Basic Research in Computer Science

Derandomizing Arthur-Merlin Games using Hitting Sets

Peter Bro Miltersen
Vinodchandran N. Variyam

BRICS Report Series **RS-99-47**

ISSN 0909-0878 **December 1999**

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/47/`

Derandomizing Arthur-Merlin games using hitting sets

Peter Bro Miltersen N.V. Vinodchandran

BRICS[*]
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark
{bromille,vinod}@brics.dk

Abstract

We prove that **AM** (and hence Graph Nonisomorphism) is in **NP** if for some $\epsilon > 0$, some language in **NE** $\cap$ **coNE** requires nondeterministic circuits of size $2^{\epsilon n}$. This improves recent results of Arvind and Köbler and of Klivans and Van Melkebeek who proved the same conclusion, but under stronger hardness assumptions, namely, either the existence of a language in **NE** $\cap$ **coNE** which cannot be *approximated* by nondeterministic circuits of size less than $2^{\epsilon n}$ or the existence of a language in **NE** $\cap$ **coNE** which requires *oracle circuits* of size $2^{\epsilon n}$ with oracle gates for SAT (satisfiability).

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 (if the existence of explicit dispersers can be assumed known) 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.

[*]Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

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 non-uniform 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 (Nisan-Wigderson) *Let $\epsilon > 0$ be any constant. If a language L in* **E** *exists, so that any circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $\frac{1}{2} + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, for all sufficiently large n, then* $\mathbf{P} = \mathbf{BPP}$.

The hardness assumption in Theorem 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 and Wigderson [IW97]. They showed in 1996 the following improvement of Theorem 1.

Theorem 2 (Impagliazzo-Wigderson) *Let $\epsilon > 0$ be any constant. If a language L in* **E** *exists so that $L \cap \{0,1\}^n$ has circuit complexity at least $2^{\epsilon n}$ for all sufficiently large n, then* **P**=**BPP**.

The proof of this theorem is technical and is built on the results of many earlier papers, including [BM84, Yao82, NW94, GL89, BFNW93, Imp95].

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 and Babai and Moran in [Bab85, BM88], 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** [Bab85, BM88, GMW91, GS89, Bab92]. 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 **AM**=**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 [BL83, BKL83].

In [AK97], Arvind and Köbler showed that the construction of [NW94] can be extended to the nondeterministic setting to get pseudorandom generators which can be used to completely derandomize **AM**. As in the case of [NW94], they needed an average-case hardness assumption in order to construct the generator. To be precise, Arvind and Köbler show

Theorem 3 (Arvind-Köbler) *Let $\epsilon > 0$ be any constant. If a language L in* **NE** $\cap$ **coNE** *exists*[1]*, so that any nondeterministic circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $\frac{1}{2} + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, for all sufficiently large n, then* **AM** = **NP**.

Recently, Klivans and Van Melkebeek [KvM99] constructed generators for derandomizing **AM** under a worst-case hardness assumption. The main observation they make is that the proof of Impagliazzo and Wigderson *relativizes*. This leads to the following theorem.

Theorem 4 (Klivans-Van Melkebeek) *Let $\epsilon > 0$ be any constant. If a language L in* **NE** $\cap$ **coNE** *exists so that $L \cap \{0,1\}^n$ has oracle circuit complexity at least $2^{\epsilon n}$ for all sufficiently large n with oracle gates for* SAT*, then* **AM**=**NP**.

Here, *oracle circuits* are Boolean circuits which contain special gates called *oracle* gates. These oracle gates are of unbounded fanin (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 and Köbler [AK97] and Van Melkebeek [vM98] asked whether **AM**=**NP** follows from the existence of a language in **NE**$\cap$**coNE** which does not have subexponential nondeterministic circuit complexity. In this paper, we answer this question affirmatively, proving the following theorem which improves Theorem 3 as well as Theorem 4.

Theorem 5 *Let $\epsilon > 0$ be any constant. If a language L in* **NE** $\cap$ **coNE** *exists so that $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$ for all sufficiently large n, then* **AM**=**NP**.

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

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, and 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 non-uniform and exponential analogue of $\mathbf{NP} \cap \mathbf{coNP}$, while oracle circuits with SAT as oracle form a non-uniform and exponential analogue of $\mathbf{P}^{\mathbf{NP}}$.

Our approach to proving Theorem 5 is completely different from the techniques of Arvind and Köbler and of Klivans and Van Melkebeek. Instead of using pseudorandom generators, we use a strengthened version of the *hitting set generator* approach to derandomization, due to Andreev, Clementi and Rolim [ACR97]. They gave, independently and almost simultaneously to Impagliazzo and Wigderson's work, two different conditions, each implying **P=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 **P=BPP** (it is easy to show that it implies **P=RP**). The latter part of the proof, i.e., the fact that the existence of the hitting set generator is enough to show **P=BPP** was proved already by Andreev, Clementi and Rolim in 1995 [ACR96b, ACR98]. Since then, the proof of this implication was simplified enormously [ACRT97, BF99].

It was (and is still not) clear if the Andreev-Clementi-Rolim 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 6 *Let $\epsilon > 0$ be any constant. If there is a language L in* **E** *so that $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$ for all sufficiently large n then* **P**= **BPP**.

Note that this differs from the Impagliazzo-Wigderson theorem "only" in the assumption being about SV-nondeterministic circuits, rather than about deterministic ones. But our proof is simpler than any known proof of the Impagliazzo-Wigderson theorem [IW97, STV99].

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 7 *For any constants $\epsilon > 0$ and $k \geq q \geq 2$, there is a polynomial time procedure with the following properties. Given as input the truth table of a function $f : \{0,1\}^m \to \{0,1\}$, so that k divides m, it outputs a subset H_f of $\{0,1\}^n$, where $n = (2m/k)2^{2m/k}$, so that for all f,* if *f cannot be computed by SV-nondeterministic circuits of size less than $2^{3(\epsilon+q/k)m}$,* then *H_f is a hitting set in $\{0,1\}^n$ with threshold $1 - 2^{-n+n^\epsilon}$ for co-nondeterministic circuits of size n^q.*

The main ingredient we add to the techniques of Andreev, Clementi and Rolim to prove Theorem 7 is to first replace f by its *low degree extension* [BFLS91]. An intuitive reason why this turns out to be useful is as follows: The technique of Andreev, Clementi and Rolim is based on *compression* in the form of hashing. As was previously noted by the first author [Mil98], hashing becomes a much easier and cleaner operation when applied to data, encoded in an error-correcting code. The low degree extension performs such an encoding for us. 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 by Andreev, Clementi and Rolim. It is interesting to note that making a low degree extension is also the first step in the Impagliazzo-Wigderson generator. However, the central fact about the extension they use (and which is used in general in the constructions of pseudorandom generators) is that it yields a *locally decodable* error correcting code. Essentially, we only use that it yields an error correcting code and need not concern ourselves with decoding.

While the above intuition was useful for coming up with the proof of Theorem 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 7, we combine it with a variation of a lemma from [ACR96a] (Lemma 12 of the present paper), and prove

Corollary 8 *For any constant $\tau > 0$, there is a constant $\gamma > 0$ 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 $2^{\tau m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n, where $n = \lceil 2^{\gamma m} \rceil$.*

This corollary then immediately implies Theorem 5. To prove Theorem 6, we use the result of [ACR98] as simplified by [ACRT97, BF99], stating that the hitting set generator of Corollary 8 derandomizes **BPP**.

The proof of the lemma of [ACR96a] uses the existence of explicit *dispersers*. The first construction of explicit dispersers with the necessary parameters were by Saks, Srinivasan and Zhou [SSZ95]. The simplest to date is by Trevisan [Tre99]; so to get the simplest possible self-contained proof of Theorem 5 and 6 one should use Trevisan's construction. His construction actually yields an extractor, a stronger notion than a disperser. However, we would like to emphasise that explicit dispersers are sufficient for our proof, as it is conceivable that the future will bring (even) simpler constructions of explicit dispersers which may not be extractors.

The existence of explicit dispersers is the only moderately heavy tool we have to bring in to prove Theorems 5 and 6. But, we would like to point out that any *relativizable* proof of Corollary 8 (such as ours) *has* to use the existence of explicit dispersers. Indeed, any algorithm with the property of Corollary 8 *itself* defines a disperser. In short: *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 of [Tre99], where the analogous statement *Any relativizable worst case hardness-based pseudorandom generator defines an extractor* is implicitly established. We make the formal statement of the hitting set/disperser correspondence with a self-contained proof as Theorem 17 in Section 5.

The fact that relativizable hitting set generators are dispersers gives an alternate way of viewing Theorem 7: It is a *shell* we can put before any

disperser. It will preserve the disperser property (with slightly weaker parameters) but also strengthen the disperser so that it obtains the properties of a relativizable hitting set generator. Thus, while Theorem 17 tells us that every relativizable hitting set generator is an explicit disperser, Theorem 7 tells us that every explicit disperser can be easily converted into a relativizable hitting set generator.

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 textbook [BDG90, Pap94] complexity theoretic notation and definitions, such as the definitions of standard complexity classes like **P**, **NP**, **E**, **NE**, and **BPP**. 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 $x \in \{0,1\}^n$,

$$x \in L \Rightarrow \Pr_{y \in \{0,1\}^{p(n)}}(\exists z \in \{0,1\}^{p(n)}\ (x,y,z) \in L') = 1$$

$$x \notin L \Rightarrow \Pr_{y \in \{0,1\}^{p(n)}}(\exists z \in \{0,1\}^{p(n)}\ (x,y,z) \in L') \leq \frac{1}{2}$$

An SVNP-procedure (SV meaning *Single Valued* [Sel96]) 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 $f(x)$.

Circuits

A *nondeterministic* Boolean circuit C contains, in addition to 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

[2]The original definition in [Bab85] of **AM** is a two-sided error version. But it is shown in [FGM+89] that this definition is equivalent to the one-sided error version, which we give here.

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 [Wil85] are Boolean circuits with special gates called *oracle* gates. These oracle gates can be of arbitrary fanin, 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 $\epsilon, \delta > 0$ and $k \geq 1$, an *explicit* (ϵ, δ)-disperser is a family of dispersers $G_n = (U_n, V_n, E_n), n = 1, 2, \ldots$ with $|U_n| = \{0,1\}^n, |V_n| = \{0,1\}^{\lceil n^\delta \rceil}$, and threshold $t_n = 2^{n^\epsilon}$ 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 was given by Saks, Srinivasan and Zhou in [SSZ95]. A construction with better parameters was given by Ta-Shma [TS98] and a simpler one was given by Trevisan in [Tre99]. For the theorem below, the original result by Saks, Srinivasan and Zhou suffices, though the proof by Trevisan is easier.

Theorem 9 (Saks-Srinivasan-Zhou) *For any $\epsilon > 0$, there is a $\delta > 0$, so that an explicit (ϵ, δ)-disperser exists.*

Hitting sets

A *hitting set* in $\{0,1\}^n$ 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 10 *If there is an SVNP-procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n then $\mathbf{AM} = \mathbf{NP}$.*

Now we state a lemma from [ACR96a][3]. Actually, the lemma is implicit already in [Sip88]. It shows that in fact it is sufficient to construct hitting sets with much bigger threshold than $\frac{1}{2}$. This lemma is a consequence of the existence of explicit dispersers. Indeed, in [Sip88], it was Sipser's motivation for defining the notion of a disperser.

Lemma 11 (Sipser, Andreev-Clementi-Rolim) *For any constant $\epsilon > 0$, there are constants $q \geq 1$ and $\delta > 0$ 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 $1 - 2^{-n+n^\epsilon}$ for circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $\frac{1}{2}$ for circuits of size n', where $n' = \lceil n^\delta \rceil$.*

What we actually need, is the analogous Lemma for co-nondeterministic circuits. This lemma is proved exactly as Lemma 11, 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 12 *For any constant $\epsilon > 0$, there are constants $q \geq 1$ and $\delta > 0$ 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 $1 - 2^{-n+n^\epsilon}$ for co-nondeterministic circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n', where $n' = \lceil n^\delta \rceil$.*

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

Proof Let $\epsilon > 0$ be fixed. According to Theorem 9, there is a δ, so that an explicit (ϵ, δ)-disperser exists. Let $G_n = (U_n, V_n, E_n)$ be this disperser, i.e., with $n' = \lceil n^\delta \rceil$, $U_n = \{0,1\}^n$, $V_n = \{0,1\}^{n'}$, and for all subsets S of U_n of size at least 2^{n^ϵ}, more than half the vertices of V are adjacent to S.

Let $H \subseteq \{0,1\}^n$ be a hitting set with threshold $1 - 2^{-n+n^\epsilon}$ 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 $\frac{1}{2}$ 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 $|Z(C')| \leq \frac{2^{n'}}{2}$. We must 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 $|Z(C)| < 2^{n^\epsilon}$: Otherwise, as G_n is a disperser, the neighbors in V_n of $Z(C)$ is more than half of V_n and thus the neighbors 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 $|Z(C)| < 2^{n^\epsilon}$, the acceptance probability of C is at least $1 - 2^{-n+n^\epsilon}$. 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 Proof of the main theorem

Recall our main theorem.

Theorem 7 *For any $\epsilon > 0$ and $k \geq q \geq 2$, there is a polynomial time procedure with the following properties. Given as input the truth table of a function $f : \{0,1\}^m \to \{0,1\}$, so that k divides m, it outputs a subset H_f of $\{0,1\}^n$, where $n = (2m/k)2^{2m/k}$, so that for all f,* if *f cannot be computed by SV-nondeterministic circuits of size less than $2^{3(\epsilon+q/k)m}$,* then *H_f is a hitting set for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+n^\epsilon}$*

Proof We first show how to efficiently generate the set H_f from the truth table for f and then we argue that it has the right property. We assume, without loss of generality, that m is sufficiently large.

We can view f as a map $f : (\{0,1\}^{m/k})^k \to \{0,1\}$. Now let $\mathbf{F}$ be the finite field with $2^{2m/k}$ elements. Identify $\mathbf{F}$ with $\{0,1\}^{2m/k}$ in any way that makes arithmetic efficient and embed $\{0,1\}^{m/k}$ in $\mathbf{F} = \{0,1\}^{2m/k}$ by padding with zeros.

Let the *low degree extension* [BFLS91] $\tilde{f} : \mathbf{F}^k \to \mathbf{F}$ of f be the unique polynomial with individual degree in each variable at most $2^{m/k} - 1$, agreeing with f on $(\{0,1\}^{m/k})^k$.

Now we define the set H_f. Informally speaking, H_f is 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 $a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k \in \mathbf{F}$, let $v_i(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ be the vector $(w_j)_{j \in \mathbf{F}}$ in $\mathbf{F}^{2^{2m/k}}$, with $w_j = \tilde{f}(a_1, a_2, \ldots, a_{i-1}, j, a_{i+1}, \ldots, a_k)$. As we have identified $\mathbf{F}$ with $\{0,1\}^{2m/k}$, we can also view $v_i(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ as a bit string in $\{0,1\}^{(2m/k)2^{2m/k}} = \{0,1\}^n$.

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

$$H_i = \{v_i(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k) | a_1, \ldots, a_k \in \mathbf{F}\}$$

and

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

First note that generating H_f from the truth table of f is a polynomial time procedure (as the size of the input is 2^m).

We should now show that H_f is a hitting set for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+n^\epsilon}$. We first informally outline the proof. 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 smaller circuits than it is assumed to have. This will be done by making a *compressed* representation of $\tilde{f}$ which can be used as non-uniform advice 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 $\mathbf{F}$. Using the circuit C, we will be able to reconstruct $\tilde{f}$ on any desired point in $\mathbf{F}^k$ from its values in S^k.

Now the proof. Assume H_f is not the desired hitting set. 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 by Z denote $\{x \in \{0,1\}^n | C_i(x) = 0\}$, i.e., those x for which there is some setting of the nondeterministic choice gates making C evaluate to 0 on x, then $|Z| \leq 2^{n^\epsilon}$ and $H_f \subseteq Z$.

Let $L \subseteq \{0,1\}^n = \{0,1\}^{(2m/k)2^{2m/k}} = \mathbf{F}^{2^{2m/k}}$ be the vectors of the form $(p(i))_{i \in \mathbf{F}}$ for some univariate polynomial of degree less than $2^{m/k}$. By construction, $H_f \subseteq L$.

Viewed as vectors in $\mathbf{F}^{2^{2m/k}}$, any two distinct elements x, y of L are the same on less than $2^{m/k}$ indices. For sufficiently large m, this is less than a $\frac{1}{4}$ fraction of all the indices. We will argue that there is a subset $S \subseteq \{1, \ldots, 2^{2m/k}\}$ of indices of size $|S| \leq n^\epsilon$ so that the projection $\pi_S : \mathbf{F}^{2^{2m/k}} \to \mathbf{F}^{|S|}$ to the indices of S is 1-1 when restricted to $Z \cap L$.

Indeed we can construct S in a greedy way as follows. Let $s = |Z \cap L|$. We choose indices $x_1, x_2, \ldots, x_r$ in $\{1, \ldots, 2^{2m/k}\}$, with $r = \lceil \log s \rceil$, so that if $S_i := \{x_1, \ldots, x_i\}$, the projection π_{S_i} makes at most $\binom{s}{2}/4^i$ unordered pairs from $Z \cap L$ collide. Since $\binom{s}{2}/4^r < 1$, we can let $S = S_r$. To construct x_{i+1}, having already constructed $x_1, \ldots, x_i$, we argue as follows: For a fixed pair of vectors which collide under π_{S_i}, a random choice of x_{i+1} will separate the pair with probability at least $\frac{3}{4}$. Thus, there is a fixed choice of x_{i+1} which will leave at most $\frac{1}{4}$ of the pairs unseparated.

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 non-uniform advice:

- The circuit C,
- the set S, and
- a table of the restriction of $\tilde{f}$ to S^k.

On input $(a_1, a_2, \ldots, a_k)$, to compute $\tilde{f}(a_1, a_2, \ldots, a_k)$ the procedure does as follows. For every element $u \in S^{k-1}$ it *guesses* the vector $v = (\tilde{f}(j, u))_{j \in \mathbf{F}}$. It checks that its guess is correct by

- Checking that v is the table of a low degree polynomial (i.e., checking that $v \in L$)
- Checking that $C(v) = 0$, by guessing a setting of the choice bits of C making the circuit evaluate to 0 on v (i.e, checking that $v \in Z$).

- Checking that the entries of v corresponding to indices in S are correct, by consulting the table for $\tilde{f}$, restricted to S^k. (i.e. checking that $\pi_S(v)$ has the right value.)

By construction, the value $v = (\tilde{f}(j,u))_{j\in\mathbf{F}}$ is the only value with these properties.

After having ensured that v is the correct value, it keeps the value $\tilde{f}(a_1,u)$ and throws away the rest of the vector v.

Having done this for every possible u, the procedure has now built a table of $\tilde{f}$, restricted to $\{a_1\}\times S^{k-1}$.

Now, for every element $u \in S^{k-2}$ it guesses the vector $v = (\tilde{f}(a_1,j,u))_{j\in\mathbf{F}}$. It checks that each guess is correct by checking that v is the table of a low degree polynomial, checking that $C(v) = 0$ and checking that the entries of v corresponding to indices in S are correct, by consulting the table for $\tilde{f}$, restricted to $\{a_1\}\times S^{k-1}$. After having ensured that v is the correct value, it keeps the value $\tilde{f}(a_1,a_2,u)$ and throws away the rest of the vector v. Having done this for every possible u, the procedure has now built a table of $\tilde{f}$, restricted to $\{a_1\}\times\{a_2\}\times S^{k-2}$.

Now it goes through a similar loop for every element $u \in S^{k-3}$, building a table of $\tilde{f}$, restricted to $\{a_1\}\times\{a_2\}\times\{a_3\}\times S^{k-3}$, and so on, until in the end it has the value of $\tilde{f}(a_1,a_2,\ldots,a_k)$ which was the value we wanted.

The time complexity of the above procedure is bounded by the time required to do less than $k|S|^{k-1}$ verifications of a v-value, each of these verifications taking the time of evaluating a circuit of size n^q (plus the time required to check that a table of size n is a low degree polynomial, which is bounded by n^2). Thus, converting the procedure into a circuit, building in the advice, we get an SV-nondeterministic circuit of size at most

$$O((n^\epsilon)^k n^q) < 2^{3(\epsilon+q/k)m}$$

for sufficiently large m. As a circuit for $\tilde{f}$ can also be used as a circuit for f, this contradicts the assumption on f. □

Combining with Lemma 12, we get

Corollary 8 *For any constant $\tau > 0$, there is a constant $\gamma > 0$ 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 $2^{\tau m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n, where $n = \lceil 2^{\gamma m}\rceil$.*

Proof Let $\tau > 0$ be given. Let $\epsilon = \tau/12$. Let q, δ be the constants of Lemma 12, guaranteed to exist for this ϵ. Let $k = 12\tau^{-1}q$. Let $\gamma = \delta/k$.

Assume m is sufficiently large. Given a truth table on m inputs with SV-nondeterministic circuit complexity at least $2^{\tau m}$, pad this truth table with zeros to obtain a truth table on m' inputs so that m' is divisible by k. The circuit complexity of the new truth table is at least $2^{(\tau/2)m'}$, as m is sufficiently large.

Applying Theorem 5 with the parameters ϵ, q, k and noticing that $\tau/2 \geq 3(\epsilon + q/k)$, we have an efficient procedure transforming this truth table into a hitting set in $\{0,1\}^n$ with threshold $1 - 2^{-n+n^\epsilon}$ for co-nondeterministic circuits of size $(n')^q$, where $n' = (2m'/k)2^{2m'/k}$.

Now apply Lemma 12 and deterministically convert this hitting set into a hitting set in $\{0,1\}^{n''}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n'' with

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

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

4 Implications

We derandomize **AM**.

Corollary 13 *For any constant $\tau > 0$, the following holds. If some language L in* **NE** $\cap$ **coNE** *exists, so that $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\tau n}$ for all sufficiently large n, then there is an SVNP-procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n.*

Proof Given τ, let γ be the corresponding constant of Corollary 8. On input 1^n, with n sufficiently large, the SVNP-procedure computes $m = \lceil \gamma^{-1} \log n \rceil$ 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 8 to it, yielding a hitting set in $\{0,1\}^{n'}$, where $n' = \lceil 2^{\gamma m} \rceil = \lceil 2^{\gamma \lceil \gamma^{-1} \log n \rceil} \rceil$. 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 10 and Corollary 13 we have the derandomization result for **AM**.

Theorem 5 *Let $\epsilon > 0$ be any constant. If a language L in* **NE** $\cap$ **coNE** *exists so that $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$ for all sufficiently large n, then* **AM**=**NP**.

As graph non-isomorphism is in **AM** [GMW91] (and trivially in **coNP**), we have in particular the following corollary.

Corollary 14 *If for some $\epsilon > 0$, there is a language $L \in$* **NE**$\cap$**coNE** *so that $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\epsilon n}$ for all sufficiently large n, then Graph Isomorphism is in* **NP** $\cap$ **coNP**.

By a proof completely analogous to the proof of Corollary 13 (and thus omitted), we have

Corollary 15 *For any constant $\tau > 0$, 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 $2^{\tau n}$ for all sufficiently large n, then there is a polynomial time procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $\frac{1}{2}$ for circuits of size n.*

Combining Corollary 15 with the fact that the hitting sets produced in this corollary are sufficient to derandomize **BPP** [ACR98, ACRT97, BF99], we obtain Theorem 6.

5 Relativizable hitting set generators are dispersers

Corollary 8 was proved by combining our main theorem with Lemma 12; the latter using the existence of explicit dispersers. Corollary 8 and its proof relativizes, i.e., the following statement has been proved.

Corollary 16 *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 $\frac{1}{2}$ 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 16 is the existence of explicit dispersers. We now note that any proof of Corollary 16 *has* to appeal to the existence of explicit dispersers or itself provide such dispersers.

Theorem 17 *Let a procedure with the property of Corollary 16 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-neighbors 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 16, 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 neighbors of a do not intersect its characteristic set, as this set is the non-neighbors of S and $a \in S$. A contradiction. □

6 Final Remarks

In addition to the derandomization of **AM**, Klivans and Van Melkebeek [KvM99] had 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 8 and obtain an improvement. Specifically, we can prove the following theorem relating two circuit lower bounds, which is identical to Theorem 5.15, [KvM99], except that there, "SV-nondeterministic circuit complexity" is replaced with "oracle circuit complexity with oracle gates for SAT".

Theorem 18 *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 and Van Melkebeek and Corollary 8 of the present paper.

7 Acknowledgement

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.

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

References

[ACR96a] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Hitting properties of hard boolean operators and their consequences on BPP. In *Electronic Colloquium on Computational Complexity, technical reports*. 1996. TR96-055, Avilable at `http://www.eccc.uni-trier.de/eccc`.

[ACR96b] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Hitting sets derandomize BPP. In *Automata, Languages and Programming, 23rd International Colloquium*, volume 1099

of *Lecture Notes in Computer Science*, pages 357–368, Paderborn, Germany, 8–12 July 1996. Springer-Verlag.

[ACR97] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Worst-case hardness suffices for derandomization: A new method for hardness-randomness trade-offs. In *Automata, Languages and Programming, 24th International Colloquium*, volume 1256 of *Lecture Notes in Computer Science*, pages 177–187, Bologna, Italy, 7–11 July 1997. Springer-Verlag.

[ACR98] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. A new general derandomization method. *Journal of the ACM*, 45(1):179–213, January 1998.

[ACRT97] Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and BPP simulations. In *38th Annual Symposium on Foundations of Computer Science*, pages 264–272, Miami Beach, Florida, 20–22 October 1997. IEEE.

[AK97] V. Arvind and Johannes Köbler. On resource-bounded measure and pseudorandomness. *Lecture Notes in Computer Science*, 1346:235–249, 1997.

[Bab85] László Babai. Trading group theory for randomness. In *Proc. 17th Ann. ACM Symp. on Theory of Computing*, pages 421–429, 1985.

[Bab92] László Babai. Bounded round interactive proofs in finite groups. *SIAM Journal on Discrete Mathematics*, 5(1):88–111, February 1992.

[BDG90] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity* II. Number 22 in EATCS Monographs on Theoretical Computer Science. Springer, Berlin-Heidelberg-New York, 1990.

[BF99] Harry Buhrman and Lance Fortnow. One-sided versus two-sided error in probabilistic computation. In *Theoretical Aspects of Computer Science, 16th Annual Symposium*, volume 1563 of *Lecture Notes in Computer Science*, pages 100–109, Trier, Germany, 4–6 March 1999. Springer-Verlag.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing*, pages 21–31. ACM Press, May 1991.

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

[BKL83] László Babai, William M. Kantor, and Eugene M. Luks. Computational complexity and the classification of finite simple groups. In *24th Annual Symposium on Foundations of Computer Science*, pages 162–171, Los Alamitos, Ca., USA, November 1983. IEEE Computer Society Press.

[BL83] László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 171–183, Boston, Massachusetts, 25–27 April 1983.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

[BM88] László Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[FGM+89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. In S. Micali, editor, *Advances in Computing Research 5: Randomness and Computation*, pages 429–442. JAI Press, Greenwich, CT, 1989.

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

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero

knowledge prrof systems. *Journal of the Association for Computing Machinery*, 38(3):691–729, 1991.

[GS89] Shafi Goldwasser and Michael Sipser. *Private coins versus public coins in interactive proof systems*, volume 5 of *Advances in Computing Research*, pages 73–90. 1989.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science*, pages 538–547, Los Alamitos, October 1995. IEEE Computer Society Press.

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

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

[Mil98] Peter Bro Miltersen. Error correcting codes, perfect hashing circuits, and deterministic dynamic dictionaries. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 556–563, San Francisco, California, 25–27 January 1998.

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

[Pap94] Christopher Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Sel96] Alen L. Selman. Much ado about functions. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 198–212, 1996.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36:379–383, 1988.

[SSZ95] Michael Saks, Aravind Srinivasan, and Shiyu Zhou. Explicit dispersers with polylog degree. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 479–488, Las Vegas, Nevada, 29 May–1 June 1995.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. In *31st ACM Symposium on Theory of Computing*, pages 537–546, 1999.

[Tre99] Luca Trevisan. Constructions of near-optimal extractros using pseudo-random generators. In *31st ACM Symposium on Theory of Computing*, pages 141–148, Atlanta, Georgia, 1999.

[TS98] Amnon Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 196–202, 1998.

[vM98] Dieter van Melkebeek. Derandomizing Arthur-Merlin games. Technical Report TR-98-08, The University of Chicago, Department of Computer Science, July 1998.

[Wil85] Chrisropher B. Wilson. Reltivized circuit complexity. *Journal of Computer and System Sciences*, 31(2):169–181, 1985.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In IEEE, editor, *23rd annual Symposium on Foundations of Computer Science, November 3–5, 1982, Chicago, IL*, pages 80–91, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1982. IEEE Computer Society Press.

Recent BRICS Report Series Publications

RS-99-47 Peter Bro Miltersen and Vinodchandran N. Variyam. *Derandomizing Arthur-Merlin Games using Hitting Sets*. December 1999. 21 pp. Appears in Beame, editor, *40th Annual Symposium on Foundations of Computer Science*, FOCS '99 Proceedings, 1999, pages 71–80.

RS-99-46 Peter Bro Miltersen, Vinodchandran N. Variyam, and Osamu Watanabe. *Super-Polynomial Versus Half-Exponential Circuit Size in the Exponential Hierarchy*. December 1999. 14 pp. Appears in Asano, Imai, Lee, Nakano and Tokuyama, editors, *Computing and Combinatorics: 5th Annual International Conference*, COCOON '99 Proceedings, LNCS 1627, 1999, pages 210–220.

RS-99-45 Torben Amtoft. *Partial Evaluation for Designing Efficient Algorithms—A Case Study*. December 1999.

RS-99-44 Uwe Nestmann, Hans Hüttel, Josva Kleist, and Massimo Merro. *Aliasing Models for Mobile Objects*. December 1999. To appear in a special FOOL '99 issue of *Information and Computation*.

RS-99-43 Uwe Nestmann. *What Is a 'Good' Encoding of Guarded Choice?* December 1999. To appear in a special EXPRESS '97 issue of *Information and Computation*. This revised report supersedes the earlier BRICS report RS-97-45.

RS-99-42 Uwe Nestmann and Benjamin C. Pierce. *Decoding Choice Encodings*. December 1999. To appear in *Journal of Information and Computation*.

RS-99-41 Nicky O. Bodentien, Jacob Vestergaard, Jakob Friis, Kåre J. Kristoffersen, and Kim G. Larsen. *Verification of State/Event Systems by Quotienting*. December 1999. 17 pp. Presented at *Nordic Workshop in Programming Theory*, Uppsala, Sweden, October 6–8, 1999.

RS-99-40 Bernd Grobauer and Zhe Yang. *The Second Futamura Projection for Type-Directed Partial Evaluation*. November 1999. Extended version of an article to appear in Lawall, editor, *ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, PEPM '00 Proceedings, 2000.