Uniform Derandomization from Pathetic Lower Bounds

Eric Allender*
Department of Computer Science
Rutgers University
New Brunswick, NJ 08855, USA
allender@cs.rutgers.edu

V Arvind
The Institute of Mathematical Sciences
C.I.T. Campus
Chennai 600 113, India
arvind@imsc.res.in

Fengming Wang†
Department of Computer Science
Rutgers University
New Brunswick, NJ, 08855 USA
fengming@cs.rutgers.edu

July 15, 2010

Abstract

A recurring theme in the literature on derandomization is that probabilistic algorithms can be simulated quickly by deterministic algorithms, if one can obtain *impressive* (i.e., superpolynomial, or even nearly-exponential) circuit size lower bounds for certain problems. In contrast to what is needed for derandomization, existing lower bounds seem rather pathetic (linear-size lower bounds for general circuits [IM02], nearly cubic lower bounds for formula size [Hås98], nearly $n \log\log n$ size lower bounds for branching programs [BSSV03], n^{1+c_d} for depth d threshold circuits [IPS97]). Here, we present two instances where "pathetic" lower bounds of the form $n^{1+\epsilon}$ would suffice to derandomize interesting classes of probabilistic algorithms.

We show:

- If the word problem over S_5 requires constant-depth threshold circuits of size $n^{1+\epsilon}$ for some $\epsilon > 0$, then any language accepted by uniform polynomial-size probabilistic threshold circuits can be solved in subexponential time (and more strongly, can be accepted by a uniform family of deterministic constant-depth threshold circuits of subexponential size.)
- If there are no constant-depth arithmetic circuits of size $n^{1+\epsilon}$ for the problem of multiplying a sequence of n 3-by-3 matrices, then for every constant d, black-box identity testing for depth-d arithmetic circuits with bounded individual degree can be performed in subexponential time (and even by a uniform family of deterministic constant-depth AC^0 circuits of subexponential size).

*Supported in part by NSF Grants DMS-0652582, CCF-0830133, and CCF-0832787. Some of this work was performed while this author was a visiting scholar at the University of Cape Town.

†Supported in part by NSF Grants CCF-0830133 and CCF-0832787.

1 Introduction

Hardness-based derandomization is one of the success stories of the past quarter century. The main thread of this line of research dates back to the work of Shamir, Yao, and Blum and Micali [Sha81, Yao82, BM84], and involves showing that, if given a suitably hard function f, one can construct pseudorandom generators and hitting-set generators. Much of the progress on this front over the years has involved showing how to weaken the hardness assumption on f and still obtain useful derandomizations [BFNW93], [AK97], [IW97], [IW01], [KvM02], [ACR99], [ACR98], [ACRT99], [BF99], [MV05], [GW99], [GVW00], [ISW06], [STV01], [SU05], [Uma03]. In rare instances, it has been possible to obtain *unconditional* derandomizations using this framework; Nisan and Wigderson showed that uniform families of probabilistic AC^0 circuits can be simulated by uniform deterministic AC^0 circuits of size $n^{\log^{O(1)} n}$ [NW94]. More often, the derandomizations that have been obtained are conditional, and rely on the existence of functions f that are hard on average. For certain large complexity classes $\mathcal{C}$ (notably including $\#\mathsf{P}, \mathsf{PSPACE}$, and exponential time), various types of random self-reducibility and hardness amplification have been employed to show that such hard-on-average functions f exist in $\mathcal{C}$ if and only if there is some problem in $\mathcal{C}$ that requires large Boolean circuits [BFNW93, IW97].

A more recent thread in the derandomization literature has studied the implications of *arithmetic* circuit lower bounds for derandomization. Kabanets and Impagliazzo showed that, if the Permanent requires large *arithmetic circuits*, then the probabilistic algorithm to test if two arithmetic *formulae* (or more generally, two arithmetic circuits of polynomial degree) are equivalent can be simulated by a quick deterministic algorithm [KI04]. Subsequently, Dvir, Shpilka, and Yehudayoff built on the techniques of Kabanets and Impagliazzo, to show that if one could present a multilinear polynomial (such as the permanent) that requires depth d arithmetic formulae of size 2^{n^ϵ}, then the probabilistic algorithm to test if two arithmetic circuits of depth $d-5$ are equivalent (where in addition, the variables in these circuits have degree at most $\log^{O(1)} n$) can be derandomized to obtain a $2^{\log^{O(1)} n}$ deterministic algorithm for the problem.

In this paper, we combine these two threads of derandomization with the recent insight that, in some cases, extremely modest-sounding (or even "pathetic") lower bounds can be amplified to obtain superpolynomial bounds [AK10]. In order to carry out this combination, we need to identify and exploit some special properties of certain functions in and near NC^1.

- The word problem over S_5 is one of the standard complete problems for NC^1 [Bar89]. Many of the most familiar complete problems for NC^1 have very efficient *strong downward self-reductions* [AK10]. We show that the word problem over S_5, in addition, is *randomly self-reducible*. (This was observed previously by Goldwasser *et al.* [GGH$^+$07].) This enables us to transform a "pathetic" *worst-case* size lower bound of $n^{1+\epsilon}$ on constant-depth threshold circuits, to a superpolynomial size *average-case* lower bound for this class of circuits. In turn, by making some adjustments to the Nisan-Wigderson generator, this average-case hard function can be used to give uniform subexponential derandomizations of probabilistic TC^0 circuits.

- Iterated Multiplication of n three-by-three matrices is a multilinear polynomial that is complete for arithmetic NC^1 [BOC92]. In the Boolean setting, this function is strongly downward self-reducible via self-reductions computable in TC^0 [AK10]. Here we show that there is a corresponding *arithmetic* self-reduction; this enables us to amplify a lower bound of size $n^{1+\epsilon}$ for constant-depth arithmetic circuits, to obtain a superpolynomial lower bound for constant-depth arithmetic circuits. Then, by building on the approach of Dvir *et al.* [DSY09], we are able to obtain subexponential derandomizations of the identity testing problem for a class of constant-depth arithmetic circuits.

The rest of the paper is organized as follows: In Section 2 we give the preliminary definitions and notation. In Section 3 we convert a modest worst-case hardness assumption to a strong average-case hardness separation of NC^1 from TC^0, and in Section 4 we use this to give a uniform derandomization of probabilistic TC^0 circuits. Finally, in Section 5 we prove our derandomization of a special case of polynomial identity testing under a modest hardness assumption.

2 Preliminaries

This paper will mainly discuss NC^1 and its subclass TC^0. The languages in NC^1 are accepted by families of circuits of depth $O(\log n)$ that are built with fan-in two AND and OR gates, and NOT gates of fan-in one. For any function $s(n)$, $\mathrm{TC}^0(s(n))$ consists of languages that are decided by constant-depth circuit families of size at most $s(n)$ which contain only unbounded fan-in MAJORITY gates as well as unary NOT gates. $\mathrm{TC}^0 = \cup_{k\geq 0}\mathrm{TC}^0(n^k)$. $\mathrm{TC}^0(\mathsf{SUBEXP}) = \cap_{\delta\geq 0}\mathrm{TC}^0(2^{n^\delta})$. The definitions of $\mathrm{AC}^0(s(n)), \mathrm{AC}^0$, and $\mathrm{AC}^0(\mathsf{SUBEXP})$ are similar, although MAJORITY gates are not allowed, and unbounded fan-in AND and OR gates are used instead.

As is usual in arguments in derandomization based on the hardness of some function f, we require not only that f not have small circuits in order to be considered "hard", but furthermore we require that f needs large circuits at *every* relevant input length. This motivates the following definition.

Definition 1 *Let A be a language, and let D_A be the set $\{n : A \cap \Sigma^n \neq \emptyset\}$. We say that $A \in$ io-$\mathrm{TC}^0(s(n))$ if there is an infinite set $I \subseteq D_A$ and a language $B \in \mathrm{TC}^0(s(n))$ such that, for all $n \in I$, $A_n = B_n$ (where, for a language C, we let C_n denote the set of all strings of length n in C). Similarly, we define io-TC^0 to be $\cup_{k\geq 0}$io-$\mathrm{TC}^0(n^k)$.*

Thus A requires large threshold circuits on *all* relevant input lengths if $A \notin$ io-TC^0. (A peculiarity of this definition is that if A is a *finite* set, or A^n is empty for infinitely many n, then $A \notin$ io-TC^0. This differs starkly from most notions of "io" circuit complexity that have been considered, but it allows us to consider "complex" sets A that are empty on infinitely many input lengths; the alternative would be to consider artificial variants of the "complex" sets that we construct, having strings of every length.)

Probabilistic circuits take an input divided into two pieces, the actual input and the random coin flips. We say an input x is accepted by such a circuit C if, with respect to the uniform distribution U_R over coin flips, $Pr_{r\sim U_R}[C(x,r)=1] \geq \frac{2}{3}$ while x is rejected by C if $Pr_{r\sim U_R}[C(x,r)=1] \leq \frac{1}{3}$.

The standard uniformity condition for small complexity classes is called $\mathsf{DLOGTIME}$-uniformity. In order to provide its proper definition, we need to mention the direct connection language associated with a circuit family.

Definition 2 *Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family. The direct connection language L_{DC} of $\mathcal{C}$ is the set of all tuples having either the form $\langle n,p,q,b\rangle$ or $\langle n,p,d\rangle$, where*

- *If $q = \epsilon$, then b is the type of gate p in C_n;*
- *If q is the binary encoding of k, then b is the kth input to p in C_n.*
- *The gate p has fan-in d in C_n.*

The circuit family $\mathcal{C}$ is $\mathsf{DLOGTIME}$-uniform if there is a deterministic Turing machine that accepts L_{DC} in linear time. For any circuit complexity class C, uC is its uniform counterpart, consisting of languages that are accepted by $\mathsf{DLOGTIME}$-uniform circuit families. For more background on circuit complexity, we refer the reader to the textbook by Vollmer [Vol99]. The term "uniform derandomization" in the title refers to the fact that we are presenting uniform circuit families that compute derandomized algorithms; this should not be confused with doing derandomization based on uniform hardness assumptions.

A particularly important complete language for NC^1 is the word problem WP for S_5, where S_5 is the symmetric group over 5 distinct elements [Bar89]. The input to the word problem is a sequence of permutations from S_5 and it is accepted if and only if the product of the sequence evaluates to the identity permutation. The corresponding *search* problem FWP is required to output the exact result of the iterated multiplication. A closely related *balanced* language is BWP, which stands for Balanced Word Problem.

Definition 3 *The input to* BWP *is a pair* $\langle w_1 w_2..w_n, S\rangle$*, where* $\forall i \in [1..n],\ w_i \in S_5$*,* $S \subseteq S_5$ *and* $|S| = 60$*. The pair* $\langle w_1 w_2..w_n, S\rangle$ *is in* BWP *if and only if* $\Pi_{i=1}^n w_i \in S$*.*

It is easy to verify that BWP is complete for NC^1 as well.

In the following sections, let FWP_n be the sub-problem of FWP where the domain is restricted to inputs of length n and let BWP_n be $\mathrm{BWP} \cap \{\langle \phi, S\rangle \mid \phi \in S_5^n,\ |\phi| = n,\ S \subseteq S_5,\ |S| = 60\}$. Note that BWP_n accepts exactly half of the instances in $\{\langle \phi, S\rangle \mid \phi \in S_5^n,\ |\phi| = n,\ S \subseteq S_5,\ |S| = 60\}$ since $|S_5| = 120$.

The following simplified version of Chernoff's bound turns out to be useful in our application.

Lemma 4 (Chernoff's bound) *Let* $X_1, .., X_m$ *be i.i.d. 0-1 random variables with* $E[X_i] = p$*. Let* $X = \Sigma_{i=1}^n X_i$*. Then for any* $0 < \delta \leq 1$*,*

$$Pr[X < (1-\delta)pm] \leq e^{-\frac{\delta^2 pm}{2}}.$$

3 The existence of an average-case hard language

In this section, we use random self-reducibility to show that, if $\mathrm{NC}^1 \neq \mathrm{TC}^0$, then there are problems in NC^1 that are hard on average for TC^0. First we recall the definition of hardness on average for decision problems.

Definition 5 *Let* U_D *denote the uniform distribution over all inputs in a finite domain* D*. For any Boolean function* $f : D \rightarrow \{0,1\}$*,* f *is* $(1-\epsilon)$*-hard for a set of circuits* S*, if, for every* $C \in S$*, we have that* $Pr_{x\sim U_D}[f(x) = C(x)] < 1-\epsilon$*.*

We will sometimes abuse notation by identifying a set with its characteristic function. For languages to be considered hard on average, we consider only those input lengths where the language contains some strings.

Definition 6 *Let* Σ *be an alphabet. Consider a language* $L = \cup_n L_n$*, where* $L_n = L \cap \Sigma^n$*, and let* $D_L = \{n : L_n \neq \emptyset\}$*. We say that* L *is* $(1-\epsilon)$*-hard for a class of circuit families* $\mathcal{C}$ *if* D_L *is an infinite set and, for any circuit family* $\{C_n\}$ *in* $\mathcal{C}$*, there exists* m_0 *such that for all* $m \in D_L$ *such that* $m \geq m_0$*,* $Pr_{x\in\Sigma^m}[f(x) = C(x)] < 1-\epsilon$*.*

The following theorem shows that if $\mathrm{FWP} \notin \text{io-}\mathrm{TC}^0$, then BWP is hard on average for TC^0.

Theorem 7 *There exist constants* $c, \delta > 0$ *and* $0 < \epsilon < 1$ *such that for any constant* $d > 0$*, if* FWP_n *is not computable by* $\mathrm{TC}^0(\delta n(s(n) + cn))$ *circuits of depth at most* $d + c$*, then* BWP_n *is* $(1-\epsilon)$*-hard for* TC^0 *circuits of size* $s(n)$ *and depth* d*.*

Proof. Let $\epsilon < \frac{1}{4\binom{120}{60}}$. We prove the contrapositive. Assume there is a circuit C of size $s(n)$ and depth d such that $Pr_x[\mathrm{BWP}_n(x) = C(x)] \geq 1-\epsilon$. We first present a probabilistic algorithm for FWP_n.

Let the input instance for FWP_n be $w_1 w_2 \ldots w_n$. Generate a sequence of $n+1$ random permutations $u_0, u_1, \ldots, u_n$ in S_5 and a random set $S \subseteq S_5$ of size 60. Let ϕ be the sequence $(u_0 \cdot w_1 \cdot u_1)(u_1^{-1} \cdot w_2 \cdot u_2)..(u_{n-1}^{-1} \cdot w_n \cdot u_n)$. Note that ϕ is a completely random sequence in S_5^n.

Let us say that ϕ is a "good" sequence if $\forall S' \subset S_5$ with $|S'| = 60,\ C(\langle \phi, S'\rangle) = \mathrm{BWP}_n(\langle \phi, S'\rangle)$.

If we have a "good" sequence ϕ (meaning that for *every* set S' of size 60, C gives the "correct" answer $\mathrm{BWP}_n(\phi, S)$ on input (ϕ, S')), then we can easily find the unique value r that is equal to $\Pi_{i=1}^n \phi_i$ where $\phi_i = u_{i-1} w_i u_i$, as follows:

- If $C(\phi, S) = 1$, then it must be the case that $r \in S$. Pick any element $r' \in S_5 \setminus S$ and observe that r is the only element such that $C(\phi, (S \setminus \{r\}) \cup \{r'\}) = 0$.
- If $C(\phi, S) = 0$, then it must be the case that $r \notin S$. Pick any element $r' \in S$ and observe that r is the only element such that $C(\phi, (S \setminus \{r'\}) \cup \{r\}) = 1$.

Thus the correct value r can be found by trying all such r'. Hence, if ϕ is good, we have

$$r = \Pi_{i=1}^n \phi_i = u_0 w_1 u_1 \Pi_{i=2}^n u_{i-1}^{-1} w_i u_i.$$

Produce as output the value $u_0^{-1} r u_n^{-1} = \Pi_{i=1}^n w_i = \mathrm{FWP}_n(w)$.

Since $\epsilon < \frac{1}{4\binom{120}{60}}$, a standard averaging argument shows that at least $\frac{3}{4}$ of the sequences in S_5^n are good. Thus with probability at least $\frac{3}{4}$, the probabilistic algorithm computes FWP_n correctly. The algorithm can be computed by a threshold circuit of depth $d+O(1)$ since the subroutines related to C can be invoked in parallel and moreover, the preparation of ϕ and the aggregation of results of subroutines can be done by constant-depth threshold circuits. Its size is at most $122s(n) + O(n)$ since there are 122 calls to C. Next, we put $10^4 n$ independent copies together in parallel and output the majority vote. Let X_i be the random variable that the outcome of the ith copy is $\Pi_{i=1}^n w_i$. By Lemma 4, on every input the new circuit computes FWP_n with probability at least $1 - \frac{120^{-n}}{2}$. Thus there is a random sequence that can be hardwired in to the circuit, with the property that the resulting circuit gives the correct output on *every* input (and in fact, at least half of the random sequences have this property). This yields a deterministic TC^0 circuit computing FWP_n exactly which is of depth at most $d + c$ and of size no more than $(122 * 10^4)n(s(n) + cn)$ for some universal constant c . Choosing $\delta \geq (122 * 10^4)$ completes the proof. □

The problem FWP is strongly downward self-reducible [AK10, Definition , Proposition 7]. Hence, its worst-case hardness against TC^0 circuit families can be amplified as observed by Allender and Koucký [AK10, Corollary 17].

Theorem 8 *[AK10] If there is a $\gamma > 0$ such that* $\mathrm{FWP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, *then* $\mathrm{FWP} \notin \text{io-}\mathrm{TC}^0$.

(Theorem 8 is not stated in terms of io-TC^0 in [AK10], but the proof shows that if there are infinitely many input lengths n where FWP has circuits of of size n^k, then there are infinitely many input lengths m where FWP has circuits of size $m^{1+\gamma}$. The strong downward self-reducibility property allows small circuits for inputs of size m to be constructed by efficiently using circuits for size $n < m$ as subcomponents.)

Since FWP is equivalent to WP via linear-size reductions on the same input length, the following corollary is its easy consequence.

Corollary 9 *If there is a $\gamma > 0$ such that* $\mathrm{WP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, *then* $\mathrm{FWP} \notin \text{io-}\mathrm{TC}^0$.

Combining Corollary 9 with Theorem 7 yields the average-case hardness of BWP from nearly-linear-size worst-case lower bounds for WP against TC^0 circuit families.

Corollary 10 *There exists a constant $\epsilon > 0$ such that if $\exists \gamma > 0$ such that* $\mathrm{WP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, *then for any k and d there exists $n_0 > 0$ such that when $n \geq n_0$,* BWP_n *is $(1-\epsilon)$-hard for any TC^0 circuit of size n^k and depth d.*

Define the following Boolean function $\mathrm{WPM}_n : S^n \times S^{60} \to \{0, 1\}$, where WPM_n stands for Word Problem over Multi-set.

Definition 11 *The input to* WPM_n *is a pair $\langle w_1 w_2..w_n, v_1 v_2..v_{60} \rangle$, where $\forall i \in [1..n]$, $w_i \in S_5$ and $\forall j \in [1..60]$, $v_i \in S_5$. $\langle w_1 w_2..w_n, v_1 v_2..v_{60} \rangle \in$* WPM *if and only if $\exists j \in [1..60]$, $\Pi_{i=1}^n w_i = v_j$.*

BWP is the restriction of WPM_n to the case where all v_is are distinct. Hence, WPM inherits the average-case hardness of BWP, since any circuit that computes WPM_n on a sufficiently large fraction of inputs also approximates BWP well. Formally,

Lemma 12 *There is an absolute constant $0 < c < 1$ such that for every $\epsilon > 0$, if BWP_n is $(1-\epsilon)$-hard for TC^0 circuits of size n^k and depth d, then WPM_n is $(1-c\epsilon)$-hard for TC^0 circuits of size n^k and depth d.*

Proof. Let $c = \frac{\binom{120}{60}}{(120)^{60}}$. Note that c is the probability that a sequence of 60 permutations contains no duplicates and is in sorted order. Suppose there is a circuit C with the property that $Pr_{x \in S^n \times S^{60}}[C(x) \neq \mathrm{WPM}(x)] \leq c\epsilon$. Then the conditional probability that $C(x) \neq \mathrm{WPM}(x)$ given that the last 60 items in x give a list in sorted order with no duplicates is at most ϵ. This yields a circuit having the same size, solving BWP with error at most ϵ, using the uniform distribution over its domain, contrary to our assumption. □

Corollary 13 *There exists a constant $\epsilon > 0$ such that if $\exists \gamma > 0$ such that $\mathrm{WP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, then for any k and d there exists $n_0 > 0$ such that when $n \geq n_0$, WPM_n is $(1-\epsilon)$-hard for TC^0 circuits of size n^k and depth d.*

Yao's XOR lemma [Yao82] is a powerful tool to boost average-case hardness. We utilize a specialized version of the XOR lemma for our purpose. Several proofs of this useful result have been published. For instance, see the text by Arora and Barak [AB09] for a proof that is based on Impagliazzo's hardcore lemma [Imp95]. For our application here, we need a version of the XOR lemma that is slightly different from the statement given by Arora and Barak. In the statement of the lemma as given by them, g is a function of the form $\{0,1\}^n \to \{0,1\}$. However, their proof works for any Boolean function g defined over any finite alphabet, because both the hardcore lemma and its application in the proof of the XOR lemma are insensitive to the encoding of the alphabet. Hence, we state the XOR Lemma in terms of functions over an alphabet set Σ.

For any Boolean function g over some domain Σ^n, define $g^{\oplus m} : \Sigma^{nm} \to \{0,1\}$ by $g^{\oplus m}(x_1, x_2, .., x_m) = g(x_1) \oplus g(x_2) \oplus .. \oplus g(x_m)$ where $\oplus$ represents the parity function.

Lemma 14 *[Yao82] Let $\frac{1}{2} < \epsilon < 1$, $k \in \mathbb{N}$ and $\theta > 2(1-\epsilon)^k$. There is a constant $c > 1$ that depends only on $|\Sigma|$ such that if g is $(1-\epsilon)$-hard for TC^0 circuits of size s and depth d, then $g^{\oplus k}$ is $(\frac{1}{2}+\theta)$-hard for TC^0 circuits of size $\frac{\theta^2 s}{cn}$ and depth $d-1$.*

Let $\Sigma = S_5$. The following corollary is an immediate consequence of Corollary 13 and Lemma 14.

Corollary 15 *If there is a $\gamma > 0$ such that $\mathrm{WP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, then for any k, k' and d there exists $n_0 > 0$ such that when $n \geq n_0$ $(\mathrm{WPM}_n)^{\oplus n}$ is $(\frac{1}{2} + \frac{1}{n^{k'}})$-hard for TC^0 circuits of size n^k and depth d.*

Let $\mathrm{WP}^{\otimes} = \cup_{n \geq 1}\{x \mid (\mathrm{WPM}_n)^{\oplus n}(x) = 1\}$. Note that it is a language in uNC^1 and, moreover, it is decidable in linear time.

Theorem 16 *If there is a $\gamma > 0$ such that $\mathrm{WP} \notin \text{io-}\mathrm{TC}^0(n^{1+\gamma})$, then for any integer $k > 0$, $\mathrm{WP}^{\otimes}$ is $(\frac{1}{2} + \frac{1}{n^k})$-hard for TC^0.*

4 Uniform derandomization

The Nisan-Wigderson generator is the canonical method to prove the existence of pseudo-random generators based on hard functions. It relies on the following definition of combinatorial designs.

Definition 17 (Combinatorial Designs) *Fix a universe of size u. An (m,l)-design of size n on $[u]$ is a list of subsets $S_1, S_2, ..., S_n$ satisfying:*

1. $\forall i \in [1..n],\ |S_i| = m$;
2. $\forall i \neq j \in [1..n],\ |S_i \cap S_j| \leq l$.

Nisan and Wigderson [NW94] invented a general approach to construct combinatorial designs for various ranges of parameters. The proof given by Nisan and Wigderson gives designs where $l = \log n$, and most applications have used that value of l. For our application, l can be considerably smaller, and furthermore, we need the S_i's to be very efficiently computable. For completeness, we present the details here. (Other variants of the Nisan-Wigderson construction have been developed for different settings; we refer the reader to one such construction by Viola [Vio05], as well as to a survey of related work [Vio05, Remark 5.3].)

Lemma 18 *[vL99] For $l > 0$, the polynomial $x^{2\cdot 3^l} + x^{3^l} + 1$ is irreducible over $\mathbb{F}_2[x]$.*

Lemma 19 *[NW94] For any integer n, any α such that $\log\log n/\log n < \alpha < 1$, let $b = \lceil \alpha^{-1} \rceil$ and $m = \lceil n^\alpha \rceil$, there is a (m,b)-design with $u = O(m^6)$. Furthermore, each S_i can be computed within $O(bm^2)$ time.*

Proof. Fix $q = 2^{2\cdot 3^l}$ for some l such that $m \leq q \leq m^3$. Let the universe be $\mathbb{F}_q \times \mathbb{F}_q$ and S_i be the graph of the ith univariate polynomial of degree at most b in the standard order. Since $q^b \geq (n^\alpha)^b \geq n$, there are at least n distinct S_is. No two polynomials share more than b points, hence, the second condition is satisfied. The first condition holds because we could simply drop elements without increasing the size of intersections.

The arithmetic operations in $\mathbb{F}_q$ are performed within $\log^{O(1)} q$ time because of the explicitness of the irreducible polynomial by Lemma 18. It is evident that for any $i \in [n]$, we are able to enumerate all elements of S_i in time $O(m \cdot b(\log^{O(1)} q)) = O(bm^2)$. □

Lemma 20 *For any constant $\alpha > 0$ and for any large enough integer n, if g is $(\frac{1}{2} + \frac{1}{n^2})$-hard for TC^0 circuits of size n^2 and depth $d+2$, then any probabilistic TC^0 circuit C of size n and depth d can be simulated by another probabilistic TC^0 circuit of size $O(n^{1+\alpha})$ and depth $d+1$ which is given oracle access to $g_{\lceil n^\alpha \rceil}$ and uses at most $O(n^{6\alpha})$ many random bits.*

Proof. This is a direct consequence of Lemma 19; we adapt the traditional Nisan-Wigderson argument to the setting of TC^0 circuits. Let n and α be given, with $0 < \alpha < 1$. Let $S_1, \ldots, S_n$ be the (m,b)-design from Lemma 19, where $m = \lceil n^\alpha \rceil$, $b = \lceil \alpha^{-1} \rceil$, and each $S_i \subset [u]$, with $u = O(m^6)$. We are given $g : \Sigma^m \to \{0,1\}$; define $h^g : \Sigma^u \to \{0,1\}^n$ by $h^g(x) = g(x|_{S_1})g(x|_{S_2})..g(x|_{S_n})$, where $x|_{S_i}$ is the sub-sequence restricted to the coordinates specified by S_i.

The new circuit samples randomness uniformly from A^u and feeds C with pseudo-random bits generated by h^g instead of purely random bits. It only has one more extra layer of oracle gates and its size is bounded by $O(n + n * n^\alpha) = O(n^{1+\alpha})$. What is left is to prove the following claim.

Claim 21 *For any constant $\epsilon > 0$, $|Pr_{x\in_U A^u}[C(h^g(x)) = 1] - Pr_{y\in_U \{0,1\}^n}[C(y) = 1]| < \epsilon$.*

Proof. Suppose there exists ϵ such that $|Pr_{x\in\{0,1\}^n}[C(x) = 1] - Pr_{y\in A^n}[C(h^g(y)) = 1]| \geq \epsilon$. We will seek a contradiction to the hardness of g via a hybrid argument.

Sample z uniformly from A^n and r uniformly from $\{0,1\}^n$. Create a sequence of $n+1$ distributions H_i on $\{0,1\}^n$ where

- $H_0 = r$;

- $H_n = h^g(z)$;
- $\forall 1 \le i \le n-1,\ H_i = h^g(z)_1 h^g(z)_2 \dots h^g(z)_i r_{i+1} \dots r_n$.

By our assumption, $|\Sigma_{j=1}^n (Pr_{x\sim H_{j-1}}[C(x)=1] - Pr_{x\sim H_j}[C(x)=1])| \ge \epsilon$. Therefore, $\exists j \in [n]$ such that $|Pr_{x\sim H_{j-1}}[C(x)=1] - Pr_{x\sim H_j}[C(x)=1]| \ge \frac{\epsilon}{n}$. Let i be one such index.

Assume $Pr_{x\sim H_i}[C(x)=1] - Pr_{x\sim H_{i-1}}[C(x)=1] \ge \frac{\epsilon}{n}$, otherwise add a not gate at the top of C, and treat the new circuit as C instead.

Consider the following probabilistic TC^0 circuit C' for g. On input x, sample z uniformly from A^n and r uniformly from $\{0,1\}^n$, replace the coordinates of z specified by S_i with x. Sample a random bit $b \in \{0,1\}$. If $C(h^g(z)_1 \dots h^g(z)_{i-1} b r_{i+1} \dots r_n) = 1$, output b, otherwise, output $1-b$.

$$\begin{aligned}
& Pr_{x\in A^{n^\alpha}}[C'(x)=f(x)] \\
= \ & \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b=f(x)] + \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)\neq b \mid b\neq f(x)] \\
= \ & \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b=f(x)] + \tfrac{1}{2} - \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b\neq f(x)] \\
= \ & \tfrac{1}{2} + \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b=f(x)] - \tfrac{1}{2}Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b\neq f(x)] \\
= \ & \tfrac{1}{2} + Pr_{x\in A^{n^\alpha}}[C'(x)=b \mid b=f(x)] - Pr_{x\in A^{n^\alpha}}[C'(x)=b] \\
= \ & \tfrac{1}{2} + (Pr_{y\in H_i}(C(y)=1) - Pr_{y\in H_{i-1}}(C(y)=1)) \\
\ge \ & \tfrac{1}{2} + \tfrac{\epsilon}{n}
\end{aligned}$$

Hence, there is a fixing of values for z, r and b satisfying the property that $Pr_{x\in A^{n^\alpha}}[C'(x,z,r,b)=f(x)] \ge \frac{1}{2}+\frac{\epsilon}{n}$. Note that in this case $\forall 1 \le k \le i-1,\ h^g(z)_k$ is function on input $x|_{S_k\cap S_i}$. Since $\forall k \neq i,\ |S_i \cap S_k| \le b$, we only need a TC^0 circuit of size at most $2^{O(b)}$ and of depth at most 2 to compute each $h^g(z)_k$. In conclusion, we obtain a TC^0 circuit C'' of size at most $(2^{O(b)}+1)n$ and of depth at most $d+2$ such that $Pr_{x\in A^{n^\alpha}}[C'(x) = f(x)] \ge \frac{1}{2}+\frac{\epsilon}{n} \ge \frac{1}{2}+\frac{1}{n^2}$ when n is large enough, a contradiction. □

□

The simulation in Lemma 20 is quite uniform, thus, plugging in appropriate segments of $\mathrm{WP}^{\otimes}$ as our candidates for the hard function g, we derive our first main result.

Theorem 22 *If* WP *is not infinitely often computed by* $\mathrm{TC}^0(n^{1+\gamma})$ *circuit families for some constant* $\gamma > 0$, *then any language accepted by polynomial-size probabilistic uniform* TC^0 *circuit family is in* $\mathsf{uTC}^0(\mathsf{SUBEXP})$.

Proof. Fix any small constant $\delta > 0$. Let L be a language accepted by some probabilistic uniform TC^0 circuit family of size at most n^k and of depth at most d for some constants k, d.

Choose m such that $n^{\frac{\delta}{12}} \le m \le n^{\frac{\delta}{6}}$, and let α be such that $m = n^\alpha$. By Theorem 16, when m is large enough, $\mathrm{WP}^{\otimes}_m$ is $(\frac{1}{2}+\frac{1}{n^{2k}})$-hard for TC^0 circuits of size n^{2k} and depth $d+c$, where c is any constant. Hence, as a consequence of Lemma 20, we obtain a probabilistic oracle TC^0 circuit for L_n of depth $d+1$. Since the computation only needs $O(m^6)$ random bits, it can be turned into a deterministic oracle TC^0 circuit of depth $d+2$ and of size at most $O(n^{2k}) * 2^{O(m^6)} \le 2^{O(n^\delta)}$ (when n is large enough), where we evaluate the previous circuit on every possible random string and add an extra MAJORITY gate at the top. The oracle gates all have fan-in $m \le n^{\delta/6}$, and thus can be replaced by DNF circuits of size $2^{O(n^\delta)}$, yielding a deterministic TC^0 circuit of size $2^{O(n^\delta)}$ and depth $d+3$.

We need to show that this construction is uniform, so that the direct connection language can be recognized in time $O(n^\delta)$. The analysis consists of three parts.

- The connectivity between the top gate and the output gate of individual copies is obviously computable in time $m^6 \le n^\delta$.

- The connectivity inside individual copies is DLOGTIME-uniform, hence, n^δ-uniform.
- By Lemma 19 each S_i is computable in time $O(dm^2)$ which is $O(m^2)$ since d is a constant only depending on δ. Moreover, notice that $\mathrm{WP}^{\otimes}$ is a linear-time decidable language. Therefore, the DNF expression corresponding to each oracle gate can be computed within time $O(m^2) \le n^\delta$.

In conclusion, the above construction produces a uniform TC^0 circuit of size 2^{n^δ}. Since δ is arbitrarily chosen, our statement holds. □

Note that the above conclusion can be strengthened to the following form: any language accepted by a polynomial-size probabilistic $o(n)$-uniform TC^0 circuit family is in $\mathsf{uTC}^0(\mathsf{SUBEXP})$.

5 Consequences of pathetic arithmetic circuit lower bounds

In this section we show that a pathetic lower bound assumption for *arithmetic circuits* yields a uniform derandomization of a special case of polynomial identity testing (introduced and studied by Dvir *et al.* [DSY09]).

The explicit polynomial that we consider is $\{\mathrm{IMM}_n\}_{n>0}$, where IMM_n is the $(1,1)^{th}$ entry of the product of n 3×3 matrices whose entries are all distinct indeterminates. Notice that IMM_n is a degree n multilinear polynomial in $9n$ indeterminates, and IMM_n can be considered as a polynomial over any field $\mathbb{F}$.

Arithmetic circuits computing a polynomial in the ring $\mathbb{F}[x_1, x_2, \ldots, x_n]$ are directed acyclic graphs with the indegree zero nodes (the inputs nodes) labeled by either a variable x_i or a scalar constant. Each internal node is either a $+$ gate or a $\times$ gate, and the circuit *computes* the polynomial that is naturally computed at the output gate. The circuit is a *formula* if the fanout of each gate is 1.

Before going further, we pause to clarify a point of possible confusion. There is another way that an arithmetic circuit C can be said to compute a given polynomial $f(x_1, x_2, \ldots, x_n)$ over a field $\mathbb{F}$; even if C does not compute f in the sense described in the preceding paragraph, it can still be the case that for all scalars $a_i \in \mathbb{F}$ we have $f(a_1, \ldots, a_n) = C(a_1, \ldots, a_n)$. In this case, we say that C *functionally* computes f over $\mathbb{F}$. If the field size is larger than the syntactic degree of circuit C and the degree of f, then the two notions coincide. Assuming that f is not *functionally* computed by a class of circuits is a *stronger* assumption than assuming that f is not computed by a class of circuits (in the usual sense). In our work in this paper, we use the weaker intractability assumption.

An *oracle* arithmetic circuit is one that has *oracle* gates: For a given sequence of polynomials $A = \{A_n\}$ as oracle, an oracle gate of fan-in n in the circuit evaluates the n-variate polynomial A_n on the values carried by its n input wires. An oracle arithmetic circuit is called *pure* (following [AK10]) if all non-oracle gates are of bounded fan-in. (Note that this use of the term "pure" is unrelated to the "pure" arithmetic circuits defined by Nisan and Wigderson [NW97].)

The class of polynomials computed by polynomial-size arithmetic formulas is known as arithmetic NC^1. By [BOC92] the polynomial IMM_n is complete for this class. Whether IMM_n has polynomial size *constant-depth* arithmetic circuits is a long-standing open problem in the area of arithmetic circuits [NW97]. In this context, the known lower bound result is that IMM_n requires exponential size multilinear depth-3 circuits [NW97].

Very little is known about lower bounds for general constant-depth arithmetic circuits, compared to what is known about constant-depth Boolean circuits. Exponential lower bounds for depth-3 arithmetic circuits over finite fields were shown in [GK98] and [GR00]. On the other hand, for depth-3 arithmetic circuits over fields of characteristic zero only quadratic lower bounds are known [SW01]. However, it is shown in [RY09] that the determinant and the permanent require exponential size *multilinear* constant-depth arithmetic circuits. More details on the current status of arithmetic circuit lower bounds can be found in Raz's paper [Raz08, Section 1.3].

Definition 23 *We say that a sequence of polynomials $\{p_n\}_{n>0}$ in $\mathbb{F}[x_1, x_2, \ldots, x_n]$ is $(s(n), m(n), d)$-downward self-reducible if there is a pure oracle arithmetic circuit C_n of depth $O(d)$ and size $O(s(n))$ that computes the polynomial p_n using oracle gates only for $p_{m'}$, for $m' \le m(n)$.*

Analogous to [AK10, Proposition 7], we can easily observe the following. It is a direct divide and conquer argument using the iterated product structure.

Lemma 24 *For each $1 > \epsilon > 0$ the polynomial sequence $\{IMM_n\}$ is $(n^{1-\epsilon}, n^{\epsilon}, 1/\epsilon)$-downward self-reducible.*

An easy argument, analogous to Theorem 8, shows that Lemma 24 allows for the amplification of weak lower bounds for $\{\mathrm{IMM}_n\}$ against arithmetic circuits of constant depth:

Theorem 25 *Suppose there is a constant $\delta > 0$ such that for all d and every n, the polynomial sequence $\{IMM_n\}$ requires depth-d arithmetic circuits of size at least $n^{1+\delta}$. Then, for any constant depth d the sequence $\{IMM_n\}$ is not computable by depth-d arithmetic circuits of size n^k for any constant $k > 0$.*

Our goal is to apply Theorem 25 to derandomize a special case of polynomial identity testing (first studied in [DSY09]). To this end we restate a result of Dvir et. al [DSY09].

Theorem 26 (Theorem 4 in [DSY09]) *Let n, s, r, m, t, d be integers such that $s \geq n$. Let $\mathbb{F}$ be a field which has at least $2mt$ elements. Let $P(x, y) \in \mathbb{F}[x_1, \ldots, x_n, y]$ be a non-zero polynomial with $\deg(P) \leq t$ and $\deg_y(P) \leq r$ such that P has an arithmetic circuit of size s and depth d over $\mathbb{F}$. Let $f(x) \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial with $\deg(f) = m$ such that $P(x, f(x)) \equiv 0$. Then $f(x)$ can be computed by a circuit of size $s' = poly(s, m^r)$ and depth $d' = d + O(1)$ over $\mathbb{F}$.*

Let the underlying field $\mathbb{F}$ be large enough ($\mathbb{Q}$, for instance). The following lemma is a variant of Lemma 4.1 in [DSY09]. For completeness, we provide its proof here.

Lemma 27 (Variant of Lemma 4.1 in [DSY09]) *Let n, r, s be integers and let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a nonzero polynomial with individual degrees at most r that is computed by an arithmetic circuit of size $s \geq n$ and depth d. Let $m = n^{\alpha}$ be an integer where $\alpha > 0$ is an arbitrary constant. Let $S_1, S_2, \ldots, S_n$ be the sets of the (m, b)-design constructed in Lemma 19 where $b = \lceil \frac{1}{\alpha} \rceil$. Let $p \in \mathbb{F}[z_1, \ldots, z_m]$ be a multilinear polynomial with the property that*

$$F(y) = F(y_1, y_2, \ldots, y_u) \triangleq f(p(y|_{S_1}), \ldots, p(y|_{S_n})) \equiv 0 \tag{1}$$

Then there exists absolute constants a and k such that $p(z)$ is computable by an arithmetic circuit over $\mathbb{F}$ with size bounded by $O((sm^r)^a)$ and having depth $d + k$.

Proof. Consider the following set of hybrid polynomials:

$$\begin{aligned} F_0(x, y) &= f(x_1, x_2, \ldots, x_n) \\ F_1(x, y) &= f(p(y|_{S_1}), x_2, \ldots, x_n) \\ &\vdots \\ F_n(x, y) &= f(p(y|_{S_1}), \ldots, p(y|_{S_n})) \end{aligned}$$

The assumption implies that $F_0 \not\equiv 0$ while $F_n \equiv 0$. Hence, there exists $0 \leq i < n$ such that $F_i \not\equiv 0$ and $F_{i+1} \equiv 0$. Notice that F_i is a nonzero polynomial in the variables $\{x_j \mid i + 2 \leq j \leq n\}$ and the variables $\{y_j \mid j \in S_1 \cup S_2 \cup \cdots \cup S_i\}$.

We recall the well-known Schwartz-Zippel lemma.

Lemma 28 (Schwartz-Zippel) *Let $\mathbb{F}$ be a field and let $f \in \mathbb{F}[x_1, ..., x_n]$ be a non-zero polynomial with total degree at most r. Then for any finite subset $S \subset \mathbb{F}$ we have*

$$|\{c \in S^n \ : \ f(c) = 0\}| \leq r \cdot |S|^{n-1} \tag{2}$$

Since $\deg(F_i) \leq nrm$, then if we assume that $\mathbb{F}$ has size more than nrm, Lemma 28 assures that we can assign values from the field $\mathbb{F}$ to the variables $\{x_j \mid i+2 \leq j \leq n\}$ and the variables $\{y_j \mid j \notin S_{i+1}\}$ so that F_i remains a nonzero polynomial in the remaining variables. More precisely, fixing these variables to scalar values yields a polynomial $\tilde{f}$ with the property that

$$\begin{array}{lcl} \tilde{f}(q_1(y|_{S_1 \cap S_{i+1}}), \ldots, q_1(y|_{S_i \cap S_{i+1}}), x_{i+1}) & \not\equiv & 0 \\ \tilde{f}(q_1(y|_{S_1 \cap S_{i+1}}), \ldots, q_1(y|_{S_i \cap S_{i+1}}), p(y|_{S_{i+1}})) & \equiv & 0 \end{array}$$

where $q_j(y|_{S_j \cap S_{i+1}})$ is the polynomial obtained from $p_j(y|_{S_j})$ after fixing the variables in $S_j \setminus S_{i+1}$.

Rename the variables $\{y_j \mid j \in S_{i+1}\}$ with $\{z_j \mid 1 \leq j \leq m\}$ and replace x_{i+1} by w. We obtain a polynomial g with the property that

$$\begin{array}{lcl} g(z_1, \ldots, z_m, w) & \not\equiv & 0 \\ g(z_1, \ldots, z_m, p(z_1, \ldots, z_m)) & \equiv & 0 \end{array}$$

In order to apply Theorem 26, the only thing that remains is to calculate the circuit complexity of g. $\forall j \neq i+1,\ |S_j \cap S_{i+1}| \leq b$ which is a constant. Hence, for any $j \leq i$, $q_j(y|_{S_j \cap S_{i+1}})$ is a polynomial depending on a constant number of variables, which can be computed by a constant-size arithmetic circuit of depth 2 (Basically, it is a sum of monomials). Under the assumption that f has a circuit of size s and depth d, g is computable by a circuit of size $s + O(n)$ and depth $d+2$ which is a composition of the aforementioned circuits. It is important to note that $\deg_w(g) = \deg_{x_{i+1}}(f) \leq r$.

Now we use Theorem 26 to obtain that $p(z)$ has a circuit of size at most $(sm^r)^a$ and depth $d+k$, which concludes our proof. □

At this point we describe our deterministic black-box identity testing algorithm for constant-depth arithmetic circuits of polynomial size and bounded individual degree. Let n, m, u, α be the parameters as in Lemma 19. Given such a circuit C over variables $\{x_i \mid i \in [n]\}$ of size n^t, depth d and individual degree r, we simply replace x_i with IMM$(y|S_i)$ where y is a new set of variables $\{y_j \mid j \in [u]\}$. Let $\tilde{C}[y_1, \ldots, y_u]$ denote the polynomial computed by the new circuit.

Notice that the total degree of $\tilde{C}$ is bounded by u^c where c is a constant depending on the combinatorial design and r. Let $R \subseteq \mathbb{F}$ be any set of $u^c + 1$ distinct points. Then by Lemma 28 the polynomial computed by $\tilde{C}$ is identically zero if and only if $\tilde{C}(a_1, a_2, \ldots, a_u) = 0$ for all $(a_1, a_2, \ldots, a_u) \in R^u$.

This gives us the claimed algorithm. Its running time is bounded by $O((u^c+1)^u) = O(2^{7\alpha n^{6\alpha}})$. Since α can be chosen to be arbitrarily small, we have shown that this identity testing problem is in deterministic sub-exponential time. The correctness of the algorithm follows from the next lemma.

Lemma 29 *If for every constant $d' > 0$, the polynomial sequence $\{IMM_n\}$ is not computable by depth-d' arithmetic circuits of size n^k for any $k > 0$, then $C[x_1, \ldots, x_n] \equiv 0$ if and only if $\tilde{C}[y_1, \ldots, y_u] \equiv 0$.*

Proof. The only-if part is easy to see. Let us focus on the if part. Suppose it is not the case, which means that $\tilde{C}[y_1, \ldots, y_u] \equiv 0$ but $C[x_1, \ldots, x_n] \not\equiv 0$. Then let $C[x_1, \ldots, x_n]$ play the role of $f[x_1, \ldots, x_n]$ in Lemma 27 and let IMM$[z_1, \ldots, z_m]$ take the place of $p[z_1, \ldots, z_m]$. Therefore, IMM$[z_1, \ldots, z_m]$ is computable by a circuit of depth $d+k$ and size at most $(n^t m^r)^a = m^{O(1)}$, a contradiction. □

Putting it together, we get the following result.

Theorem 30 *If there exists $\delta > 0$ such that for any constant $e > 0$, IMM requires depth-e arithmetic circuits of size at least $n^{1+\delta}$, then the black-box identity testing problem for constant-depth arithmetic circuits of polynomial size and bounded individual degree is in deterministic sub-exponential time.*

Next, we notice that the above upper bound can be sharpened considerably. The algorithm simply takes the OR over subexponentially-many evaluations of an arithmetic circuit; if any of the evaluations does not evaluate to zero, then we know that the expressions are not equivalent; otherwise they are. Note that evaluating an arithmetic circuit can be accomplished in logspace. (When evaluating a circuit over $\mathbb{Q}$, this is shown in [HAB02, Corollary 6.8]; the argument for other fields is similar, using standard results about the complexity of field arithmetic.) Note also that every language computable in logspace has AC^0 circuits of subexponential size. (This appears to have been observed first by Gutfreund and Viola [GV04]; see also [AHM$^+$08] for a proof.) This yields the following uniform derandomization result.

Theorem 31 *If there are no constant-depth arithmetic circuits of size $n^{1+\epsilon}$ for the polynomial sequence $\{IMM_n\}$, then for every constant d, black-box identity testing for depth-d arithmetic circuits with bounded individual degree can be performed by a uniform family of constant-depth AC^0 circuits of subexponential size.*

We call attention to an interesting difference between Theorems 22 and 31. In Theorem 31, in order to solve the identity testing problem with uniform AC^0 circuits of size 2^{n^ϵ} for smaller and smaller ϵ, the depth of the AC^0 circuits increases as ϵ decreases. In contrast, in order to obtain a deterministic threshold circuit of size 2^{n^ϵ} to simulate a given probabilistic TC^0 algorithm, the argument that we present in the proof of Theorem 22 gives a circuit whose depth is not affected by the choice of ϵ. We do not know if a similar improvement of Theorem 31 is possible, but we observe here that the depth need not depend on ϵ if we use threshold circuits for the identity test.

Theorem 32 *If there are no constant-depth arithmetic circuits of size $n^{1+\epsilon}$ for the polynomial sequence $\{IMM_n\}$, then there is a constant c such that, for every constant d and every $\gamma > 0$, black-box identity testing for depth-d arithmetic circuits with bounded individual degree can be performed by a uniform family of depth $d + c$ threshold circuits of size 2^{n^γ}.*

Proof. We provide only a sketch. Choose $\alpha < \gamma/14$, where α is the constant from the discussion in the paragraph before Lemma 29. Thus, our identity testing algorithm will evaluate a depth d arithmetic circuit $C(x_1, \ldots, x_n)$ at fewer than $2^{n^{\gamma/2}}$ points $\vec{v} = (v_1, \ldots, v_n)$, where each v_i is obtained by computing an instance of IMM_{n^α} consisting of n^α 3-by-3 matrices, whose entries without loss of generality have representations having length at most n^α. Thus these instances of IMM have DNF representations of size $2^{O(n^{2\alpha})}$. These DNF representations are uniform, since the direct connection language can be evaluated by computing, for a given input assignment to IMM_{n^α}, the product of the matrices represented by that assignment, which takes time at most $(n^\alpha)^3 < \log(2^{n^{\gamma/2}})$. Evaluating the circuit C on $\vec{v}$ can be done in uniform TC^0 [AAD00, HAB02]. □

Acknowledgments

The possibility of applying random self-reductions to derandomize small classes was suggested to us by Rahul Santhanam. We thank Luke Friedman for many helpful discussions, and we thank Lance Fortnow for some useful suggestions.

References

[AAD00] Manindra Agrawal, Eric Allender, and Samir Datta. On TC^0, AC^0, and arithmetic circuits. *Journal of Computer and System Sciences*, 60(2):395–421, 2000.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity, a modern approach.* Cambridge University Press, 2009.

[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, 1998.

[ACR99] 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. *Theoretical Computer Science*, 221(1-2):3–18, 1999.

[ACRT99] Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and BPP simulations. *SIAM Journal on Computing*, 28(6):2103–2116, 1999.

[AHM$^+$08] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008.

[AK97] Vikraman Arvind and Johannes Köbler. On resource-bounded measure and pseudorandomness. *Proc. Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, pages 235–249, 1997.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57:14:1–14:36, 2010.

[Bar89] David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

[BF99] Harry Buhrman and Lance Fortnow. One-sided versus two-sided error in probabilistic computation. *Proc. of Symp. on Theo. Aspects of Comp. Sci. (STACS)*, pages 100–109, 1999.

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

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

[BOC92] Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992.

[BSSV03] Paul Beame, Michael E. Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, 2003.

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth circuits. *SIAM Journal on Computing*, 39(4):1279–1293, 2009.

[GGH$^+$07] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. A (de)constructive approach to program checking. Technical Report TR07-047, Electronic Colloquium on Computational Complexity (ECCC), 2007. See also STOC 2008.

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 577–582, 1998.

[GR00] D. Grigoriev and A. Razborov. Exponential complexity lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 10:465–487, 2000.

[GV04] Dan Gutfreund and Emanuele Viola. Fooling parity tests with parity gates. *APPROX-RANDOM*, pages 381–392, 2004.

[GVW00] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. Simplified derandomization of BPP using a hitting set generator. *Electronic Colloquium on Computational Complexity*, 7(4), 2000.

[GW99] Oded Goldreich and Avi Wigderson. Improved derandomization of BPP using a hitting set generator. *RANDOM-APPROX*, pages 131–137, 1999.

[HAB02] William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4):695–716, 2002.

[Hås98] Johan Håstad. The shrinkage exponent of de Morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.

[IM02] Kazuo Iwama and Hiroki Morizumi. An explicit lower bound of $5n - o(n)$ for boolean circuits. In *Proc. of Math. Foundations of Comp. Sci. (MFCS)*, volume 2420 of *Lecture Notes in Computer Science*, pages 353–364, 2002.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. *Proc. IEEE Symp. on Found. of Comp. Sci. (FOCS)*, pages 538–545, 1995.

[IPS97] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size-depth tradeoffs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.

[ISW06] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Reducing the seed length in the Nisan-Wigderson generator. *Combinatorica*, 26(6):647–681, 2006.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 220–229, 1997.

[IW01] Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *Journal of Computer and System Sciences*, 63(4):672–688, 2001.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KvM02] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[MV05] Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

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

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Raz08] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 711–720, 2008.

[RY09] Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Journal of Computational Complexity*, 18(2):171–207, 2009.

[Sha81] Adi Shamir. On the generation of cryptographically strong pseudo-random sequences. *Proc. of International Conference on Automata, Languages, and Programming (ICALP)*, pages 544–550, 1981.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[SU05] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52(2):172–216, 2005.

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[Uma03] Christopher Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67(2):419–440, 2003.

[Vio05] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005.

[vL99] Jacobus H. van Lint. *Introduction to Coding Complexity*. Springer-Verlag, 1999.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity*. Springer, 1999.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). *Proc. IEEE Symp. on Found. of Comp. Sci. (FOCS)*, pages 80–91, 1982.