $\mathbf{NEXP} \cap \mathbf{coNEXP}$ has functions with super-polynomial SV-nondeterministic circuit complexity, $\mathbf{AM} \subseteq \mathbf{NSUBEXP}$. Later, Umans gave optimal constructions for all ranges of hardness parameters (Umans 2003). It is interesting to note that, like our construction, the constructions given in Shaltiel & Umans (2001) and Umans (2003) do not use *Nisan–Wigderson designs* which have been an important ingredient in most of the earlier derandomization results.

This paper uses nonuniform assumptions for derandomizing **AM**. Many derandomization results based on uniform assumptions have been obtained recently. Lu, using the terminology of pseudo-classes introduced by Kabanets (2001), established a certain kind of uniform derandomization for **AM** (Lu 2001). Impagliazzo, Kabanets & Wigderson (2002), under the assumption that $\mathbf{NEXP} \neq \mathbf{EXP}$, showed that **AM** can be simulated in nondeterministic subexponential time with sublinear advice. Gutfreund, Shaltiel & Ta-Shma (2003) observed a certain resilience property of our hitting set construction to prove a uniform "high-end" derandomization result for **AM**.

There are essentially three circuit models that have been used in the literature for derandomizing **AM** under nonuniform assumptions: SAT-oracle circuits (Klivans & van Melkebeek 2002), nondeterministic circuits (Arvind & Köbler 2001; Shaltiel & Umans 2001), and SV-nondeterministic circuits (used in this paper). It is clear from the definitions (given in Section 2) that an SV-nondeterministic circuit of size s that computes a function f can be easily converted into a nondeterministic circuit of size $O(s)$ that computes the same function. On the other hand, if complexity classes such as $\mathbf{NE} \cap \mathbf{coNE}$ or **E** which are closed under complement have nondeterministic circuits of size s, then they also have SV-nondeterministic circuits of size $O(s)$. A nondeterministic circuit can easily be seen as a restriction of a SAT-oracle circuit.

From the definitions it appears that the hardness assumption against SV-nondeterministic circuits is weaker than the corresponding hardness assumption against SAT-oracle circuits. A very recent work by Shaltiel & Umans (2004) showed that these hardness assumptions are essentially equivalent. In particular, they established a surprising result that if $\mathbf{NE} \cap \mathbf{coNE}$ has oracle circuits of size s which make *nonadaptive* oracle queries to SAT then $\mathbf{NE} \cap \mathbf{coNE}$ also has

SV-nondeterministic circuits of size $s^{O(1)}$. Since the proof of Theorem 1.4 given by Klivans & van Melkebeek (2002) goes through even under the assumption that there is a function in $\mathbf{NE} \cap \mathbf{coNE}$ that is hard against SAT-oracle circuits which make only nonadaptive queries, this result implies that the hardness assumption used by Klivans & van Melkebeek (2002) and the one used in this paper are essentially equivalent.

Finally, it is worth mentioning that in addition to results in Gutfreund *et al.* (2003) which use our construction, Barak, Ong & Vadhan (2003) applied our main result to obtain certain results in cryptography.

Acknowledgements

Most of the work was done while the second author was a postdoctoral fellow at BRICS, Department of Computer Science, University of Aarhus, Denmark. Both authors were supported by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT) and by BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation. The second author is supported by the NSF grant CCF-0430991 during the preparation of the final journal version.

The authors would like to thank Dieter van Melkebeek and Luca Trevisan for very helpful discussions.

References

A. E. Andreev, A. E. F. Clementi & J. D. P. Rolim (1996a). Hitting properties of hard Boolean operators and their consequences on BPP. Technical Report TR96-055, *Electronic Colloquium on Computational Complexity*. Available at `http://www.eccc.uni-trier.de/eccc`.

A. E. Andreev, A. E. F. Clementi & J. D. P. Rolim (1996b). Hitting sets derandomize BPP. In *Proc. 23rd International Colloquium on Automata, Languages and Programming*, Lecture Notes in Comput. Sci. 1099, Springer, 357–368.

A. E. Andreev, A. E. F. Clementi & J. D. P. Rolim (1997a). Worst-case hardness suffices for derandomization: a new method for hardness-randomness trade-offs. In *Proc. 24th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Comput. Sci. 1256, Springer, 177–187.

A. E. Andreev, A. E. F. Clementi, J. D. P. Rolim & L. Trevisan (1997b). Weak random sources, hitting sets, and BPP simulations. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, 264–272.

A. E. Andreev, A. E. F. Clementi & J. D. P. Rolim (1998). A new general derandomization method. *J. ACM* **45**, 179–213.

V. Arvind & J. Köbler (2001). On pseudorandomness and resource-bounded measure. *Theoret. Comput. Sci.* **255**, 205–221.

L. Babai (1985). Trading group theory for randomness. In *Proc. 17th ACM Symposium on Theory of Computing*, 421–429.

L. Babai (1992). Bounded round interactive proofs in finite groups. *SIAM J. Discrete Math.* **5**, 88–111.

L. Babai, L. Fortnow, L. A. Levin & M. Szegedy (1991). Checking computations in polylogarithmic time. In *Proc. 23rd ACM Symposium on Theory of Computing*, 21–31.

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

L. Babai, W. M. Kantor & E. M. Luks (1983). Computational complexity and the classification of finite simple groups. In *Proc. 24th IEEE Symposium on Foundations of Computer Science*, 162–171.

L. Babai & E. M. Luks (1983). Canonical labeling of graphs. In *Proc. 15th ACM Symposium on Theory of Computing*, 171–183.

L. Babai & S. Moran (1988). Arthur–Merlin games: a randomized proof system, and a hierarchy of complexity classes. *J. Comput. System Sci.* **36**, 254–276.

J. L. Balcázar, J. Díaz & J. Gabarró (1995). *Structural Complexity* II. EATCS Monogr. Theoret. Comput. Sci. 22, Springer.

B. Barak, S. J. Ong & S. P. Vadhan (2003). Derandomization in cryptography. In *Proc. 23rd International Cryptology Conference (CRYPTO 2003)*, Lecture Notes in Comput. Sci. 2729, Springer, 229–315.

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

H. Buhrman & L. Fortnow (1999). One-sided versus two-sided error in probabilistic computation. In *Proc. 16th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Comput. Sci. 1563, Springer, 100–109.

M. Fürer, O. Goldreich, Y. Mansour, M. Sipser & S. Zachos (1989). On completeness and soundness in interactive proof systems. In *Advances in Computing Research 5: Randomness and Computation*, S. Micali (ed.), 429–442.

O. Goldreich & L. A. Levin (1989). A hard-core predicate for all one-way functions. In *Proc. 21th ACM Symposium on Theory of Computing*, 25–32.

O. Goldreich, S. Micali & A. Wigderson (1991). Proofs that yield nothing but their validity or all languages in NP have zero knowledge proof systems. *J. ACM* **38**, 691–729.

O. Goldreich, S. Vadhan & A. Wigderson (2000). Simplified derandomization of BPP using a hitting set generator. Technical Report TR00-004, *Electronic Colloquium on Computational Complexity*. Available at `http://www.eccc.uni-trier.de/eccc`.

S. Goldwasser & M. Sipser (1989). Private coins versus public coins in interactive proof systems. In: *Advances in Computing Research 5*, 73–90.

D. Gutfreund, R. Shaltiel & A. Ta-Shma (2003). Uniform hardness vs. randomness tradeoffs for Arthur–Merlin games. In *Proc. 18th IEEE Conference of Computational Complexity*, 33–47.

R. Impagliazzo (1995). Hard-core distributions for somewhat hard problems. In *Proc. 36th IEEE Symposium on Foundations of Computer Science*, 538–547.

R. Impagliazzo, V. Kabanets & A. Wigderson (2002). In search of an easy witness: exponential time vs. probabilistic polynomial time. *J. Comput. System Sci.* **65**, 672–694.

R. Impagliazzo, R. Shaltiel & A. Wigderson (1999). Near-optimal conversion of hardness into pseudo-randomness. In *Proc. 40th IEEE Symposium on Foundations of Computer Science*, 181–190.

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

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

V. Kabanets (2001). Easiness assumptions and hardness tests: trading time for zero error. *J. Comput. System Sci.* **63**, 236–252.

A. R. Klivans & D. van Melkebeek (2002). Graph Nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.* **31**, 1501–1526.

C.-J. Lu (2001). Derandomizing Arthur–Merlin games under uniform assumptions. *Comput. Complexity* **10**, 247–259.

D. van Melkebeek (1998). Derandomizing Arthur–Merlin games. Technical Report TR-98-08, The University of Chicago, Department of Computer Science.

P. B. Miltersen (1998). Error correcting codes, perfect hashing circuits, and deterministic dynamic dictionaries. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 556–563.

P. B. Miltersen (2001). Derandomizing complexity classes. In *Handbook of Randomized Computing*, Kluwer, 843–941.

P. B. Miltersen & N. V. Vinodchandran (1999). Derandomizing Arthur–Merlin games using hitting sets. In *Proc. 40th IEEE Symposium on Foundations of Computer Science*, 71–80.

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

C. Papadimitriou (1994). *Computational Complexity*. Addison-Wesley.

M. Saks, A. Srinivasan & S. H. Zhou (1998). Explicit OR-dispersers with polylogarithmic degree. *J. ACM* **45**, 123–154.

A. L. Selman (1996). Much ado about functions. In *Proc. 11th IEEE Conference on Computational Complexity*, 198–212.

R. Shaltiel (2002). Recent developments in extractors. *Bull. Europ. Assoc. Theoret. Comput. Sci.* **77**, 67–95.

R. Shaltiel & C. Umans (2001). Simple extractors for all minentropies and a new pseudo-random generator. In *Proc. 41st IEEE Symposium on Foundations of Computer Science*, 648–657.

R. Shaltiel & C. Umans (2004). Pseudorandomness for approximate counting and sampling. Technical Report TR04-84, *Electronic Colloquium on Computational Complexity*. Available at `http://www.eccc.uni-trier.de/eccc`.

M. Sipser (1988). Expanders, randomness, or time versus space. *J. Comput. System Sci.* **36**, 379–383.

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

A. Ta-Shma (2002). Almost optimal dispersers. *Combinatorica* **22**, 123–145.

A. Ta-Shma, C. Umans & D. Zuckerman (2001). Loss-less condensers, unbalanced expanders, and extractors. In *Proc. 33rd ACM Symposium on Theory of Computing*, 143–152.

L. Trevisan (2001). Extractors and pseudorandom generators. *J. ACM* **48**, 860–879.

C. Umans (2003). Pseudo-random generators for all hardnesses. *J. Comput. System Sci.* **67**, 419–440.

C. B. Wilson (1985). Relativized circuit complexity. *J. Comput. System Sci.* **31**, 169–181.

A. C. Yao (1982). Theory and application of trapdoor functions. In *Proc. 23rd IEEE Symposium on Foundations of Computer Science*, 80–91.

Manuscript received 24 May 2004

Peter Bro Miltersen
Department of Computer Science
University of Aarhus
IT-parken, Aabogade 34
DK-8200 Aarhus N, Denmark
bromille@daimi.au.dk

N. V. Vinodchandran
Department of Computer
Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588, U.S.A.
vinod@cse.unl.edu