Electronic Colloquium on Computational Complexity, Report No. 133 (2011)

Marginal Hitting Sets Imply Super-Polynomial Lower Bounds for Permanent

Maurice Jansen*
Laboratory for Foundations of Computer Science
School of Informatics
The University of Edinburgh
maurice.julien.jansen@gmail.com

Rahul Santhanam†
Laboratory for Foundations of Computer Science
School of Informatics
The University of Edinburgh
rsanthan@inf.ed.ac.uk

October 4, 2011

Abstract

Suppose f is a univariate polynomial of degree $r = r(n)$ that is computed by a size n arithmetic circuit. It is a basic fact of algebra that a nonzero univariate polynomial of degree r can vanish on at most r points. This implies that for checking whether f is identically zero, it suffices to query f on an arbitrary test set of $r + 1$ points. Could this brute-force method be improved upon by *a single point*? We develop a framework where such a marginal improvement implies that Permanent does not have polynomial size arithmetic circuits.

More formally, we formulate the following hypothesis for any field of characteristic zero: There is a fixed depth d and some function $s(n) = O(n)$, such that for arbitrarily small $\epsilon > 0$, there exists a hitting set $\mathcal{H}_n \subset \mathbb{Z}$ of size at most $2^{s(n^\epsilon)}$ against univariate polynomials of degree at most $2^{s(n^\epsilon)}$ computable by size n constant-free[1] arithmetic circuits, where $\mathcal{H}_n$ can be encoded by uniform TC^0 circuits of size $2^{O(n^\epsilon)}$ and depth d. We prove that the hypothesis implies that Permanent does not have polynomial size constant-free arithmetic circuits.

Our hypothesis provides a unifying perspective on several important complexity theoretic conjectures, as it follows from these conjectures for different degree ranges as determined by the function $s(n)$. We will show that it follows for $s(n) = n$ from the widely-believed assumption that *poly* size Boolean circuits cannot compute the Permanent of a $0, 1$-matrix over $\mathbb{Z}$. The hypothesis can also be easily derived from the Shub-Smale τ-conjecture [SS95], for any $s(n)$ with $s(n) = \omega(\log n)$ and $s(n) = O(n)$. This implies our result strengthens a theorem by Bürgisser [Bür00], who derives the same lower bound from the τ-conjecture. For $s(n) = 0$, the hypothesis follows from the statement that $(n!)$ is ultimately hard, a statement that is known to imply P $\neq$ NP over $\mathbb{C}$ [SS95].

We apply our randomness-to-hardness theorem to prove the following unconditional result for Permanent: either Permanent does not have uniform constant-depth threshold circuits of sub-exponential size, or Permanent does not have polynomial-size constant-free arithmetic circuits.

*Supported by EPSRC Grant H05068X/1

†Supported in part by EPSRC Grant H05068X/1

[1] All our circuits use the operations addition and multiplication only. For a constant-free arithmetic circuit the only allowed constant labels in the circuit are in $\{-1, 1\}$. Our hardness-to-randomness theorem can be generalized to a circuit model where arbitrary constants from $\mathbb{F}$ are allowed, using a theorem of [Bür00]. The latter result assumes the Generalized Riemann Hypothesis.

ISSN 1433-8092

Turning to the Boolean world, we give a simplified proof of the following strengthening of Allender's lower bound [All99] for the (0,1)-Permanent: either the (0,1)-Permanent is not simultaneously in polynomial time and sub-polynomial space, or logarithmic space does not have uniform constant-depth threshold circuits of polynomial size.

1 Introduction

Polynomial identity testing (PIT) is the problem of deciding for a multivariate polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, given in some succinct representation, e.g. an arithmetic circuit, whether f is identical to the zero element of $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Using the Schwartz-Zippel-deMillo-Lipton Lemma [DL78, Sch80, Zip79], Ibarra and Moran [IM83] show this problem is in coRP, when f is given in the arithmetic circuit representation over $\mathbb{Z}$.

Whether PIT can be solved efficiently without randomization is closely connected to the quest for proving lower bounds. This connection has been known at least already since the work of Heintz and Schnorr [HS80]. In a seminal work, Kabanets and Impagliazzo [KI04] show that giving an NSUBEXP time algorithm for PIT implies that either the permanent polynomial $\mathrm{per}_n = \sum_{\sigma \in S_n} \prod_{j=1}^{n} x_{j\sigma(j)}$ does not have polynomial size arithmetic circuits, or that NEXP $\not\subseteq$ P/*poly*. Agrawal [Agr05] shows that the construction of an explicit *poly*(n) size hitting set against the class of multilinear polynomials computed by size n arithmetic circuits, would yield an exponential arithmetic circuit size lower bound for a multilinear polynomial with coefficients computable in PSPACE. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is a hitting set against some class of polynomials $\mathcal{C}$ in n variables, if for every nonzero $f \in \mathcal{C}$, there exists $h \in \mathcal{H}$ with $f(h) \neq 0$.

Unfortunately, all currently known randomness-to-hardness results based on derandomization of low-degree (or multilinear) *multivariate* PIT fall short of establishing a sufficient condition for proving a super-polynomial lower bound for a polynomial as explicit as the permanent. Koiran [Koi11] proposes deriving such lower bounds from the stronger[2] assumption that we can derandomize *exponential degree univariate* PIT. In this paper we further explore the univariate route to explicit lower bounds. Trivially, any set of $r+1$ distinct points is a hitting set against any class of univariate polynomials where degrees are bounded by r, which we think of as a 'brute-force hitting set'. Our main contribution here is to develop a framework where it holds that improvement over brute-force by *a single point* for polynomials computed by size n arithmetic circuits already implies a super-polynomial lower bound for Permanent. For this purpose we state the following derandomization assumption (in fact, we use a somewhat weaker assumption which is stated as Hypothesis 1 in Section 3):

Hypothesis 1. *At some fixed depth d, for some nondecreasing function $s(n) = O(n)$, for arbitrarily large $k \in \mathbb{N}$, for infinitely many n, there exists a hitting set $\mathcal{H}_n \subset \mathbb{Z}$ of size at most $2^{s(\lceil n^{1/k} \rceil)}$ against the class of univariate polynomials of degree at most $2^{s(\lceil n^{1/k} \rceil)}$ that are computable by size n constant-free arithmetic circuits. Furthermore, $\mathcal{H}_n$ can be encoded by a uniform* TC^0 *circuit C_n of size $2^{O(n^{1/k})}$ and depth d with $s(\lceil n^{1/k} \rceil)$ many variable inputs.*

In the above, when we say the hitting set $\mathcal{H}_n$ is encoded by C_n, it means that $\mathcal{H}_n \subseteq \{C_n(a) : a \in \{0,1\}^{s(\lceil n^{1/k} \rceil)}\}$, where we use the standard binary representation of integers. We will work over a field $\mathbb{F}$ of characteristic zero only. In the constant-free model the only constants used for labelling

[2] For example, the multilinear case can be reduced to the univariate case by letting $x_i = x^{2^i}$, for $i \in [n]$.

gates are in $\{-1, 1\}$, cf. [Bür09, KP11]. All of our circuits are restricted to have addition and multiplication gates only. We let $\tau(f)$ denote the constant-free (division-free) arithmetic circuit size of f, cf. [Bür09]. We establish the following connection:

Theorem 1. *If Hypothesis 1 is true, then Permanent does not have polynomial size constant-free arithmetic circuits.*

Perhaps the most striking aspect of our hypothesis is that it asks for a hitting set of size at most $2^{s(\lceil n^{1/k} \rceil)}$, whereas we know that we can do brute-force testing with *any* set of size $2^{s(\lceil n^{1/k} \rceil)} + 1$. Recently, Williams [Wil10, Wil11] has initiated a program to prove circuit lower bounds by improving on exhaustive search for circuit satisfiability or approximating the number of satisfying assignments for a circuit. He has used this approach [Wil11] to show that NEXP does not have polynomial-size ACC^0 circuits.

A natural question [Wil10] is whether some analogue of the connection found by Williams between lower bounds and algorithmic savings over exhaustive search holds in the arithmetic setting. Theorem 1 can be seen as a partial answer to his question. On the one hand, while Williams' results need a super-polynomial savings over exhaustive search, in our setting, just a reduction of the search space by one point already gives us lower bounds. However, we do require this savings to hold in the context of hitting sets, which correspond to black-box derandomization, while in Williams' results the algorithm improving on exhaustive search is allowed access to the circuit for whose acceptance probability an approximation is required.

We demonstrate the viability of our framework by applying Theorem 1 to obtain strong unconditional lower bounds for Permanent (See Section 1.1 below). This shows that already elementary methods for constructing hitting sets can yield strong lower bounds when combined with our techniques. By taking advantage of the algebraic structure of the problem, it is possible we could do much better.

Another salient aspect of our framework is that it provides a unifying perspective on several important complexity theoretic conjectures. Namely, Hypothesis 1 follows from these conjectures for different degree ranges as determined by the function $s(n)$. We will observe that the hypothesis with $s(n) = n$ follows from the widely believed assumption that polynomial size Boolean circuits cannot compute $0, 1$-permanent over $\mathbb{Z}$. We also note that our randomness-to-hardness theorem strengthens the result of Ref. [Bür09], which shows that $\tau(\mathrm{per}_n) \neq n^{O(1)}$, in case the Shub-Smale τ-conjecture [SS95] is true. The statement of our hypothesis can be easily derived with $s(n) = \omega(\log n)$ from the τ-conjecture (See Section 3), and appears to be a much weaker statement. At the very low-end, for $s(n) = 0$, we will show that the Hypothesis is true if $(n!)$ is ultimately hard[3] in the sense of Ref. [SS95]. The latter is defined to mean that for any sequence (a_n) of nonzero integers, $\tau(a_n \cdot n!)$ is not *polylog*(n) bounded. Ref. [SS95] shows that if $(n!)$ is ultimately hard to compute, then one has the separation $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$ for the Blum-Shub-Smale model.

Incidentally, by an easy counting argument one can demonstrate the existence of the hitting sets as posed in the hypothesis (for various $s(n)$, and $s(n) = 0$ in particular), but where the set is encoded by *nonuniform* TC^0 circuits of the required size and fixed depth. The real issue is to get a *uniform* encoding, or at least a sufficiently *succinct* encoding in the sense of Ref. [JS11].

We note that Theorem 1 generalizes to the setting where circuits are allowed to carry arbitrary constants from $\mathbb{F}$, due to a result of Bürgisser [Bür00], provided we assume the Generalized Riemann

[3] It is well-known that $\tau(n!) = \mathit{polylog}(n)$ implies that factoring integers is in P/*poly*, cf. [Bür09]. Related to this, Lipton [Lip94] shows that if factoring is hard on average, then a somewhat weaker version of the τ-conjecture is true.

Hypothesis. In this case the hitting set has to work against circuits over $\mathbb{F}$, but also the resulting lower bound will be for circuits over $\mathbb{F}$. In this case it is only interesting to consider the case where $s(n) = \omega(\log n)$. For example, for $s(n) = O(\log n)$, for any $h_1, h_2, \ldots, h_t \in \mathbb{F}$ with $t = 2^{s(n^\epsilon)} = n^{O(\epsilon)}$, $(x - h_1)(x - h_2) \ldots (x - h_t)$ can be computed by a size $n^{O(\epsilon)}$ arithmetic circuit over $\mathbb{F}$, so we cannot get a hitting set of size $t = 2^{s(n^\epsilon)}$ against size n circuits in this case.

The work most closely related to ours is Ref. [Koi11], where lower bounds are derived for the permanent from certain kinds of hitting sets for classes of univariate polynomials. However, the emphasis there is on finding the simplest possible class of univariate polynomials for which the randomness-hardness connection holds rather than on the size of the hitting set. Koiran requires his hitting sets to be of polynomial or slightly super-polynomial size. In contrast, we are interested in the weakest possible assumption on hitting set size which still yields superpolynomial lower bounds. An important benefit of our approach is that there is no a priori required degree bound for which we must derandomize univariate PIT, where in Ref. [Koi11] this bound is *exponential*. For example, even at the high end for $s(n) = n$, where Hypothesis 1 is implied by the assumed hardness of the 0, 1-permanent, we can get away with essentially only considering subexponential degrees. For $s(n) = \omega(\log n)$, which is the regime where the hypothesis is warranted by the τ-conjecture, all one needs to do is marginally improve upon the brute-force method for the class of polynomials of degree $2^{s(\lceil n^{1/k} \rceil)} = 2^{\omega(\frac{1}{k} \log n)}$ computed by size n circuits. For moderately growing $s(n)$ this is only slightly super-polynomial in n.

1.1 Unconditional Lower Bounds

Using Theorem 1 we will derive the following unconditional hardness result for the permanent:

Theorem 2. *At least one of the following items must be true:*

- *For every integer $d \geq 1$, there exists $\epsilon > 0$ such that 0, 1-permanent can not be computed by uniform TC^0 circuits of size 2^{n^ϵ} and depth d.*
- *Permanent does not have constant-free arithmetic circuits of polynomial size.*

Note that the first item of the above disjunction by itself, is stronger than the currently best-known uniform TC^0 circuit lower bound for permanent, due to Allender [All99]. The latter bound is of level $T(n)$, for any function $T(n)$ such that for any constant k the kth iterate $T^{(k)}(n) = 2^{o(n)}$. Let us also emphasize that the separate parts of this disjunction make a statement about the hardness of the *same* function, albeit in different computational models.

Turning to the Boolean world, we give a simple proof of the following strengthening of Allender's [All99] lower bound for Permanent against uniform TC^0.

Theorem 3. *Either (0,1)-Permanent* $\notin \mathrm{DSPACE}(n^{o(1)}) \cap \mathrm{P}$ *or* $L \not\subseteq \mathrm{TC}^0$.

Theorem 3 implies that the Permanent is not in uniform TC^0 since $L \subseteq \mathrm{P} \cap \mathrm{DSPACE}(n^{o(1)})$.

1.2 Techniques

Let us first consider Theorem 1, and for simplicity let us assume that $s(n) = n$. In the univariate setting, given a family of hitting sets $\{\mathcal{H}_n\}$ of size $2^{n^{1/k}}$ against size n circuits computing polynomials of degree $r = |\mathcal{H}_n|$, there is a natural polynomial f_n of degree r that requires size n^k circuits.

Namely, take $f_n = \prod_{h \in \mathcal{H}_{n^k}} (x-h)$. By Hypothesis 1 we can do this for arbitrarily large k. Moreover, we have uniform TC^0 circuits of size $2^{O(n)}$ and some fixed depth d for enumerating the 2^n elements of $\mathcal{H}_{n^k}$. One key idea is that the size and depth bounds for these circuits are independent of k (although the circuits themselves may very well depend on k). Multiplying out we can express the 2^n coefficients of f_n as elementary symmetric polynomials in elements of $\mathcal{H}_n$. Using the uniform TC^0 circuits for iterated integer multiplication due to Hesse, Allender and Barrington [HAB01], we get uniform TC^0 circuits of size $2^{O(n)}$ computing the coefficients of f_n.

For the heart of the proof we derive a contradiction by means of a 'compression argument' to get n^c size circuits for f_n for some constant c that *does not depend on* k, based on the assumption that $\tau(\mathrm{per}_n) = n^{O(1)}$. This kind of argument has been key in Refs. [Bür09, Koi11, JS11]. Assuming $\tau(\mathrm{per}_n) = n^{O(1)}$, for a first compression step, one uses the relation between the counting hierarchy CH and TC^0 to get the coefficients of f_n "weakly-definable" in CH. Weak-definability in CH means we can decide the ith bit of the coefficient in CH given an $O(n)$ bit index i. If $\tau(\mathrm{per}_n) = n^{O(1)}$, then we have the collapse CH/*poly* = P/*poly*. This means that the coefficients are weakly-definable in P/*poly*. For a second compression step one exploits this fact and applies Valiant's Criterion to get f_n as a projection of some polynomial h_n in VNP^0, where the latter is Valiant's analogue of NP in the (constant-free) algebraic model. Permanent is more or less complete for the latter class, aside from some minor technical issues related to the constant-free model. Note that f_n has degree 2^n. In the arithmetic circuit model a power like x^{2^n} can be represented succinctly by $O(n)$ circuitry by repeated squaring. This fact can be utilized to yield some extra amount of compression. Also in the second compression step, one leverages the assumption that $\tau(\mathrm{per}_n) = n^{O(1)}$ one more time to use a collapse result for VNP^0 in order to finally get n^c size constant-free circuits for f_n, for some constant c.

The crucial observation for us is that the size and depth parameters of the TC^0 circuits we start with are not dependent on k, and neither are any of the subsequently applied collapse results. This means that the constant c does not depend on k. Since k can be chosen to be arbitrarily large this yields a contradiction. This completes the sketch of the proof of Theorem 1.

Next we consider the applications. Unfortunately we cannot prove Hypothesis 1 at the present moment. How then do we obtain unconditional lower bounds? The key idea is to use a *win-win* argument. We can show that hitting sets of the form we desire are constructible in a "large" complexity class, specifically in a fixed level of the Polynomial Hierarchy. Now either the Polynomial Hierarchy has sub-exponential size uniform TC^0 circuits or it does not. If it does, then Hypothesis 1 holds and by Theorem 1, we get that Permanent does not have polynomial-size constant-free arithmetic circuits. If it does not, then using theorems of Valiant [Val79], Toda [Tod91] and Zankó [Zan91], we have that Permanent is hard for PH, to the extent that we can show that the Permanent does not have uniform sub-exponential size TC^0 circuits. This yields Theorem 2. We note that our construction of hitting sets in the Polynomial Hierarchy is pretty simple - it just uses a counting argument. This already gives us unconditional lower bounds for the Permanent. By taking advantage of the algebraic structure of the problem, it is possible we could do much better.

The proof of Theorem 3 is completely different, as it is purely a result about the Boolean world. Allender's proof [All99] of uniform TC^0 lower bounds for Permanent proceeds by considering the question of whether a P-complete language has small TC^0 circuits or not, and deriving a lower bound in either case. We simplify his proof by considering instead a question about inclusions between larger complexity classes, namely whether a PSPACE-complete language is in CH, and showing that either way, an interesting lower bound holds. If yes, then we show that Permanent cannot be both in P and DSPACE($n^{o(1)}$), i.e., there is a tradeoff between time and space for computing the Permanent. If no, we show a separation between two low-level complexity classes - logarithmic space and uniform TC^0. Note that in the first of these two cases, a much stronger lower bound than uniform TC^0 holds for the Permanent, while in the second case, a TC^0 lower bound holds for a class that is much weaker in computational power than the Permanent.

2 Preliminaries

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of variables and let $\mathbb{F}$ be a field. We assume throughout the paper that $\mathbb{F}$ has characteristic zero. This means that $\mathbb{Z} \subset \mathbb{F}$. An arithmetic circuit Φ over X and $\mathbb{F}$ is given by a labelled directed acyclic graph. Nodes with in-degree zero must be labelled with elements of $X \cup \mathbb{F}$. Nodes with higher in-degree must be labelled by $+$ or $\times$. To each node in Φ (also called a gate), we associate a polynomial $\in \mathbb{F}[X]$ in the standard way. Polynomials associated at gates in Φ are called the polynomials computed by Φ. For the size $s(\Phi)$ we count the number of edges in the underlying graph. The notation $|\Phi|$ is synonymous with $s(\Phi)$. For a polynomial $f \in \mathbb{F}[X]$, the arithmetic circuit complexity $L(f)$ is taken to be $L(f) = \min\{|\Phi| : \Phi \text{ computes } f\}$.

The formal degree of nodes in an arithmetic circuit is defined inductively: all input nodes have formal degree 1, and for addition we take the maximum formal degree of of its inputs. For multiplication we add the formal degrees of its inputs. For a *constant-free* arithmetic circuits the only field constants that are allowed for labels are $\in \{-1, 1\}$. For a polynomial $f \in \mathbb{Z}[X]$, the τ-complexity of f, denoted by $\tau(f)$, is defined to be the size of any smallest constant-free arithmetic circuit computing f, cf. [Bür09, KP11].

We next define Valiant's algebraic complexity classes. A family $\{f_n\}$ of polynomials belongs to VP^0 if there exists a family of constant-free arithmetic circuits $\{\Phi_n\}$ with size and formal degrees polynomially bounded, such that Φ_n computes f_n. Similarly, in case the circuits $\{\Phi_n\}$ are over $\mathbb{F}$, we obtain the class $\mathrm{VP}_{\mathbb{F}}$. The nondeterministic counter parts VNP^0 and $\mathrm{VNP}_{\mathbb{F}}$ of these classes are defined as follows. For polynomials $a(n), b(n)$, VNP^0 is the class of polynomials $\{f_n\}$, for which there exists $\{g_n\} \in \mathrm{VP}^0$ such that $f_n = \sum_{e \in \{0,1\}^{a(n)-b(n)}} g_n(x_1, \ldots, x_{b(n)}, e_1, \ldots, e_{a(n)-b(n)})$. Similarly, if the family $\{g_n\} \in \mathrm{VP}_{\mathbb{F}}$, we obtain $\mathrm{VNP}_{\mathbb{F}}$. We need the following result:

Proposition 1 (Proposition 2.10 in [Bür09]). *Suppose $\tau(\mathrm{per}_n) = n^{O(1)}$. Then for any family $(h_n) \in \mathrm{VNP}^0$, there exists a polynomial $p(n)$ such that $\tau(2^{p(n)} h_n) = n^{O(1)}$.*

For definitions of standard complexity classes like $\mathrm{P}, \mathrm{NP}, \mathrm{PH}$, etc., we refer the reader to the various excellent standard textbooks on complexity theory for a definition. Some of the frequently used classes we will define next. The class of functions $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that there exists a language $A \in \mathrm{P}$ and a polynomial $p(n)$ such that $f(x) = |\{w \in \{0,1\}^{p(|x|)} : (x, w) \in A\}|$ is denoted by #P. The class of function $f - g$, where $f, g \in \#\mathrm{P}$ is denoted by GapP. Valiant [Val79] proved that computing $\mathrm{per}_n(M)$ for M with entries in $\{0,1\}$ over $\mathbb{Z}$ is complete for #P. Toda [Tod91] proved that $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. The majority operator **C.** acting on a complexity class is defined as follows. Given a class $\mathcal{C}$, $\mathbf{C.}\mathcal{C}$ is the class of all languages L for which there exists $L' \in \mathcal{C}$ and a polynomial $p(n)$ such that $x \in L \Leftrightarrow |\{w \in \{0,1\}^{p(|x|)} : (x, w) \in L'\}| > 2^{p(|x|)-1}$. The counting hierarchy, introduced by Wagner [Wag86], is defined to be $\mathrm{CH} := \bigcup_{i \geq 0} \mathrm{C}_i\mathrm{P}$, where $\mathrm{C}_0\mathrm{P} = \mathrm{P}$, and for all $i \geq 1$, $\mathrm{C}_i\mathrm{P} = \mathbf{C.}\mathrm{C}_{i-1}\mathrm{P}$. Note the first level $\mathrm{C}_1\mathrm{P}$ equals PP. Torán [Tor91] characterization of the counting hierarchy states that $\mathrm{C}_{i+1}\mathrm{P} = \mathrm{PP}^{\mathrm{C}_i\mathrm{P}}$, for all $i \geq 0$. An advice function is a function of type $h : \mathbb{N} \rightarrow \{0,1\}^*$. For a complexity class $\mathcal{C}$, define $\mathcal{C}/poly$ to be the class of languages for which there exists $L' \in \mathcal{C}$, and advice function h with $|h(n)| = n^{O(1)}$, such that $x \in L \Leftrightarrow (x, h(|x|)) \in L'$. We use the following lemma, which follows from Lemma 2.6 and Lemma 2.13 in [Bür09].

Lemma 1 ([Bür09])**.** *If $\tau(\mathrm{per}_n) = n^{O(1)}$, then $\mathrm{CH}/poly = \mathrm{P}/poly$.*

We also use the following result:

Lemma 2 (Valiant's Criterion, cf. [Koi11])**.** *Suppose that $p(n)$ is a polynomial, and that for $f : \mathbb{N} \times \mathbb{N} \to \mathbb{Z}$ the map $1^n 0 j \mapsto f(j,n)$, where n is given in unary and j in binary is in $\mathrm{GapP}/poly$. Then the family of polynomials $\{g_n\}$ defined by $g_n(x_1, x_2, \ldots, x_{p(n)}) = \sum_{j \in \{0,1\}^{p(n)}} f(j,n) x_1^{j_1}, x_2^{j_2}, \ldots, x_{p(n)}^{j_{p(n)}}$ is in VNP^0, where j_k is the kth bit of j.*

Next follow some remarks about Boolean circuit classes. AC^0 is the class of all Boolean functions computable by polynomial size constant depth circuits with unbounded fan-in gates in $\{\vee, \wedge, \neg\}$. TC^0 is the class of all Boolean function that can be decided by polynomial size constant depth unbounded fan-in threshold circuits. We sometimes use TC^0 to refer to a *type* of circuit, i.e., constant depth unbounded fan-in threshold circuits, without the size bound implicit. For threshold circuits all gates either compute the negation, or the majority function. NC^1 is the class of all Boolean functions that can be decided by polynomial size $O(\log n)$ depth circuits of bounded fan-in. We have that $\mathrm{AC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1$.

We import some definitions from Ref. [JS11]. We will use the notion of weak-definability, originating from Ref. [KP11, Bür09] (See [JS11] for a discussion of the differences). An integer sequence of bit size $q(n)$ is given by a function $a(n,k)$, such that there exist polynomials $p(n)$ and so that $a(n,k) \in \mathbb{Z}$ is defined for all $n \geq 0$, and all $0 \leq k < 2^{p(n)}$, and where the bit size of $a(n,k)$ is bounded by $q(n)$. We will often write $a_n(k)$ instead of $a(n,k)$. We define the language $uBit(a)$ to be the set of all tuples $(1^n, k, j, b)$ such that the jth bit of $a(n,k)$ equals b. Here k and j are encoded in binary, while 1^n denotes a unary encoding of n. For a sequence $a(n,k)$ and a complexity class $\mathcal{C}$, if $uBit(a) \in \mathcal{C}$, then we say that the sequence $a(n,k)$ is weakly-definable in $\mathcal{C}$.

For the set $\{x_1, x_2, \ldots, x_n\} \cup \{-1, 1\} \cup \{+, \times\}$, we fix some naming scheme that assigns to each element an $O(\log n)$ bit binary string, which is called a *type*. We assume that circuit gates have been labelled by unique binary strings, part of which contains the type. We also assume for the output gate(s) we have fixed a simple naming scheme, where for the ith output i in binary is embedded in the name.

Definition 1 ([JS11])**.** *A representation of a constant-free arithmetic circuit Φ is given by a Boolean circuit C_n that accepts precisely all tuples (t, a, b, q) such that 1) In case $q = 1$ (connection query), a and b are numbers of gates in Φ, b is a child of a, and a has type t. 2) In case $q = 0$ (type query only), a is a number of a gate in Φ, and a is of type t.*

Let $a(n), b(n)$ be two functions. For a family of arithmetic circuits $\{\Phi_n\}$, we say it is $(a(n), b(n))$-succinct, if there exists a non-uniform family of Boolean $\{\vee, \wedge, \neg\}$-circuits $\{C_n\}$, such that C_n represents Φ_n, where for all large enough n, C_n has $\leq a(n)$ inputs and is of size $\leq b(n)$. By convention, if $a(n) = O(\log n)$, we drop it from the notation, and just write $b(n)$-succinct.

The notion of $(a(n), b(n))$-succinct Boolean circuits is defined analogously. In this case types names refer to elements of $\{x_1, x_2, \ldots, x_n\} \cup \{0, 1\} \cup \{\vee, \wedge, \neg, \mathrm{MAJ}\}$. A *poly* size Boolean circuit family $\{C_n\}$ is DLOGTIME-uniform, if given (n, t, a, b, q) with n in binary, we can answer the queries of Definition 1 in time $O(\log n)$ on a Turning machine. Note that if a Boolean circuit family

$\{C_n\}$ is DLOGTIME-uniform, then it is $O(\log n)$-succinct. For the rest of the paper, when we speak about a uniform circuit complexity class $\mathcal{C}$, it is intended to mean DLOGTIME-uniform $\mathcal{C}$.

For ITERATED INTEGER MULTIPLICATION the problem is, given n integers $A_1, A_2, \ldots, A_n$ of n bits each, to compute the bits of $A_1 A_2 \ldots A_n$. Hesse, Allender and Barrington [HAB01] prove uniform TC^0 circuits can solve this problem. The analogous problem of ITERATED INTEGER ADDITION can also be done in uniform TC^0, cf. [Vol99]. Zankó [Zan91], cf. [All99] improves Valiant's completeness to shows that $0,1$-per_n over $\mathbb{Z}$ is complete for #P under DLOGTIME uniform-AC^0 reductions. The following result is proved in [JS11] using Ref.[HAB01, Zan91]:

Proposition 2 ([JS11]). *For any $F \in \mathrm{GapP}$ there exists constants d', d'' and $c' \geq 1$, such that for any $c_0, d \in \mathbb{N}$ and $\gamma \in \mathbb{R}$, if $\{\mathrm{per}_n\}$ can be computed by $n^{1/\gamma}$-succinct size n^{c_0} depth d constant-free arithmetic circuits, then F can be computed by $(O(c' c_0 \log n), n^{c'/\gamma})$-succinct depth $d \cdot d'' + d'$ TC^0 circuits of size at most $n^{c' c_0}$.*

Finally, we need some simple fact about the elementary symmetric polynomial in n variables of degree d defined by $S_n^d = \sum_{I \subseteq [n], |I| = d} \prod_{i \in I} x_i$.

Lemma 3. *There exist uniform TC^0 circuits $\{C_n\}$ of poly(n,m) such that C_n has n input arrays of m bits, and one array of $\lfloor \log n \rfloor + 1$ bits, such that for any non-negative m-bit integers $a_1, a_2, \ldots, a_n$ and $0 \leq d \leq n$ of at most $\lfloor \log n \rfloor + 1$ bits, $C_n(a_1, a_2, \ldots, a_n, d)$ outputs $S_n^{n-d}(a_1, a_2, \ldots, a_n)$.*

Proof. The proof of this is similar to Corollary 3.12 in Ref.[Bür09]. For some t, consider $\prod_{r=0}^{n}(2^t + a_i) = \sum_{r=0}^{n} S_n^r(a_1, a_2, \ldots, a_n)(2^t)^{n-r}$. For any d, we can bound $|S_n^d(a_1, a_2, \ldots, a_n)| < 2^{(m+1)n}$. Hence if we take $t = 2(m+1)n$ in the above, for every r, the bits of $S_n^r(a_1, a_2, \ldots, a_n)$ can be read off from $\prod_{r=0}^{n}(2^t + a_i)$. To compute this product we can use the uniform TC^0 circuits for iterated integer multiplication of Ref. [HAB01]. The difference $n - d$ can be computed in uniform TC^0. We can easily add uniform AC^0 circuits to this for multiplexing the output dependent on $n - d$. □

3 Lower Bounds from Derandomization of Univariate ACIT

For $\mathcal{H}_n \subset \mathbb{Z}$, we say it is encoded by a Boolean circuit C_n with $s(n)$ many inputs if $\mathcal{H}_n \subseteq \{C_n(a) : a \in \{0,1\}^{s(n)}\}$, where we use standard binary representation of integers. More generally, we say that the family $\{C_n\}$ encodes $\{\mathcal{H}_n\}$, if this holds for all but finitely many n. In this situation, we can fix an integer sequence $a_n(i)$ defined for $0 \leq i < 2^{s(n)}$, which we say is associated to $\{\mathcal{H}_n\}$, by taking $a_n(i) = C_n(i)$. Note that if $t(n)$ bounds the number of outputs gates of C_n, then we have that elements of $\mathcal{H}_n$ are at most $t(n)$ bits long, and $a_n(i)$ is an integer sequence of bit length $t(n)$. In particular this holds if C_n has size at most $t(n)$ (where we also count input gates). We say a set $\mathcal{H} \subseteq \mathbb{F}^n$ is a hitting set against some class of polynomials $\mathcal{C}$ in n variables, if for every nonzero $f \in \mathcal{C}$, there exists $h \in \mathcal{H}$ with $f(h) \neq 0$.

Hypothesis 1 (Formal Statement). *There exist $d \in \mathbb{N}$ and a nondecreasing function $s(n) : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ with $s(n) = O(n)$, such that for every $\epsilon > 0$ with $1/\epsilon \in \mathbb{N}$, there exists[4] a family $\{\mathcal{H}_n^\epsilon\}$ of subsets of $\mathbb{Z}^+$ encoded by $(O(n^\epsilon), O(n^\epsilon))$-succinct TC^0 circuits of size $2^{O(n^\epsilon)}$ and depth d with $s(\lceil n^\epsilon \rceil)$ many variable inputs. Furthermore, it holds for infinitely many $n \in \mathbb{N}$ that*

[4] The assumption of non-negativity can be made at an ignorable expense. We also remark that we prefer to state the hypothesis in its weakest form using succinct TC^0 circuits for encoding the hitting set. One may replace this by the stronger condition that asks for uniform TC^0 circuits of size $2^{O(n^\epsilon)}$ and depth d with $s(\lceil n^\epsilon \rceil)$ many variable inputs.

- *for any nonzero polynomial $f(x)$ of degree at most $2^{s(\lceil n^\epsilon \rceil)}$ computed by a constant-free arithmetic circuit of size n over a single variable x, there exist $a \in \mathcal{H}_n^\epsilon$ such that $f(a) \neq 0$.*

For $s(n) = n$, our hypothesis is implied by super-polynomial lower bounds on Boolean circuit size for the Permanent - we give a proof of this in Section 4. This gives strong evidence for the plausibility of our hypothesis.

Also, our hypothesis follows for any function $s(n)$ with $s(n) = \omega(\log n)$ and $s(n) = O(n)$, from the Shub-Smale τ-conjecture [SS95]. For a univariate polynomial f, let $Z(f)$ denote the set of roots of f. According to the τ-conjecture, there exists an absolute constant $c > 0$, so that for all $f \in \mathbb{Z}[x]$, $|Z(f) \cap \mathbb{Z}| \leq (1 + \tau(f))^c$. If the latter is true, then we know that $\mathcal{H}_n = \{0, 1, \ldots, (n+1)^c + 1\}$ is a hitting set against size n constant-free arithmetic circuits, where we do not even use the given degree bound. For each ϵ, we can easily encode $\{\mathcal{H}_n\}$ by circuits computing the identity mapping on $s(\lceil n^\epsilon \rceil) = \omega(\epsilon \log n)$ bits. Ref. [Bür09] shows that the τ-conjecture implies that $\tau(\mathrm{per}_n) \neq n^{O(1)}$. The main result of this section (Theorem 4 below) strengthens this implication by showing that the same lower bound follows from Hypothesis 1.

Another observation is that without the succinctness condition on the circuits computing the hitting set, the above hypothesis would be easy to prove. To give an extreme example for $s(n) = 0$, by counting we know there exist singleton sets $\mathcal{H}_n = \{a_n\}$, where a_n has bit size n^3, such that for every $\epsilon > 0$, for all large enough n, for any nonzero polynomial $f(x)$ of degree[5] at most 1 computed by a constant-free circuit of size at most n, it holds that $f(a_n) \neq 0$. This collection $\{\mathcal{H}_n\}$ can obviously be encoded by non-uniform TC^0 circuits of size n^3 (with no variable inputs), but the problem is that Hypothesis 1 is asking for a succinct encoding of $\{\mathcal{H}_n\}$, so this does not establish the $s(n) = 0$ case. We note that in case $(n!)$ is ultimately hard in the sense of Ref. [SS95], it is straightforward to get the hypothesis for $s(n) = 0$. Recall we say $n!$ is ultimately hard, if for any sequence (a_n) of nonzero integers, $\tau(a_n \cdot n!)$ is not *polylog*(n) bounded. Ref. [SS95] shows that if $(n!)$ is ultimately hard to compute, then one has the separation $\mathrm{P}_\mathbb{C} \neq \mathrm{NP}_\mathbb{C}$ for the Blum-Shub-Smale model. We have the following proposition:

Proposition 3. *If $(n!)$ is ultimately hard, then Hypothesis 1 holds for $s(n) = 0$.*

Proof. Let $\{C_m\}$ be the uniform family of TC^0 circuits for iterated multiplication of Ref. [HAB01]. Let d be the depth of these circuits. Let $\epsilon > 0$ with $1/\epsilon \in \mathbb{N}$ be given. Define the integer sequence $t_n = (2^{\lceil n^\epsilon \rceil}!)$. We can easily compute t_n by uniform TC^0 circuits of size $2^{O(n^\epsilon)}$ and depth $O(d)$ (not depending on ϵ) with only constant inputs as as follows. Namely, for the first layer we enumerate all numbers $1, 2, \ldots, 2^{\lceil n^\epsilon \rceil}$ in binary, and we multiply these by adding below this the appropriate circuit from the family $\{C_m\}$. Note t_n has bit length $2^{O(n^\epsilon)}$. Suppose, for all large enough n, there exists nonzero $f_n(x) = a_n x - b_n$ that is computed by a size n constant-free arithmetic circuit, such that $f_n(t_n) = 0$. Note that $\tau(b_n) \leq n$ (set $x = 0$ in the circuit for f_n). This means that $\tau(a_n \cdot t_n) \leq n$. By our assumption, for some function $g(m) \in \omega(1)$, $\tau(c_m \cdot m!) \geq (\log m)^{g(m)}$, for any sequence (c_m). Hence $\tau(a_n \cdot t_n) \geq (\lceil n^\epsilon \rceil^{g(m)}) = n^{\omega(1)}$. We have reached a contradiction. □

As remarked on before, perhaps the most striking aspect of our hypothesis is that it ask for a hitting set of size at most $2^{s(\lceil n^\epsilon \rceil)}$, where we know that *any* set of size $2^{s(\lceil n^\epsilon \rceil)} + 1$ is a hitting set. Despite this seeming weakness, we show that the hypothesis is sufficient for deducing the following strong lower bound for permanent:

[5] We can observe this irrespective of the degree of f.

Theorem 4 (Theorem 1 restated)**.** *If Hypothesis 1 is true, then* $\tau(\mathrm{per}_n) \neq n^{O(1)}$*.*

Proof. Suppose for all large enough n, $\tau(\mathrm{per}_n) \leq n^{c_0}$, for some constant c_0. Assume that Hypothesis 1 is true, let $d \in \mathbb{N}$ be the fixed number given there, and choose arbitrary $\epsilon > 0$ with $1/\epsilon \in \mathbb{N}$. We will argue that we can derive a contradiction, provided ϵ was chosen small enough. Let $m = m(n) = n^{1/\epsilon}$. Let $a_n(i)$ be the integer sequence associated to $\{\mathcal{H}_n^\epsilon\}$ given by Hypothesis 1. Then for all but finitely many n, $a_n(i)$ has bit size at most $2^{O(n^\epsilon)}$ and is defined for $0 \leq i < 2^{s(\lceil n^\epsilon \rceil)}$. We have that $a_m(i)$ is of bit size $2^{O(n)}$ and defined for $0 \leq i < 2^{s(n)}$. Let

$$f_n = \prod_{0 \leq i < 2^{s(n)}} (x - a_m(i)).$$

Lemma 4. *We have that* $f_n = \sum_{i=0}^{2^{s(n)}} b_n(i) x^i$*, where it holds that the coefficient* $b_n(i)$ *equals* $(-1)^i \cdot S_{2^{s(n)}}^{2^{s(n)}-i}(a_m(0), a_m(1), \ldots, a_m(2^{s(n)} - 1))$*, and where furthermore it holds that*

- *For* $b_n(i)$ *we can state a bound of* $2^{O(n)}$ *on the bit size, where the latter bound does not depend on* ϵ*.*

- $b_n(i)$ *is weakly-definable in* P*/poly, where the magnitude of the corresponding circuit bound does not depend on* ϵ*.*

Proof. The first two claims of the lemma are obvious. Next we will argue the last item. First we construct TC^0 circuits for computing $b_n(i)$.

Claim 1. *There exist* TC^0 *circuits* $\{D_n\}$ *such that*

- D_n *has* $s(n) + 1$ *inputs, and on input* $0 \leq i \leq 2^{s(n)}$ *in binary,* $D_n(i) = b_n(i)$*.*

- $|D_n| = 2^{O(n)}$ *and depth*$(D_n) = O(1)$*. Furthermore, these depth and size bounds are independent*[6] *of* ϵ*.*

- $\{D_n\}$ *is represented by a family of Boolean circuit* $\{B_n\}$*, where* $|B_n| = O(n)$*. Furthermore, the latter stated size bound is independent of* ϵ*.*

Proof. Let us describe the circuits D_n. For the first part it consists of $2^{s(n)}$ copies of TC^0-circuits computing $a_m(i)$, for $0 \leq i < 2^{s(n)}$, with i in binary hardcode in each copy. Each copy is of size at most $2^{O(m^\epsilon)} = 2^{O(n)}$ and depth d. Clearly, this $2^{O(n)}$ size bound and depth bound of d do not depend on ϵ. Each copy can be described by a Boolean circuit with the number of inputs and size bounded by $O(m^\epsilon) = O(n)$. Obtaining a representation for this first part of the circuit is done by adding $s(n) = O(n)$ bits to gate names. We can easily obtain a representation with size and number of inputs bounded by $O(n)$.

Let $\{C_n\}$ be the uniform TC^0 circuits from Lemma 3 for computing elementary symmetric polynomials, where we have catered for enough inputs bits so that $C_{2^{s(n)}}$ is able to receive all $a_m(i)$'s as inputs. For the second part of D_n we use $C_{2^{s(n)}}$. $C_{2^{s(n)}}$ has $2^{s(n)} + 1$ inputs, which are fed in the bits of the $2^{s(n)}$ numbers $a_m(i)$ for $0 \leq i < 2^{s(n)}$, each of size $2^{O(n)}$, and i of $s(n) + 1$ bits. We can give a size bound of $poly(2^{s(n)}, 2^{O(n)}) = 2^{O(n)}$ and depth bounds of $O(1)$ for $C_{2^{s(n)}}$.

[6] The circuits D_n and B_n themselves may very well depend on ϵ, but all we need for our argument is that the given size and depth bounds do not.

The uniformity implies that we have a Boolean circuit with number of inputs and size bounded by $O(n)$ representing $C_{2^{s(n)}}$. None of these bounds depend on ϵ. Finally, we fix the sign bit of the output to take account of the $(-1)^i$ factor. This is easily done by letting the sign bit of the output equal the least significant bit of i. Clearly, the circuit D_n we have described computes $b_n(i)$.

We can easily merge the representations of the first and second part of the circuit D_n to get a Boolean circuit with the number of inputs and size bounded by $O(n)$. It is also clear that neither the given size and and depth bounds for D_n or the size of its representation are dependent on ϵ. □

We can now use the circuit families $\{D_n\}$ and $\{B_n\}$ from Claim 1 to do a 'scaling-up' to CH/*poly* argument, where B_n will be given as advice. Since we start with size $2^{O(n)}$ depth $O(1)$ TC^0 circuits and advice $O(n)$, where all these bounds are independent of ϵ, there will be no dependency on ϵ for the end result. From our assumption for permanent we get the collapse CH/*poly* = P/*poly* (again independent of ϵ). Hence we will get polynomial size Boolean circuits computing $b_n(i)$, for which we can give a size bound that is independent of ϵ. We give the details in the next subsection.

3.1 Scaling-Up Argument

Let $\{D_n\}$ be the circuit family provided by Claim 1. The family $\{D_n\}$ is $(O(n), O(n))$-succinct. Let $\{B_n\}$ be the corresponding family of Boolean circuits of with number of inputs and size bounded by $O(n)$, where B_n represents D_n. In this representation names of gates in D_n are $O(n)$ bits long. Wlog. assume that we have constant $c \in \mathbb{N}$ such that gate names of D_n are exactly cn bits long. Let d' be the depth of D_n. For $0 \leq r \leq d'$, let L_r be the language of tuples $(G, 1^n, i, b)$ for which

- G is the name of a gate on level r in D_n. It outputs b when D_n is given input i in binary.
- The input i is given in binary and satisfies $0 \leq i \leq 2^{s(n)}$.

Claim 2. *For each $0 \leq r \leq d'$, $L_i \in$ CH/poly.*

Proof. We will prove the claim by induction on r. It is easy to check the input format (e.g. for technical convenience one may assume $s(n)$ in unary is given as advice of length padded to $O(n)$). So we assume that the input is of form $(G, 1^n, i, b)$ with G of cn bits, i a $s(n)+1$ bit number, and $b \in \{0, 1\}$. We assume that B_n is given as advice for this input length. In the following argument we make connection and type queries for the circuit D_n by evaluating B_n on certain inputs. Since the circuit value problem is in P, and $|B_n| = O(n)$ any such queries take time *poly*(n).

For the base case $r = 0$, first we use B_n to check that G is a correct gate name, by making $O(1)$ type queries. It is easy to check whether the gate G is labelled by a variable, since for a gate labelled by a variable x_ℓ, ℓ in binary is part of the gate name. Then one just need to fetch the ℓth bit of i. Gates labelled by Boolean constants are dealt with even more easily as these constants appear in the gate name itself. To check whether G is on level 0 we can assume wlog.[7] this information can be obtained from the gate name. We can easily do all of the above computation in polynomial time using the advice B_n, for some polynomial not depending on ϵ. By attaching a clock to this computation we can ensure the run-time is independent of ϵ for all inputs.

Now assume the claim hold for L_r. By Torán [Tor91] characterization of the counting hierarchy it suffices to show that $L_{r+1} \in \mathrm{PP}^{L_r}/poly$, This is done as follows. Given input $(G, 1^n, i, b)$, we

[7] Alternatively, one can add another level of oracle calling to the argument by making existential queries to B_n of the form "Does there exist H such that G is a child of H?".

assume the gate G is of majority type. Negation gates are handled similarly. Let N be a NTM that on input $(G, 1^n, i, b)$ nondeterministically guesses the cn size name of a gate H, uses the advice B_n to check that $H \to G$ is a wire in D_n. If this is not true, nondeterministically flip a bit b' and accept if $b' = 1$, reject if $b' = 0$. Otherwise, query $(H, 1^n, i, b) \overset{?}{\in} L_i$. Accept if the answer to this query is yes, reject otherwise. Observe that N accepts on the majority of its nondeterministic guesses iff the majority of the inputs to G are outputting b in $D_n(e, j)$. Similarly as before, evaluation of B_n can be done in time $poly(|B_n|) = poly(n)$, where this upper bound does not depend on ϵ. By attaching a clock, we get a computation running in nondeterministic $poly(n)$ time, independent of ϵ. This shows that $L_{i+1} \in \mathrm{PP}^{L_i}/poly$, and proves the claim. □

By the above claim, $L_{d'} \in \mathrm{CH}/poly$, and from the proof we see that all underlying machines (according to Torán's [Tor91] characterization) have running time independent of ϵ, and furthermore d' itself is independent of ϵ. The advice possibly depends on ϵ, but the amount of advice does not (we can pad out the representing circuits for which we have a size bound $O(n)$ independent of ϵ).

Since we assume $\tau(\mathrm{per}_n) \leq n^{c_0}$ and by Lemma 1, we have that $\mathrm{CH}/poly = \mathrm{P}/poly$. Hence $L_r \in \mathrm{P}/poly$. Since c_0 does not depend on ϵ and by remarks in the previous line we get polynomial size Boolean circuits for L_r, whose size does not depend on ϵ. This is easily seen to imply that $b_n(i)$ is weakly-definable in $\mathrm{P}/poly$, where the magnitude of the corresponding circuits bound does not depend on ϵ. We have completed the proof of Lemma 4. □

3.2 Finishing Up: Valiant's Criterion & Completeness of Permanent

Let e be an absolute constant such that $b'_n(i)$ has bit length at most 2^{en}, plus a sign bit $s_n(i)$. Write $b'_n(i) = (-1)^{s_n(i)} \sum_{j=0}^{2^{en}} b'_n(i)_j 2^j$. Then $f_n = \sum_{i=0}^{2^{s(n)}} (-1)^{s_n(i)} \sum_{j=0}^{2^{en}} b'_n(i)_j 2^j x^i$. Take

$$h_n(y_1^{,} \dots y_\ell, z_1, \dots z_\ell) \;=\; \sum_{i=0}^{2^{s(n)}} \sum_{j=0}^{2^{en}} (-1)^{s_n(i)} b'_n(i)_j y_1^{i_1} \dots y_\ell^{i_\ell} z_1^{j_1} \dots z_\ell^{z_\ell},$$

where $\ell = \max(s(n) + 1, en + 1)$. We have that $f_n = h_n(x^{2^0}, x^{2^1}, \dots, x^{2^\ell}, 2^{2^0}, 2^{2^1}, \dots, 2^{2^\ell})$. We also have that h_n is in VNP^0. Namely, Lemma 4 gives us that we have Boolean circuits of size $poly(n)$ (not depending on ϵ) for computing the jth bit of $(-1)^{s_n(i)} b'_n(i)$ given i and j in binary. An application of Valiant's criterion (Lemma 2) then gives that $h_n \in \mathrm{VNP}^0$.

By Proposition 1, for some polynomials $p(n)$ and $q(n)$, $\tau(2^{p(n)} h_n) \leq q(n)$. Furthermore, since the constant c_0, ℓ and the size of the circuits for computing $(-1)^{s_n(i)} b'_n(i)$ do not depend on ϵ, we get that $p(n)$ and $q(n)$ do not depend on ϵ. To compute the powers of 2 and x can be done with $\ell = O(n)$ operations. We conclude that for all but finitely many n, $f'_n := 2^{p(n)} f_n$ can be computed by constant-free arithmetic circuits of size $r(n)$, for some fixed polynomial $r(n)$ not depending on ϵ. We also have that f'_n is a nonzero polynomial of degree $2^{s(n')}$ that vanishes on the set $\{a_m(i) : 0 \leq i < 2^{s(n')}\}$. The latter set includes $\mathcal{H}_m^\epsilon$. This means that for infinitely many n, f'_n requires constant-free arithmetic circuits of size $\lfloor n^{1/\epsilon} \rfloor$. We can choose $\epsilon > 0$ small enough, so that for all large enough n, $r(n) < \lfloor n^{1/\epsilon} \rfloor$. We have reached a contradiction. □

There is some room in this proof for getting different randomness to hardness trade-offs. For example, for obtaining quasi-polynomial lower bounds for Permanent one can straightforwardly modify the proof of Theorem 4 to yield the following theorem:

Theorem 5. *Suppose there exist $d \in \mathbb{N}$ and a nondecreasing function $s(n) : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ with $s(n) = O(n)$, such that for every $\epsilon > 0$ with $1/\epsilon \in \mathbb{N}$, there exists a family $\{\mathcal{H}_n^\epsilon\}$ of subsets of $\mathbb{Z}^+$ encoded by $(O(2^{\log^\epsilon n}), O(2^{\log^\epsilon n}))$-succinct* TC^0 *circuits of size $2^{O(2^{\log^\epsilon n})}$ and depth d with $s(2^{\lceil \log^\epsilon n \rceil})$ many inputs. Furthermore, suppose it holds that for infinitely many n that for any nonzero polynomial $f(x)$ of degree at most $2^{s(2^{\lceil \log^\epsilon n \rceil})}$ computed by a constant-free arithmetic circuit of size n over a single variable x, there exist $a \in \mathcal{H}_n^\epsilon$ such that $f(a) \neq 0$. Then there does not exist k, such that $\tau(\mathrm{per}_n) = 2^{O(\log^k n)}$.*

3.3 Generalization to Circuits with Arbitrary Constants

So far the focus has been on constant-free circuits, but using a result by Bürgisser [Bür00], we can generalize the randomness-to-hardness theorem to the setting where circuits use arbitrary constants from $\mathbb{F}$. The result of Ref. [Bür00] assumes the Generalized Riemann Hypothesis (GRH). We have the following theorem. Note that the derandomization condition posed is as in Hypothesis 1, but with the hitting set required to work against univariate circuits using constant from $\mathbb{F}$ of size n.

Theorem 6. *We assume (GRH). Let $\mathbb{F}$ be a field of characteristic zero. Suppose there exist $d \in \mathbb{N}$ and a nondecreasing function $s(n) : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ with $s(n) = O(n)$, such that for every $\epsilon > 0$ with $1/\epsilon \in \mathbb{N}$, there exists a family $\{\mathcal{H}_n^\epsilon\}$ of subsets of $\mathbb{Z}^+$ encoded by $(O(n^\epsilon), O(n^\epsilon))$-succinct* TC^0 *circuits of size $2^{O(n^\epsilon)}$ and depth d with $s(\lceil n^\epsilon \rceil)$ many inputs. Furthermore, suppose it holds that for infinitely many n that for any nonzero polynomial $f(x)$ of degree at most $2^{s(\lceil n^\epsilon \rceil)}$ computed by an arithmetic circuit over $\mathbb{F}$ of size n over a single variable x, there exist $a \in \mathcal{H}_n^\epsilon$ such that $f(a) \neq 0$. Then $\{\mathrm{per}_n\} \notin \mathrm{VP}_{\mathbb{F}}$.*

Proof. For purpose of contradiction suppose that the preconditions as stated in the theorem are satisfied, but that $\{\mathrm{per}_n\} \in \mathrm{VP}_{\mathbb{F}}$. Corollary 1.2 in [Bür00] shows that the latter condition implies that $\#\mathrm{P}/\mathit{poly} = \mathrm{FP}/\mathit{poly}$, provided (*GRH*) is true. This implies that $\mathrm{CH}/\mathit{poly} = \mathrm{P}/\mathit{poly}$. We can now proceed exactly as in the proof of Theorem 4 to define f_n of degree $2^{s(n')}$ that requires univariate circuits of size $\lfloor n^{1/\epsilon} \rfloor$ over $\mathbb{F}$. Leveraging the $\mathrm{CH}/\mathit{poly} = \mathrm{P}/\mathit{poly}$ collapse after the scaling to CH argument just as before, we get that the coefficients of f_n are integers computable by Boolean circuits of polynomial size. By Valiant's Criterion over $\mathbb{F}$, this puts $f_n \in \mathrm{VNP}_{\mathbb{F}}$. Since we are assuming that $\{\mathrm{per}_n\} \in \mathrm{VP}_{\mathbb{F}}$ we get that $\mathrm{VNP}_{\mathbb{F}} = \mathrm{VP}_{\mathbb{F}}$. Hence we get polynomial size arithmetic circuits for f_n over $\mathbb{F}$. Just as before, this upper bound can be seen to be independent of ϵ, which is a contradiction, provided ϵ was chosen large enough. □

Note that for the arbitrary constants model it is only interesting to consider the setting where $s(n) = \omega(\log n)$. For example, for $s(n) = O(\log n)$, for any $h_1, h_2, \ldots, h_t \in \mathbb{F}$ with $t = 2^{s(\lceil n^\epsilon \rceil)} = n^{O(\epsilon)}$, $(x - h_1)(x - h_2) \ldots (x - h_t)$ can be computed by a size $n^{O(\epsilon)}$ arithmetic circuit over $\mathbb{F}$.

4 Deriving Hypothesis 1 from Boolean Circuit Lower Bounds for Permanent

In this section, we show that Hypothesis 1 can be derived from a Boolean circuit lower bound for Permanent. We divide the proof into two parts. First, we show that if Permanent does not have polynomial-size Boolean circuits, then there is a pseudo-random generator computable by sub-exponential size TC^0 circuits which fools Boolean circuits. Then we show that a pseudo-random generator fooling Boolean circuits can be viewed as a hitting set against the class of univariate polynomials of sub-exponential degree computable by small constant-free arithmetic circuits.

For the first part, we mostly give proof sketches rather than proofs because the arguments follow along standard lines.

Our pseudo-random generator will be based on the worst-case hardness of the following problem. One could equally well consider other versions of the Permanent, such as computing the permanent of a general integer matrix, and derive the same consequence, but we focus on this one for concreteness.

Definition 2. *(0,1)-Permanent is the following computational problem: the input is an $N \times N$ matrix with (0,1)-entries, represented by a bitstring of size N^2, and the output is the permanent of the input matrix over the ring $\mathbb{Z}$.*

Lemma 5. *If (0,1)-Permanent cannot be computed by polynomial-size Boolean circuits, then there exists a constant c and a language $L \in \mathrm{PP}$ such that no polynomial-size family of Boolean circuits decides L correctly on a $1 - 1/n^c$ fraction of inputs for all input lengths n.*

Proof Sketch. Assume (0,1)-Permanent cannot be computed by polynomial-size Boolean circuits. Then, by random self-reducibility of Permanent [BF90, Lip90], there is a constant d such that for an appropriately chosen decision version L' of Permanent (eg. ModPerm [IW98]), no polynomial-size family of Boolean circuits decides L' correctly on more than a $1 - 1/n^d$ fraction of inputs. But $L' \in \mathrm{P}^{\mathrm{PP}}$, so let L be the PP language to which L' is polynomial-time reducible. It follows that no polynomial-size family of Boolean circuits computes L correctly on more than $1 - 1/n^c$ fraction of inputs of length n, where c is a constant which depends on d and the number of queries made to L by the polynomial-time oracle machine deciding L'. □

Now, we can use Yao's XOR Lemma [Lev87] to amplify the hardness of the PP language. We state the XOR Lemma in a somewhat weaker form than usual which is sufficient for our purposes.

Theorem 7. *[Lev87] Let L be a language for which there exists a constant c such that no polynomial-size family of circuits decides L correctly on more than a $1-1/n^c$ fraction of inputs for all input lengths n. Given a polynomial p define the language $XOR - L_p$ as follows: the language consists of all tuples $< x_1, x_2 \ldots x_{p(n)} >$ where $|x_i| = n$ for each i and an odd number of elements of the tuple belong to L. Then there exists a polynomial p such that no polynomial-size family of circuits decides $XOR - L_p$ correctly on more than a $1/2 + 1/m^2$ fraction of inputs for each input length m.*

Lemma 6. *If there is a language $L \in \mathrm{PP}$ for which there is a constant c such that no polynomial-size family of circuits decides L correctly on more than a $1 - 1/n^c$ fraction of inputs of length n for each integer n, then there is a language $L' \in \mathrm{PP}$ such that no polynomial-size family of circuits decides L' correctly on more than a $1/2 + 1/m^2$ fraction of inputs of length m for each integer m.*

Lemma 6 follows from Lemma 7 simply by choosing $L' = XOR - L_p$ for an appropriate polynomial p. By the result of Fortnow and Reingold [FR91] that PP is closed under truth-table reductions, if $L \in \mathrm{PP}$, it follows that $L' \in \mathrm{PP}$.

Next, we will show that if the Permanent is hard, then there is a pseudo-random generator computable by uniform subexponential-size threshold circuits which fools Boolean circuits of polynomial size. We will need an efficiently computable version of the Nisan-Wigderson generator. We first define pseudo-random generators against Boolean circuits.

Definition 3. *Given functions $l, s : \mathbb{N} \rightarrow \mathbb{N}$, an infinitely-often pseudo-random generator (i.o.PRG) with seed length l against Boolean circuits of size s is a sequence of functions $\{G_n\} : \{0,1\}^{l(n)} \rightarrow \{0,1\}^n$ such that for any family $\{C_n\}$ of circuits with $|C_n| \leq s(n)$, for infinitely many n, $|\underset{x \in \{0,1\}^n}{Pr} C(x) - \underset{y \in \{0,1\}^{s(n)}}{Pr} C(G(y))| \leq 1/n$. Given a complexity class* C*, we say a PRG G is computable within* C *if the language $\{< 1^n, y, i > ||y| = l(n), G(y)_i = 1\}$ belongs to* C.

Theorem 8. *If (0,1)-Permanent cannot be computed by polynomial-size Boolean circuits, then for each constant $\epsilon > 0$, there is an i.o.PRG with seed length $O(n^\epsilon)$ against Boolean circuits of size n^4 which is computable by uniform constant-depth threshold circuits of size $2^{n^{O(\epsilon)}}$.*

Proof Sketch. Assume (0,1)-Permanent cannot be computed by polynomial-size Boolean circuits. Then, by Lemma 5 and Lemma 6, it follows that there is a language $L \in$ PP such that no polynomial-size family of Boolean circuits computed L correctly on more than a $1/2 + 1/m^2$ fraction of inputs of length m for all integers m.

The black-box pseudo-random generator construction of Nisan and Wigderson [NW94] together with the efficient design construction of Viola [Vio04] yields generators from n^ϵ bits to n bits computable by constant-depth *oracle* circuits of size $2^{O(n^\epsilon)}$ making oracle queries of size at most n^ϵ such that whenever a language L which is strongly average-case hard against polynomial-size Boolean circuits is used as the oracle, the resulting generator works infinitely often against Boolean circuits of any fixed polynomial size, as long as ϵ is small enough. Now, if we plug in the $L \in$ PP which is strongly average-case hard, by using the fact that any $L \in$ PP is computable by uniform constant-depth threshold circuits of size 2^{n^c} for some constant c, we get an i.o.PRG with seed length n^ϵ against Boolean circuits of size n^4 computable by uniform constant-depth threshold circuits of size $2^{n^{O(\epsilon)}}$. □

Now we show how to interpret PRGs against Boolean circuits as hitting sets against univariate polynomials of not too large degree computed by small constant-free arithmetic circuits.

Theorem 9. *Let $0 < \epsilon < 1$ be any constant. If G_n is an i.o.PRG with seed length n^ϵ against Boolean circuits of size n^4, then by interpreting the output of G as the binary representation of an integer, the range of G is a hitting set of size at most 2^{n^ϵ} for infinitely many n against univariate polynomials with degree at most 2^{n^ϵ} that are computable by size n constant-free arithmetic circuits.*

Proof. Suppose otherwise, and let f_n be a sequence of univariate polynomials of degree at most 2^{n^ϵ} and computable by constant-free arithmetic circuits D_n of size at most n such that for all but finitely many n, f_n is not identically zero and yet evaluates to zero on all elements of the range of G_n. We will show how to define a sequence of circuits $\{C_n\}$ such that for each n, $|C_n| \leq n^4$ and for all but finitely many n, the acceptance probability of C_n with respect to the uniform distribution on inputs differs from the probability with respect to the uniform distribution on the range of G_n by at least $1/n$.

The circuit C_n simply evaluates the small arithmetic circuit for f_n modulo a certain prime p_n of size n^2, and accepts iff the output is 0. We will describe how p_n is chosen later. Note that if the arithmetic circuit has size at most n, then C_n can be implemented in size at most $n^3 polylog(n)$, which is at most n^4 for large enough n.

The sequence of primes $\{p_n\}$ we choose will have the following property: For every integer x in the range of G_n, if $D_n(x)$ is non-zero, then so is $D_n(x)$ mod p_n. We will only argue that the primes p_n exist - they can then be hard-coded into the circuit C_n. The argument is via the probabilistic method. Given a non-negative integer $y < 2^n$, $D_n(y)$ cannot be larger than 2^{n2^n}. Therefore, for a random prime p_n of bitsize n^2, the probability that p divides $D_n(y)$ is at most $2^{n+O(\log(n))-n^2}$ - here we use the Prime Number theorem. By a union bound, the probability that there exists a $y \in \{0,1\}^n$ in the range of G_n for which $D_n(y)$ is non-zero but $D_n(y)$ mod p_n is zero is at most $2^{2n+O(\log(n))-n^2}$ which is less than 1 when n is large enough. Thus there must exist a p_n for which the desired property holds - this is the prime we hard-code into the circuit C_n.

To complete the argument, we need to show that C_n can distinguish the uniform distribution on n bits from the uniform distribution on the range of G_n for all but finitely many n, assuming that f_n evaluates to zero on all elements in the range of G_n for all but finitely many n. By the assumption on f_n, C_n accepts with probability 1 on the range of G_n for all but finitely many n. Since f_n is of degree at most 2^{n^ϵ}, we have that f_n has at most $2^{n^\epsilon}+1$ integer roots, and therefore f_n is non-zero with probability at least $1/2$ on a random non-negative integer $< 2^n$. By our choice of p_n, C_n rejects whenever f_n is non-zero on a non-negative integer $< 2^n$, thus we have that C_n rejects with probability at least $1/2$ for all but finitely many n. This is a contradiction to the assumption that G_n is an i.o.PRG against Boolean circuits of size n^4. □

Putting together Theorem 8 and Theorem 9, we have the following corollary:

Corollary 1. *If (0,1)-Permanent does not have polynomial-size Boolean circuits, then Hypothesis 1 holds.*

5 Applications

First we prove a lemma:

Lemma 7. *There exists an integer sequence (a_n) of bit size $O(n^3)$, such that (a_n) is weakly-definable in the polynomial hierarchy, and for which the following holds:*

- *For any constant-free arithmetic circuit Φ of size n over a single variable x, if $\Phi(x)$ computes a nonzero polynomial of degree at most one, then $\Phi(a_n) \neq 0$.*

Proof. Define a_n to be the smallest number of n^3 many bits that satisfies $p \cdot a_n + q \neq 0$, for any integers p and q computable by constant-free arithmetic circuit Φ of size $2n+4$. By counting we can bound the number of constant-free arithmetic circuits of size n by $2^{O(n^2)}$, so we know such a_n exists in $\{0,1\}^{n^3}$. Observe that a_n satisfies for any constant-free arithmetic circuit Φ of size n over a single variable x, if $\Phi(x)$ computes a nonzero polynomial $f_n = p_n \cdot x + q_n$ then $f_n(a_n) \neq 0$. Indeed, we can compute $q_n = f_n(0)$ by size at most n, and $p_n = f_n(1) - f_n(0)$ by size at most $2n+4$.

Note that a_n can be computed by a constant-free arithmetic circuits of Ψ size $O(n^3)$, by going over its binary expansion in the obvious way. Let us call this the canonical circuit for a_n. For $a \in \{0,1\}^{n^3}$, define the predicate $R_n(a)$ to be true if "For every constant-free arithmetic circuits Φ_1, Φ_2 of size at most $2n+4$, the circuit $\Phi_1 \cdot \Psi_a + \Phi_2$ is not identically zero", where Ψ_a is the canonical constant-free circuit of size $O(n^3)$ computing a. Testing where $\Phi - \Psi_a \equiv 0$ is an instance of arithmetic circuit identity testing over $\mathbb{Z}$, which is in coRP [IM83]. This implies R_n is a $\mathrm{coNP}^{\mathrm{RP}}$ predicate. By binary search in $\{0,1\}^{n^3}$ making queries of form "$\exists a' < a, R_n(a')$?", a $\mathrm{P}^{\mathrm{NP}^{\mathrm{coNP}^{\mathrm{RP}}}}$ machine can find the lexicographical least number for which R_n holds, i.e. compute a_n. This implies $uBit(a_n)$ is in PH. □

Using Theorem 4 together with the above lemma, we obtain the following theorem (this result immediately implies Theorem 2):

Theorem 10. *One of the following items must be true:*

- *For every integer $d \geq 1$, there exists $\epsilon > 0$ such that $0,1$-permanent can not be computed by (n^ϵ, n^ϵ)-succinct* TC^0 *circuits of size 2^{n^ϵ} and depth d.*
- *$\tau(\mathrm{per}_n)$ is not polynomially bounded.*

Proof. For the purpose of deriving a contradiction, suppose there exist $d \geq 1$, such that for every $\epsilon > 0$, we have a family of (n^ϵ, n^ϵ)-succinct TC^0 circuits of size 2^{n^ϵ} and depth d for computing $0,1$-permanent over $\mathbb{Z}$. Let $a_n(i)$ be the integer sequence given by Lemma 7. By Toda's Theorem and Valiant's completeness result for per_n, since $uBit(a_n) \in \mathrm{PH}$, we get that $uBit(a_n)$ can be decided in polynomial time with one query to the $0,1$-permanent. This is done in three steps: first apply $R_1 \in \mathrm{FP}$ to x, then apply $\mathrm{per}(R_1(x))$. Finally compute $R_2 \in \mathrm{FP}$ to obtain $R_2(\mathrm{per}(R_1(x)))$. Since $\mathrm{FP} \subseteq \#\mathrm{P}$, and due to Proposition 2, for some constant b not depending on ϵ, we obtain $(n^{\epsilon b}, n^{\epsilon b})$-succinct TC^0-circuits for R_1 and R_2 (with depth not depending on ϵ) of size $2^{n^{\epsilon b}}$. Putting all three TC^0 circuits together yields TC^0-circuits for $uBit(a_n)$, where for some constant k not depending on ϵ, this family is $(n^{\epsilon k}, n^{\epsilon k})$-succinct and has size at most $2^{n^{\epsilon k}}$, and whose depth does not depend on ϵ . The constant k can be picked larger than b to deal with the increase in size when joining the three representations.

It is easy to go from a circuit for $uBit(a_n)$ to a circuit computing a_n by having $O(n^3)$ separate copies for each output bit. This kind of duplication can be done by adding $O(\log n)$ bits to gate names, and adding $polylog(n)$ circuitry to the representing circuits. We conclude that there exist a constants $\tilde{k} > 1$ and $\tilde{d} \in \mathbb{N}$ not depending on ϵ, so that for any $\epsilon > 0$, (a_n) can be computed by $(n^{\epsilon \tilde{k}}, n^{\epsilon \tilde{k}})$-succinct TC^0 circuits of size at most $2^{n^{\epsilon \tilde{k}}}$ and depth $\tilde{d}$. The bit size of a_n is $O(n^3)$, which is less than 2^{n^ϵ}, provided n is large enough. This means that Hypothesis 1 is satisfied for depth $\tilde{d}$ and constant function $s(n) = 0$. Therefore, we get that $\tau(\mathrm{per}_n)$ is not polynomially bounded by Theorem 4. □

Finally, we give a simplified proof of Allender's superpolynomial lower bound for the Permanent against uniform TC^0 - in fact, we prove a stronger result. We will need the following proposition, which follows using padding from the standard fact that uniform TC^0 corresponds to the polylogarithmic-time fragment of CH.

Proposition 4 ([All99]). *If* $\mathrm{L} \subseteq \mathrm{TC}^0$*, then* $\mathrm{PSPACE} \subseteq \mathrm{CH}$.

Theorem 11. *Either (0,1)-Permanent* $\notin \mathrm{DSPACE}(n^{o(1)}) \cap \mathrm{P}$ *or* $\mathrm{L} \nsubseteq \mathrm{TC}^0$

Proof. Either PSPACE $\subseteq$ CH or not.

In the first case, we assume (0,1)-Permanent $\in \mathrm{DSPACE}(n^{o(1)}) \cap \mathrm{P}$ and derive a contradiction. If (0,1)-Permanent $\in$ P, then PP $=$ P since (0,1)-Permanent is hard for PP. This implies CH $=$ P. Since PSPACE $\subseteq$ CH, we have that PSPACE $=$ P. Now, we know that (0,1)-Permanent is hard for NP and hence for P. Thus we have that (0,1)-Permanent is hard for PSPACE and now using the assumption that (0,1)-Permanent $\in \mathrm{DSPACE}(n^{o(1)})$, we derive a contradiction to the space hierarchy theorem.

If PSPACE $\nsubseteq$ CH, then by Proposition 4, we immediately have $\mathrm{L} \nsubseteq \mathrm{TC}^0$. □

Corollary 2 ([All99]). *(0,1)-Permanent $\notin TC^0$*

The corollary follows from Theorem 11 simply because $L \subseteq \mathrm{P} \cap \mathrm{DSPACE}(n^{o(1)})$.

References

[Agr05] M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proc. 25th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 92–105, 2005.

[All99] E. Allender. The permanent requires large uniform threshold circuits. *Chicago Journal of Theoretical Computer Science*, 1999. article 7.

[BF90] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *Proc. 7th Annual Symposium on Theoretical Aspects of Computer Science*, volume 415 of *Lect. Notes in Comp. Sci.*, pages 37–48. Springer Verlag, 1990.

[Bür00] Peter Bürgisser. Cook's versus Valiant's hypothesis. *Theor. Comp. Sci.*, 235:71–88, 2000.

[Bür09] P. Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18:81–103, 2009.

[DL78] R. DeMillo and R. Lipton. A probabilistic remark on algebraic program testing. *Inf. Proc. Lett.*, 7:193–195, 1978.

[FR91] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. In *Proc. 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 13–15, 1991.

[HAB01] W. Hesse, E. Allender, and D.A.M. Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Comp. Sys. Sci.*, 64(4):695–716, 2001.

[HS80] J. Heintz and C.P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proc. 12th Annual ACM Symposium on the Theory of Computing*, pages 262–272, 1980.

[IM83] O. Ibarra and S. Moran. Probabilistic algorithms for deciding equivalence of straight-line programs. *J. Assn. Comp. Mach.*, 30:217–228, 1983.

[IW98] R. Impagliazzo and A. Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998. to appear.

[JS11] M. Jansen and R. Santhanam. Permanent does not have succinct polynomial size arithmetic circuits of constant depth. In *Proc. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011)*, pages 724–735, 2011.

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity testing means proving circuit lower bounds. *Computational Complexity*, 13(1–2):1–44, 2004.

[Koi11] P. Koiran. Shallow circuits with high powered inputs. In *Proc. 2nd Symp. on Innovations in Computer Science*, 2011.

[KP11] P. Koiran and S. Perifel. Interpolation in Valiant's theory. *Computational Complexity*, 20(1):1–20, 2011.

[Lev87] L. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[Lip90] R. Lipton. New directions in testing. In J.Feigenbaum and M.Merritt, editors, *Distributed Computing and Cryptography*, pages 191–202. American Mathematical Society, 1990.

[Lip94] R. Lipton. Straight-line complexity and integer factorization. *Algorithmic Number Theory, LNCS 877*, pages 71–79, 1994.

[NW94] N. Nisan and A. Wigderson. Hardness versus randomness. *J. Comp. Sys. Sci.*, 49:149–167, 1994.

[Sch80] J.T. Schwartz. Fast probabilistic algorithms for polynomial identities. *J. Assn. Comp. Mach.*, 27:701–717, 1980.

[SS95] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and and algebraic vesion of "NP $\neq$ P". *Duke Math J.*, 81:47–54, 1995.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20:865–877, 1991.

[Tor91] J. Torán. Complexity classes defined by counting quantifiers. *J. Assn. Comp. Mach.*, 38(3):753–774, 1991.

[Val79] L. Valiant. The complexity of computing the permanent. *Theor. Comp. Sci.*, 8:189–201, 1979.

[Vio04] E. Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3–4):147–188, 2004.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity.* Springer-Verlag, 1999. A uniform approach.

[Wag86] K. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23:325–356, 1986.

[Wil10] R. Williams. Improving exhaustive search implies superpolynomial lower bounds. In *Proc. 42nd Annual ACM Symposium on the Theory of Computing*, pages 231–240, 2010.

[Wil11] R. Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of 26th IEEE Conference on Computational Complexity*, 2011.

[Zan91] V. Zankó. #P-completeness via many-one reductions. *International Journal of Foundations of Computer Science*, 2:77–82, 1991.

[Zip79] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Manipulation (EUROSAM '79)*, volume 72 of *Lect. Notes in Comp. Sci.*, pages 216–226. Springer Verlag, 1979.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de