Electronic Colloquium on Computational Complexity, Report No. 132 (2018)

Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits

Mrinal Kumar* Ramprasad Saptharishi† Anamay Tengse‡

July 17, 2018

Abstract

The classical lemma of Ore-DeMillo-Lipton-Schwartz-Zippel states that any nonzero polynomial $f(x_1, \ldots, x_n)$ of degree at most s will evaluate to a nonzero value at some point on a grid $S^n \subseteq \mathbb{F}^n$ with $|S| > s$. Thus, there is a deterministic polynomial identity test (PIT) for all degree-s size-s algebraic circuits in n variables that runs in time $\text{poly}(s) \cdot (s+1)^n$. In a surprising recent result, Agrawal, Ghosh and Saxena (STOC 2018) showed any deterministic blackbox PIT algorithm for degree-s, size-s, n-variate circuits with running time as bad as $\left(s^{n^{0.5-\delta}}\right) \text{HUGE}(n)$, where $\delta > 0$ and $\text{HUGE}(n)$ is an arbitrary function, can be used to construct blackbox PIT algorithms for degree-s size s circuits with running time $s^{\exp \circ \exp(O(\log^* s))}$.

The authors asked if a similar conclusion followed if their hypothesis was weakened to having deterministic PIT with running time $s^{o(n)} \cdot \text{HUGE}(n)$. In this paper, we answer their question in the affirmative. We show that, given a deterministic blackbox PIT that runs in time $s^{o(n)} \cdot \text{HUGE}(n)$ for all degree-s size-s algebraic circuits over n variables, we can obtain a deterministic blackbox PIT that runs in time $s^{\exp \circ \exp(O(\log^* s))}$ for all degree-s size-s algebraic circuits over n variables. In other words, any blackbox PIT with just a slightly non-trivial exponent of s compared to the trivial $s^{O(n)}$ test can be used to give a nearly polynomial time blackbox PIT algorithm.

*mrinalkumar08@gmail.com. Center for Mathematical Sciences and Applications, Harvard University, Cambridge, USA. A part of this work was done while visiting TIFR, Mumbai.

†ramprasad@tifr.res.in. Tata Institute of Fundamental Research, Mumbai, India. Research supported by Ramanujan Fellowship of DST

‡tengse.anamay@tifr.res.in. Tata Institute of Fundamental Research, Mumbai, India. Supported by a fellowship of the DAE.

ISSN 1433-8092

1 Introduction

Multivariate polynomials are the primary protagonists in the field of algebraic complexity and algebraic circuits form a natural robust model of computation for multivariate polynomials. For completeness, an algebraic circuit is defined via a directed acyclic graph with internal gates labeled by $+$ (addition) and $\times$ (multiplication) and with leaves labeled by either variables or field constants; computation flows in the natural way.

In the field of algebraic complexity, much of the focus has been restricted to studying n-variate polynomials whose degree is bounded by a polynomial function in n, and such polynomials are called *low-degree polynomials*. This restriction has several a-priori and a-posteriori motivations, and excellent discussions of this can be seen in the thesis of Forbes [For14, Section 3.2] and Grochow's answer [Gro] on `cstheory.SE`. The central question in algebraic complexity is to find a family of low-degree polynomials that requires large algebraic circuits to compute it. Despite having made substantial progress in various subclasses of algebraic circuits (cf. surveys [SY10, Sap15]), the current best lower bound for general algebraic circuits is merely an $\Omega(n \log d)$ lower bound of Baur and Strassen [BS83].

An interesting approach towards proving lower bounds for algebraic circuits is via showing good *upper bounds* for the *algorithmic* task called *polynomial identity testing*. Our results deal with this approach and we elaborate on this now.

1.1 Polynomial Identity Testing

Polynomial identity testing (PIT[1]) is the algorithmic task of checking if a given algebraic circuit C of size s computes the identically zero polynomial. As discussed earlier, although a circuit of size s can compute a polynomial of degree 2^s, this question typically deals only with circuits whose *formal degree*[2] is bounded by the size of the circuit.

This algorithmic question has two flavours: whitebox PIT and blackbox PIT. Whitebox polynomial identity tests consist of algorithms that can inspect the circuit (that is, look at the underlying gate connections etc.) to decide whether the circuit computes the zero polynomial or not. A stronger algorithm is a *blackbox polynomial identity test* where the algorithm is only provided basic parameters of the circuit (such as its size, the number of variables, a bound on the formal degree) and only has evaluation access to the circuit C. Hence, a blackbox polynomial identity test for a class $\mathcal{C}$ of circuits is just a list of evaluation points $\mathcal{H} \subseteq \mathbb{F}^n$ such that every nonzero circuit $C \in \mathcal{C}$ is guaranteed to have some $\mathbf{a} \in \mathcal{H}$ such that $C(\mathbf{a}) \neq 0$. Such sets of points are also called *hitting sets* for $\mathcal{C}$. Therefore, the running time of a blackbox PIT algorithm is essentially given by the size of the *hitting set* and the time taken to generate it given the parameters of the circuit.

The classical Ore-DeMillo-Lipton-Schwartz-Zippel Lemma [Ore22, DL78, Zip79, Sch80] states

[1] We use the abbreviation PIT for both the noun 'polynomial identity test' and gerund/adjective 'polynomial identity testing'. The case would be clear from context.

[2] This is defined inductively by setting the formal degree of leaves as 1, and taking the sum at every multiplication gate and the max at every sum gate.

that any nonzero polynomial $f(x_1, \ldots, x_n)$ of degree at most d will evaluate to a nonzero value at a randomly chosen point from a grid $S^n \subseteq \mathbb{F}^n$ with probability at least $1 - \frac{d}{|S|}$. Therefore, this automatically yields a *randomized* polynomial time blackbox PIT algorithm, and also a deterministic $(d+1)^n \cdot \text{poly}(s)$ blackbox PIT algorithm, for the class of size s and formal-degree d circuits. Furthermore, a simple counting/dimension argument also says that there exist (non-explicit) $\text{poly}(s)$ sized hitting sets for the class of polynomials computed by size s algebraic circuits. The major open question is to find a better *deterministic* algorithm for this problem.

PIT is an important algorithmic question of its own right, and many classical results such as the primality testing algorithm [AKS04], $\mathsf{IP} = \mathsf{PSPACE}$ [LFKN90, Sha90], algorithms for graph matching [MVV87, FGT16, ST17] all have a polynomial identity test at its core. Yet another reason why PIT is an important algorithmic question is its intimate connections with the question of proving explicit lower bounds for algebraic circuits.

Heintz and Schnorr [HS80], and Agrawal [Agr05] observed that given an explicit hitting set for size s circuits, any nonzero polynomial that is designed to vanish on every point of the hitting set cannot be computable by size s circuits. By tailoring the number of variables and degree of the polynomial in this observation, they showed that polynomial time blackbox PITs yield an E-computable family $\{f_n\}$ of n-variate multilinear polynomials that require $2^{\Omega(n)}$ sized circuits. This connection between PIT and lower bounds was strengthened further by Kabanets and Impagliazzo [KI04] who showed that explicit families of hard functions can be used to give non-trivial derandomizations for PIT. Thus, the question of proving explicit lower bounds and the task of finding upper bounds for PIT are essentially two sides of the same coin.

1.2 Bootstrapping

A recent result of Agrawal, Ghosh and Saxena [AGS18] showed, among other things, the following surprising result: blackbox PIT algorithms for size s and n-variate circuits with running time as bad as $\left(s^{n^{0.5-\delta}} \cdot \text{HUGE}(n)\right)$, where $\delta > 0$ and HUGE is an arbitrary function of n, can be used to construct blackbox PIT algorithms for size s circuits with running time $s^{\exp \circ \exp(O(\log^* s))}$. Note that $\log^* n$ refers to the smallest i such that the i-th iterated logarithm $\log^{\circ i}(n)$ is at most 1. This shows that good-enough derandomizations of PIT would be sufficient to get a nearly complete derandomization. Their proof uses a novel *bootstrapping* technique where they use the connection between hardness and derandomization repeatedly so that by starting with a weak hitting set we can obtain better and better hitting sets.

One of the open questions of Agrawal, Ghosh and Saxena [AGS18] was whether the hypothesis can be strengthened to a *barely* non-trivial derandomization. That is, suppose we have a blackbox PIT algorithm, for the class of size s and n-variate circuits, that runs in time $s^{o(n)} \cdot \text{HUGE}(n)$, can we use this to get a nearly complete derandomization? Note that we have a trivial $s^{O(n)}$ algorithm from the Ore-DeMillo-Lipton-Schwartz-Zippel lemma [Ore22, DL78, Zip79, Sch80]. Our main result is an answer to this question in the affirmative.

Theorem 1.1 (Main theorem). *Let* $\text{HUGE} : \mathbb{N} \to \mathbb{N}$ *be an arbitrary function. Suppose there is an explicit hitting set of size* $s^{o(n)}\text{HUGE}(n)$ *for all degree-*d *size-*s *circuits over* n *variables. Then, there is an explicit hitting set of size* $s^{\exp \circ \exp(O(\log^* s))}$ *for the class of degree-*s *size-*s *circuits over* s *variables.*

1.3 Proof overview

The basic intuition for the proofs in this paper, and as per our understanding also for the proofs of the results in the work of Agrawal et al. [AGS18], comes from the results of Kabanets and Implagliazzo [KI04] and those of Heintz and Schnorr [HS80] and Agrawal [Agr05]. We start by informally stating these results.

Theorem 1.2 (Informal, Heintz and Schnorr [HS80], Agrawal [Agr05]). *Let* $H(n, d, s)$ *be an explicit hitting set for circuits of size* s*, degree* d *in* n *variables. Then, for every* d' *and* $k \leq n$ *such that* $d'k < d$ *and* $d'^k > |H(n, d, s)|$*, there is a nonzero polynomial on* n *variables and individual degree* d' *that vanishes on the hitting set* $H(n, d, s)$*, and hence cannot be computed by a circuit of size* s*.*

In a nutshell, given an explicit hitting set, we can obtain hard polynomials. In fact, playing around with the parameters d' and $k \leq n$, we can get a hard polynomial on k variables, degree kd' for all k, d' satisfying $d'k < d$ and $d'^k > |H(n, d, s)|$.

We now state a result of Kabanets and Impagliazzo [KI04] that shows that hardness can lead to derandomization.

Theorem 1.3 (Informal, Kabanets and Impagliazzo [KI04]). *A superpolynomial lower bound for algebraic circuits for an explicit family of polynomials implies a deterministic blackbox PIT algorithm for all algebraic circuits in* n *variables and degree* d *of size* $\text{poly}(n)$ *that runs in time* $\text{poly}(d)^{n^\varepsilon}$ *for every* $\varepsilon > 0$*.*

Now, we move on to the main ideas in our proof. Suppose we have hitting sets of size $s^{o(n)}$ for size s, degree $d \leq s$ circuits on n variables. The goal is to obtain a blackbox PIT for circuits of size s, degree s on s variables with a much better dependence on the number of variables.

Observe that if the number of variables was much much smaller than s, say at most a constant, then the hitting set in the hypothesis has a polynomial dependence on s, and we are done. With this in mind, the hitting sets for s variate circuits in the conclusion of Theorem 1.1 are designed iteratively starting from hitting sets for circuits with very few variables. In each iteration, we start with a hitting set for size s, degree $d \leq s$ circuits on n variables with some dependence on n and obtain a hitting set for size s, degree $d \leq s$ circuits on $m = 2^{n^\delta}$ variables (for some $\delta > 0$), that has a *much* better dependence on m. Then, we repeat this process till the number of variables increases up to s, which takes $O(\log^* s)$ iterations. We now briefly outline the steps in each such iteration.

- **Obtaining a family of hard polynomials :** The first step is to obtain a family of explicit hard polynomials from the given hitting sets. This step is done via Theorem 1.2, which simply uses interpolation to find a nonzero polynomial Q on k variables and degree d that vanishes on the hitting set for size s', degree d' circuits on n variables, for some s', d' to be chosen appropriately.

- **Variable reduction using Q :** Next, we take a Nisan-Wigderson design (see Definition 2.1) $\{S_1, S_2, \ldots, S_m\}$, where each S_i is a subset of size k of a universe of size k^2, and $|S_i \cap S_j| \leq \sqrt{k}$. Consider the map $\Gamma : \mathbb{F}[x_1, x_2, \ldots, x_m] \to \mathbb{F}[y_1, y_2, \ldots, y_{k^2}]$ given by $\Gamma(C(x_1, x_2, \ldots, x_m)) = C\left(Q(\mathbf{y}\,|_{S_1}), Q(\mathbf{y}\,|_{S_2}), \ldots, Q(\mathbf{y}\,|_{S_m})\right)$. As Kabanets and Impagliazzo show in the proof of Theorem 1.3, Γ preserves the nonzeroness of all algebraic circuits of size s on m variables, provided Q is hard enough, i.e. $s' = s^a$ for a sufficiently large a.

- **Blackbox PIT for m-variate circuits of size s and degree s :** We now take the hitting set given by the hypothesis for the circuit $\Gamma(C)$ (invoked with appropriate size and degree parameters) and evaluate $\Gamma(C)$ on this set. From the discussion so far, we know that if C is nonzero, then $\Gamma(C)$ cannot be identically zero, and hence it must evaluate to a nonzero value at some point on this set. The number of variables in $\Gamma(C)$ is at most $k^2 = \log^2 m$, whereas its size turns out to be *not too much* larger than s. Hence, the size of the hitting set for C obtained via this argument turns out to have a better dependence on the number of variables m than the hitting set in the hypothesis.

Similarities and differences with the proof of Agrawal et al. [AGS18]. The high level outline of our proof is essentially the same as that of Agrawal et al. [AGS18]. However, there are some quantitative differences in the argument, that make our final arguments shorter and simpler than those of Agrawal et al. and lead to a stronger and near optimal boostrapping statement in Theorem 1.1.

The primary differences between our proof and that of Agrawal et al. are rather technical but we try to briefly describe them. The first difference is in the choice of Nisan-Wigderson designs. The designs used in this paper are based on the standard Reed-Solomon code and they yield larger set families than the designs used by Agrawal et. al.[3] The second difference is the evolution of parameters in the inductive argument. We believe the primary difference is in the main inductive hypothesis [AGS18, Lemma 18] that assumes a non-trivial hitting set for some n-variate circuits for n smaller than some large constant n_0 (which they then use to bootstrap). However, this hypothesis does not degrade gracefully for n-variate circuits when n is a little larger than n_0 and appears to force them to use more stringent parameters. Also, their proof is quite involved and we are unsure if there are other constraints in their proof that force such choices of parameters. Our proof, though along almost exactly the same lines, appears to be more transparent and more malleable with respect to the choice of parameters.

The strength of the hypothesis. The hypothesis of Theorem 1.1 and also those of the results in the work of Agrawal et al. [AGS18] is that we have a non-trivial explicit hitting set for algebraic circuits of size s, degree d on n variables where d and s could be arbitrarily large as a function of n. This seems like an extremely strong assumption, and also slightly non-standard in the following sense. In a typical setting in algebraic complexity, we are interested in PIT for size s, degree d

[3] However, even without these improved design parameters, our proof can be used to provide the same conclusion when starting off with a hitting set of size $s^{n^{1-\delta}} \cdot \mathrm{HUGE}(n)$, instead of the hypothesis of Theorem 1.1.

circuits on n variables where d and s are polynomially bounded in the number of variables n. A natural open problem here, which would be a more satisfying statement to have, would be to show that one can weaken the hypothesis in Theorem 1.1 to only hold for circuits whose degree and size are both polynomially bounded in n. It is not clear to us if such a result can be obtained using the current proof techniques, or is even true.

Remark. *Throughout the paper, we shall assume that there are suitable $\lfloor\cdot\rfloor$'s or $\lceil\cdot\rceil$'s if necessary so that certain parameters chosen are integers. We avoid writing this purely for the sake of readability.*

Furthermore, we make absolutely no attempt *to optimise constants. Several of the inequalities used are weak and tightening them makes little qualitative difference to the final theorem statements.* ◊

2 Preliminaries

2.1 Notation

- For a positive integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.
- We use boldface letters such as $\mathbf{x}_{[n]}$ to denote a set $\{x_1, \ldots, x_n\}$. We drop the subscript whenever the number of elements is clear or irrelevant in the context.
- We use $\mathcal{C}(n, d, s)$ to denote the class of n-variate polynomials of formal degree at most d that are computable by algebraic circuits of size at most s. This class may also include polynomials that actually depend on fewer variables but are masquerading to be n-variate polynomials.
- For a polynomial $f(x_1, \ldots, x_n)$, we shall say its *individual degree* is at most k to mean that the exponent of any of the x_i's in any monomial is at most k.

2.2 Some basic definitions and lemmas

Definition 2.1 (Nisan-Wigderson designs [NW94])**.** *A family of sets $S_1, \ldots, S_m \subseteq [\ell]$ is said to be an (ℓ, k, r)-design if*

- $|S_i| = k$,
- $|S_i \cap S_j| < r$ *for any* $i \neq j$. ◊

The following is a standard construction of such designs based on the *Reed-Solomon* code.

Lemma 2.2 (Construction of NW designs)**.** *There is an algorithm that, given parameters ℓ, k, r satisfying $\ell = k^2$ and $r \leq k$ with k being a power of 2, outputs an (ℓ, k, r)-design $\{S_1, \ldots, S_m\}$ for $m \leq k^r$ in time* $\mathrm{poly}(m)$.

Proof. Since k is a power of 2, we can identify $[k]$ with the field $\mathbb{F}_k$ of k-elements and $[\ell]$ with $\mathbb{F}_k \times \mathbb{F}_k$. For each univariate polynomial $p(x) \in \mathbb{F}_k[x]$ of degree less than r, define the set S_p as

$$S_p = \{(i, p(i)) \ : \ i \in \mathbb{F}_k\}.$$

Since there are k^r such polynomials we get k^r subsets of $\mathbb{F}_k \times \mathbb{F}_k$ of size k each. Furthermore, since any two distinct univariate polynomials cannot agree at r or more places, it follows that $\left|S_p \cap S_q\right| < r$ for $p \neq q$. □

Hardness-randomness connection

For a fixed (ℓ, k, r)-design $S_1, \ldots, S_m$ and a polynomial $Q(z_1, \ldots, z_k) \in \mathbb{F}[\mathbf{x}]$ we shall use the notation $Q[\![\ell, k, r]\!]_{\mathrm{NW}}$ to denote the vector of polynomials

$$Q[\![\ell, k, r]\!]_{\mathrm{NW}} := \left(Q(\mathbf{y}\,|_{S_1}), Q(\mathbf{y}\,|_{S_2}), \ldots, Q(\mathbf{y}\,|_{S_m})\right) \in \left(\mathbb{F}[y_1, \ldots, y_\ell]\right)^m.$$

Lemma 2.3 (Generators from hard polynomials [KI04])**.** *Let $S_1, \ldots, S_m$ be an (ℓ, k, r)-design and let $Q(z_1, \ldots, z_k)$ be a k-variate polynomial of individual degree bounded by d. Suppose there is a size s circuit computing a nonzero polynomial $P(x_1, \ldots, x_m)$ of degree at most D such that $P(Q[\![\ell, k, r]\!]_{\mathrm{NW}}) \equiv 0$. Then, there is a circuit of size at most $(s + m(d+1)^r)^e$ (for a universal constant e) computing $Q(z_1, \ldots, z_k)$.* □

Remark. *The exponent e in the above theorem, in the setting when $D \leq s$, is at most 5 using the bound on the complexity of factors due to Bürgisser [Bür00, Theorem 2.21].* ♢

Conversely, if $Q(\mathbf{z}_{[k]})$ was *hard enough*, then $P(Q[\![\ell, k, r]\!]_{\mathrm{NW}})$ is nonzero ℓ-variate polynomial whenever $P(\mathbf{x}_{[m]})$ is nonzero.

Lemma 2.4 (Hitting sets to hardness [HS80, Agr05])**.** *Let H be an explicit hitting set for $\mathcal{C}(n, d, s)$, the class of n-variate of formal degree at most d polynomials computable by size s algebraic circuits. Then, for any $k \leq n$ such that $k|H|^{1/k} \leq d$, there is a polynomial $Q(z_1, \ldots, z_k)$ of degree at most $k|H|^{1/k}$ that is computable in time* $\mathrm{poly}(|H|)$ *such that it cannot be computed by algebraic circuits of size s. Moreover $Q(\mathbf{x})$ has an algebraic circuit of size $O(|H|)$.*

Proof. This is achieved by finding a nonzero k-variate polynomial, for $k \leq n$, of individual degree smaller than $|H|^{1/k}$ that vanishes on the hitting set H for $\mathcal{C}(n, d, s)$. The degree of Q_k is at most $k \cdot |H|^{1/k} \leq d$ from the hypothesis. Such a Q_k can be found by solving a system of linear equations in time $\mathrm{poly}(|H|)$. By the definition of the hitting set, we must have that $Q_k(z_1, \ldots, z_k)$ cannot be an element of $\mathcal{C}(n, d, s)$ and therefore Q_k cannot be computed by algebraic circuits of size s. However, note that Q_k is a sum of at most $|H|$ monomials over k variables and thus has an algebraic circuit of size at most $k + 1 + |H|$. □

3 Bootstrapping Hitting Sets

In this section, we give a simple proof of the main result of Agrawal et al. [AGS18] along the same lines as the original proof albeit with different parameters. This proof, besides being a more transparent exposition, would also ensure that any constraints while setting various parameters are made clear.

Theorem 3.1. (A weaker version of [AGS18, Theorem 3]) *Let n_0 be a large enough[4] constant that is a power of 2, and let s be a growing parameter. Suppose $g : \mathbb{N} \to \mathbb{N}$ is a non-decreasing function with $30 \cdot g(n_0) < n_0^{1/4}$ such that, for all large enough values of s, there is an explicit hitting set of size $s^{g(n_0)}$ for $\mathcal{C}(n_0, s, s)$.*

Then there is an explicit hitting set for $\mathcal{C}(s, s, s)$ of size $s^{\exp \circ \exp(O(\log^* s))}$.

The following lemma describes the main inductive statement using which Theorem 3.1 follows readily.

Lemma 3.2. *Let n_0 be a large enough[4] power of 2, and let s be a growing parameter. Suppose $g : \mathbb{N} \to \mathbb{N}$ is a non-decreasing function with $30 \cdot g(n_0) < n_0^{1/4}$ such that, for all large enough s, there is an explicit hitting set of size $s^{g(n_0)}$ for degree-s size-s circuits over n_0 variables.*

Then for $n_1 = 2^{n_0^{1/4}} > n_0$ and $h : \mathbb{N} \to \mathbb{N}$ given by $h(n) = 30 \cdot (g((\log n)^4))^2$, there is an explicit hitting set of size $s^{h(n_1)}$ for degree-s size-s circuits on n_1 variables. Furthermore, $h(n_1)$ also satisfies $30 \cdot h(n_1) < n_1^{1/4}$.

We will defer the proof of this lemma and finish the proof of Theorem 3.1.

Proof of Theorem 3.1. The hypothesis and conclusion of Lemma 3.2 admit repeated applications of the lemma to get hitting sets for polynomials depending on larger sets of variables. The natural strategy is therefore to apply the lemma repeatedly to obtain a hitting set for the class $\mathcal{C}(s, s, s)$. We now set up some basic notation to facilitate this analysis.

We start with an explicit hitting set of size $s^{g(n_0)}$ for $\mathcal{C}(n_0, s, s)$ circuits and say after i applications of Lemma 3.2 we have an explicit hitting set for the class $\mathcal{C}(n_i, s, s)$ of size s^{t_i}. We wish to track the evolution of n_i and t_i. Recall that $n_i = 2^{n_{i-1}^{1/4}}$ after one iteration of Lemma 3.2. Let $\{m_i\}_i$ be such that $m_0 = \log n_0$ and, for every $i > 0$, let $m_i = 2^{(m_{i-1}/4)}$ so that $m_i = \log n_i$. Similarly to keep track of the complexity of the hitting set, if s^{t_i} is the size of the hitting set for $\mathcal{C}(n_i, s, s)$, then by Lemma 3.2 we have $t_0 = g(n_0)$ and $t_i = 30 \cdot t_{i-1}^2$ for all $i \geq 1$.
The following facts are easy to verify.

- $m_i \geq \log s$ for $i = O(\log^* s)$,
- for all j, we have $t_j = 30^{(2^j - 1)} \cdot t_0^{2^j} = \exp \circ \exp(O(j))$.
- the exponent of s in the complexity of the final hitting set is $t_{O(\log^* s)} = \exp \circ \exp(O(\log^* s))$.

Therefore we have an $s^{\exp \circ \exp(O(\log^* s))}$ sized explicit hitting set for $\mathcal{C}(s, s, s)$. □

Proof of Lemma 3.2. We would need to fix some parameters: $k = n_0^{1/2}$, $\ell = n_0$ and $r = n_0^{1/4}$.

Constructing a hard polynomial: The first step is to construct a polynomial $Q_k(z_1, \ldots, z_k)$ that cannot be computed by n_0-variate size s^{15} circuits. This can be done by using Lemma 2.4. The polynomial $Q_k(\mathbf{z})$ will therefore have the following properties.

[4] This is to ensure that $2^{n^{1/4}} > n$ for all $n > n_0$ and this is true for any $n_0 > 2^{16}$.

- Individual degree $\leq s^{15g(n_0)/k} \leq s$, and degree $\leq s^{15g(n_0)/k} \cdot s \leq s^2$.
- Q_k is not computable by circuits of size s^{15}.
- Q_k has an algebraic circuit of size $\leq 2s^{15g(n_0)}$.

Building the NW design: Using Lemma 2.2 [5], construct an (ℓ, k, r)-design $\{S_1, \ldots, S_{n_1}\}$ for $2^r = n_1$, which is bigger than n_0 since n_0 is large enough[4].

Variable reduction using Q_k: Let $0 \not\equiv P(x_1, \ldots, x_m) \in \mathcal{C}(m, s, s)$. Suppose $P(Q[\![\ell, k, r]\!]_{\mathrm{NW}}) \equiv 0$, then Lemma 2.3 forces Q_k to have an algebraic circuit of size bounded by

$$\left(s + n_1 \cdot 2s^{15g(n_0)r/k}\right)^e \leq (s + s^2)^5 \leq s^{15},$$

which we know is false by our construction of Q_k. Therefore, $P(Q[\![\ell, k, r]\!]_{\mathrm{NW}})$ is a nonzero polynomial.

Since P has a circuit of size s and Q_k has a circuit of size $s^{15g(n_0)}$, it follows that the polynomial $P(Q_k[\![\ell, k, r]\!]_{\mathrm{NW}})$ has a circuit of size at most $s + k \cdot 2s^{15g(n_0)} \leq s^{30g(n_0)} =: s'$. Furthermore, the degree of the polynomial $P(Q_k[\![\ell, k, r]\!]_{\mathrm{NW}})$ is at most $s \cdot k \cdot s^{15g(n_0)/k} \leq s^{30g(n_0)}$.

Hitting set for $\mathcal{C}(n_1, s, s)$: From the above discussion, the polynomial $P(\mathbf{x}) \equiv 0$ if and only if $P' = P(Q_k[\![\ell, k, r]\!]_{\mathrm{NW}}) \equiv 0$. We also know that $P' \in \mathcal{C}(\ell, s', s')$ where $s' = s^{30g(n_0)}$. Therefore, by composing the hitting set for $\mathcal{C}(\ell, s', s')$ with $Q_k[\![\ell, k, r]\!]_{\mathrm{NW}}$, we obtain a hitting set for $\mathcal{C}(n_1, s, s)$. The size of the hitting set is

$$(s')^{g(n_0)} = s^{30(g(n_0))^2} \leq s^{30g((\log n_1)^4)^2} = s^{h(n_1)}.$$

Re-establishing the invariant:

$$30h(n_1) = (30g((\log n_1)^4))^2 = (30g(n_0))^2 < n_0^{1/2} = (\log n_1)^2 < 30n_1^{1/4},$$

where the last inequality uses the fact that $(\log n)^2 < 30n^{1/4}$ for all $n \geq 1$.

It is clear that the entire construction is in polynomial time in the size of the hitting set of the conclusion and the running time of the hitting set construction in the hypothesis. □

3.1 Near-optimal bootstrapping

To finish the proof of the main theorem (Theorem 1.1), we show how we can go from the hypothesis of Theorem 1.1 to the hypothesis of Theorem 3.1. This is again along the same lines as the proof of Lemma 3.2 but with a different choice of parameters.

[5] The lemma can provide more sets but this weaker version is chosen to just make some calculations easier.

Lemma 3.3. *Let $f : \mathbb{N} \to \mathbb{N}$ be a growing function and let $\text{HUGE} : \mathbb{N} \to \mathbb{N}$ be an arbitrary function. Suppose there is an explicit hitting set of size $\left(s^{n/f(n)}\right)\text{HUGE}(n)$ for all degree-s size-s circuits over n variables.*

Then there exists a constant m_0 that is a power of 2 and large enough[4], and a non-decreasing function $h : \mathbb{N} \to \mathbb{N}$ satisfying $30 \cdot h(m_0) < m_0^{1/4}$ such that for all large enough values of s, there is an explicit hitting set of size $s^{h(m_0)}$ for degree-s size-s circuits over m_0 variables.

Again, we will defer the proof of this lemma as Theorem 1.1 follows readily from this lemma.

Proof of Theorem 1.1. As a consequence of Lemma 3.3, we have that for large enough m_0 and $h : \mathbb{N} \to \mathbb{N}$ satisfying $30h(m_0) < m_0^{1/4}$ (given by Lemma 3.3), there is an explicit hitting set of size $s^{h(m_0)}$ for $\mathcal{C}(m_0, s, s)$ for all large enough values of s.

Since this conclusion satisfies the hypothesis of Theorem 3.1 we can infer that there is an explicit hitting set for $\mathcal{C}(s, s, s)$ of size $s^{\exp \circ \exp(O(\log^* s))}$. □

Proof of Lemma 3.3. The strategy is exactly along the lines of Lemma 3.2 but we would have to work with slightly different parameters. Let n be the smallest integer satisfying the following constraints:

- n is at least 7 and is a power of 2,
- $\sqrt{f(n)} > 30$.

Fix $n_0 := n^2$. Note that for every $s \geq \max_{m \leq n_0}(\text{HUGE}(m))$ and every $m \leq n_0$, we have an explicit hitting set of size at most $s^{2m/f(m)}$ for the class $\mathcal{C}(m, s, s)$.

Fix the parameters $\ell := n_0$, $k := \sqrt{n_0} = n$ and $r := \sqrt{f(n)}$.

Constructing a suitably hard polynomial: We will construct a polynomial $Q_n(z_1, \ldots, z_n)$ that is not computable by algebraic circuits of size s^{15}. We will again invoke Lemma 2.4 to do this. The polynomial $Q_n(\mathbf{z})$ has the following properties.

- Individual degree $\leq s^{30/f(n)} \leq s$, and degree $\leq s^{30/f(n)} \cdot s \leq s^2$.
- Q_n is not computable by circuits of size s^{15}.
- Q_n has an algebraic circuit of size $\leq 2s^{30n/f(n)}$.

Building the NW design: We will use Lemma 2.2 to construct an (ℓ, n, r) design $\{S_1, \ldots, S_{m_0}\}$ with $m_0 := n^r = n^{\sqrt{f(n)}}$.

Variable reduction: Let $P(x_1, \ldots, x_{m_0})$ be a nonzero circuit from $\mathcal{C}(m_0, s, s)$. Like in Lemma 3.2, since

$$\left(s + m_0 \cdot 2s^{30\sqrt{f(n)}/f(n)}\right)^5 \leq (s + s^2)^5 \leq s^{15},$$

and Q_n is hard for circuits of size s^{15} by construction, we have that $P(Q[\![\ell, n, r]\!]_{\mathrm{NW}})$ must be nonzero.

Note that P has a circuit of size s and Q_n is trivially computable by a circuit of size $2s^{30n/f(n)}$. Therefore $P(Q[\![\ell, n, r]\!]_{\mathrm{NW}})$ has a circuit of size $s + m_0 \cdot 2s^{30n/f(n)} \leq s^{60n/f(n)} = s'$ (say). Also the degree of the polynomial computed by $P(Q[\![\ell, n, r]\!]_{\mathrm{NW}})$ is at most $s \cdot ns^{30/f(n)} \leq s^3 \leq s^{60n/f(n)}$.

Hitting set for $\mathcal{C}(m_0, s, s)$: Starting with a nonzero circuit from $\mathcal{C}(m_0, s, s)$, we have now obtained a nonzero circuit in the class $\mathcal{C}(n_0, s', s')$. We now apply the hypothesis to the circuit on n_0 variables thereby obtaining an explicit hitting set for $\mathcal{C}(m_0, s, s)$ of size

$$\left(s'\right)^{2n_0/f(n_0)} \leq \left(s^{60n/f(n)}\right)^{2n_0/f(n_0)} = s^{120n \cdot n_0/(f(n) \cdot f(n_0))} \leq s^{120n^3}.$$

Bounding $h(m_0)$: Define $h(m_0) := 120n^3$, for our choice of $m_0 = n^{\sqrt{f(n)}}$. Therefore $30h(m_0) = 30 \cdot 120n^3 = 3600n^3$ and $m_0^{1/4} = n^{\frac{1}{4}\sqrt{f(n)}}$. Since n was chosen so that $n > 7$ and $\sqrt{f(n)} > 30$, we have $m_0^{1/4} > n^{30/4}$ and $m_0 > 2^{16}$. For such a choice of n we also have $3600n^3 < n^{15/2}$ and hence m_0 is large enough and $30 \cdot h(m_0) < m_0^{1/4}$ as required.

It is easy to verify that the entire construction runs in time that is polynomial in (1) the time required for the construction of the hitting set from the hypothesis and (2) the size of the hitting set in the conclusion $(s^{h(m)})$. □

4 Conclusions

The main results show that it suffices to construct hitting sets of size $s^{o(n)}$, that are barely better than the trivial hitting set of s^n, to obtain an almost complete derandomisation. A natural question in the spirit of the results in this paper, and those in Agrawal et al. [AGS18] seems to be the following : Can we hope to bootstrap lower bounds? In particular, can we hope to start from a mildly non-trivial lower bound for general arithmetic circuits (e.g. superlinear or just superpolynomial), and hope to amplify it to get a stronger lower bound (superpolynomial or truly exponential respectively). In the context of non-commutative algebraic circuits, Carmosino et al. [CILM18] recently showed such results, but no such result appears to be known for commutative algebraic circuits.

Acknowledgements: Ramprasad and Anamay would like to thank the organisers of the Workshop on Algebraic Complexity Theory (WACT 2018) where we first started addressing this problem.

References

[Agr05] Manindra Agrawal. Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2005.

[AGS18] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 1166–1179. ACM, 2018. eccc:TR18-035.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[BS83] Walter Baur and Volker Strassen. The Complexity of Partial Derivatives. *Theoretical Computer Science*, 22:317–330, 1983.

[Bür00] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer, 2000.

[CILM18] Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin. Hardness Amplification for Non-Commutative Arithmetic Circuits. In *Proceedings of the 33rd Annual Computational Complexity Conference (CCC 2018)*, volume 102 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. eccc:TR18-095.

[DL78] Richard A. DeMillo and Richard J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978.

[FGT16] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 754–763. ACM, 2016. eccc:TR15-177.

[For14] Michael Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

[Gro] Joshua Grochow. http://cstheory.stackexchange.com/questions/19261/degree-restriction-for-polynomials-in-mathsfvp/19268#19268.

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing Polynomials which Are Easy to Compute (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, pages 262–272. ACM, 1980.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic Methods for Interactive Proof Systems. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*, pages 2–10, 1990.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. Preliminary version in the *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. Available on `citeseer:10.1.1.83.8416`.

[Ore22] Øystein Ore. Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

[Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015.

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Sha90] Adi Shamir. IP=PSPACE. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*, pages 11–15, 1990.

[ST17] Ola Svensson and Jakub Tarnawski. The Matching Problem in General Graphs Is in Quasi-NC. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2017)*, pages 696–707. IEEE Computer Society, 2017. `arXiv:1704.01929`.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il