SIAM J. COMPUT.
Vol. 45, No. 3, pp. 1064–1079

ARITHMETIC CIRCUITS: A CHASM AT DEPTH 3[*]

ANKIT GUPTA[†], PRITISH KAMATH[†], NEERAJ KAYAL[†], AND RAMPRASAD SAPTHARISHI[‡]

Abstract. We show that, over $\mathbb{Q}$, if an n-variate polynomial of degree $d = n^{O(1)}$ is computable by an arithmetic circuit of size s (respectively, by an arithmetic branching program of size s), then it can also be computed by a depth-3 circuit (i.e., a $\Sigma\Pi\Sigma$ circuit) of size $\exp(O(\sqrt{d \log n \log d \log s}))$ (respectively, of size $\exp(O(\sqrt{d \log n \log s}))$. In particular this yields a $\Sigma\Pi\Sigma$ circuit of size $\exp(O(\sqrt{d} \cdot \log d))$ computing the $d \times d$ determinant Det_d. It also means that if we can prove a lower bound of $\exp(\omega(\sqrt{d} \cdot \log d))$ on the size of any $\Sigma\Pi\Sigma$ circuit computing the $d \times d$ permanent Perm_d, then we get superpolynomial lower bounds for the size of any arithmetic branching program computing Perm_d. We then give some further results pertaining to derandomizing polynomial identity testing and circuit lower bounds. The $\Sigma\Pi\Sigma$ circuits that we construct have the property that (some of) the intermediate polynomials have degree much higher than d. Indeed such a counterintuitive construction is unavoidable—it is known that in any $\Sigma\Pi\Sigma$ circuit C computing either Det_d or Perm_d, if every multiplication gate has fanin at most d (or any constant multiple thereof), then C must have size at least $\exp(\Omega(d))$.

Key words. arithmetic circuits, determinant, permanent, depth-3 circuits, depth reduction

AMS subject classifications. 68W30, 12E05

DOI. 10.1137/140957123

1. Introduction.

Arithmetic circuits. The most natural way to compute a polynomial function $f(x_1, x_2, \ldots, x_n)$ starting from its inputs $x_1, x_2, \ldots, x_n$ is via a sequence of basic arithmetic operations consisting of addition, multiplication, and subtraction. Such a computation can be visualized as an arithmetic circuit. We typically allow the incoming edges to a + gate to be labeled with constants from the underlying field $\mathbb{F}$ so that a + gate can in fact compute an arbitrary $\mathbb{F}$-linear combination of its inputs. Two relevant complexity measures for an arithmetic circuit are its size (the total number of arithmetic operations involved) and its depth (the maximum length of a path from an input to the output). The goal here is to understand the optimal complexity (in terms of size and depth) of computing a given polynomial family. Two closely related families of polynomials, the determinant and the permanent, defined as

$$\begin{aligned} \mathsf{Det}_d &= \sum_{\sigma \in S_d} \operatorname{sign}(\sigma) \cdot \prod_{i=1}^{d} x_{i,\sigma(i)}, \\ \mathsf{Perm}_d &= \sum_{\sigma \in S_d} \prod_{i=1}^{d} x_{i,\sigma(i)}, \end{aligned}$$

are of particular interest as they feature in many different areas of mathematics and computer science. Although these two polynomials look very similar, they have strik-

[*] Received by the editors February 14, 2014; accepted for publication (in revised form) May 8, 2015; published electronically June 29, 2016.
http://www.siam.org/journals/sicomp/45-3/95712.html

[†] Microsoft Research India, Bangalore, Karnataka 560001, India (ankitgupta.iitkanpur@gmail.com, pritish.kamath@gmail.com, neeraka@microsoft.com).

[‡] Chennai Mathematical Institute, Kelambakkam 603103, India (ramprasad@cmi.ac.in).

ingly different complexities. The determinant and permanent are in fact complete[1] problems for the classes VP and VNP, respectively, which are algebraic analogues of P and NP [34]. A grand challenge in this direction is to show that Perm_d cannot be computed by arithmetic circuits of polynomial size.

Depth reduction. Circuits with low depth correspond to computations which are highly parallelizable and therefore it is natural to try to minimize the depth of a circuit while allowing the size to increase somewhat. Csanky [6] showed that Det_d can be computed by circuits of size $d^{O(1)}$ *having only* $(\log d)^{O(1)}$ *depth*. Subsequently Valiant et al. [35] discovered a remarkable generalization. They showed that if a polynomial f of degree d can be computed by a circuit of size s, then it can in fact be computed by a circuit of depth $O(\log d \cdot \log s)$ and size $(sd)^{O(1)}$. Pushing this line of investigation of size-depth trade-offs further, recent work has considered reduction to circuits of even smaller depth while allowing the addition and multiplication gates to have arbitrary (unbounded) fanin. In this direction, the work of Agrawal and Vinay [2] and a subsequent strengthening by Koiran [21] and Tavenas [33] showed that if f has circuits of size $s = d^{O(1)}$, then f can in fact be computed by depth-4 circuits of size[2] $2^{O(\sqrt{d}\cdot\log d)}$. Despite the large blow-up, the reason that such reductions to constant depth circuits are interesting is that these reductions "explain" the lack of progress toward lower bounds even for constant-depth arithmetic circuits.[3] Viewed optimistically, such a reduction entails that one *merely* needs to prove a (good enough) lower bound for depth-4 circuits in order to prove lower bounds for arbitrary circuits. Indeed, motivated by this, we recently proved lower bounds for depth-4 circuits [14], which come very close to the threshold required.[4] In a similar spirit Raz [25] showed that close-to-optimal lower bounds for small degree tensors[5] imply superpolynomial lower bounds on formula size.[6]

Depth-3 circuits. Being the shallowest nontrivial subclass of arithmetic circuits, depth-3 arithmetic circuits, also denoted as $\Sigma\Pi\Sigma$ circuits, have been intensely investigated. Such a circuit C computes a polynomial in the following manner:

$$C(\mathbf{x}) = \sum_{i=1}^{s}\prod_{j=1}^{d_i}\ell_{ij}(\mathbf{x}), \tag{1.1}$$

where each $\ell_{ij}(\mathbf{x})$ is an affine form over the input variables. $\Sigma\Pi\Sigma$ circuits (more specifically tensors) arise naturally in the investigation of the complexity ofpolynomial

[1] To be more accurate, Det_d is complete for the algebraic analogue of P under quasi-polynomial sized projections.

[2] A simple but nonconstructive counting argument shows that over any field $\mathbb{F}$, most n-variate polynomials of degree d require general circuits of size $\Omega(\sqrt{\binom{n+d}{d}}) = 2^{\Omega(d\log(n/d))}$ for $d \ll n$.

[3] Note that in contrast to arithmetic circuits, exponential lower bounds are known for constant depth boolean circuits with $\wedge, \vee$ gates of unbounded fanin.

[4] Specifically, [14] shows a lower bound of $\exp(\Omega(\sqrt{d}))$ on the size of $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ computing Perm_d. In comparison, Koiran showed that a lower bound of $\exp(\omega(\sqrt{d}\log^2 d))$ on the size of $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ computing Perm_d entails superpolynomial lower bounds for general circuits computing Perm_d.

[5] A subclass of $\Sigma\Pi\Sigma$ also known as set-multilinear $\Sigma\Pi\Sigma$ circuits.

[6] Specifically, Raz showed that for $d = d(n) \le \frac{\log n}{\log\log n}$, any explicit example of a tensor $A : [n]^d \mapsto \mathbb{C}$ with tensor rank $\ge n^{d(1-o(1))}$ implies an explicit superpolynomial lower bound for the size of general arithmetic formulas. Because of the restriction on the degree d, Raz's result does not seem to be applicable to the permanent.

multiplication and matrix multiplication.[7] Moreover, the optimal formula/circuit for some well-known families of polynomials are in fact depth-3 circuits. In particular, the best known circuit for computing the permanent Perm_d is known as Ryser's formula [26], which is a (homogeneous[8]) depth-3 circuit of size $O(d^2 \cdot 2^d)$. For more on $\Sigma\Pi\Sigma$ circuits, we refer the reader to the thesis of Shpilka [29] and the references therein.

Circuit lower bounds. While there has been significant progress in upper bounds, progress in proving lower bounds for arithmetic circuits has been much slower. This is generally considered to be one of the most challenging problems in computer science. The difficulty of the problem has led researchers to focus on natural subclasses of arithmetic circuits. We refer the interested reader to the recent surveys by Shpilka and Yehudayoff [32] and Chen, Kayal, and Wigderson [5] for more on lower bounds for various subclasses of arithmetic circuits. Bounded depth circuits being one such natural subclass has received a lot of attention. The simplest nontrivial such subclass is that of $\Sigma\Pi\Sigma$ circuits. Nisan and Wigderson [24] showed that over any field $\mathbb{F}$, any homogeneous $\Sigma\Pi\Sigma$ circuit computing the determinant Det_d must be of size $2^{\Omega(d)}$. Grigoriev and Karpinski [12] and Grigoriev and Razborov [13] showed that any $\Sigma\Pi\Sigma$ arithmetic circuit over any *fixed* finite field computing Det_d must be of size at least $2^{\Omega(d)}$. This also implies that any $\Sigma\Pi\Sigma$ arithmetic circuit *over integers* computing Det_d must be of size at least $2^{\Omega(d)}$. Raz and Yehudayoff give $2^{\Omega(d)}$ lower bounds for *multilinear* $\Sigma\Pi\Sigma$ circuits.[9] But without any restrictions, even a superpolynomial lower bound for $\Sigma\Pi\Sigma$ circuits (over an infinite field) has remained elusive. The best known lower bound in the general $\Sigma\Pi\Sigma$ case is the quadratic lower bound due to Shpilka and Wigderson [31]. Wigderson [36] highlighted this frontier in arithmetic complexity by concluding his plenary talk on P, NP, and mathematics at ICM 2006 with the problem of proving superpolynomial lower bounds for $\Sigma\Pi\Sigma$ circuits computing the determinant.

Our contribution. The $2^{\Omega(d)}$ lower bounds for various restrictions of $\Sigma\Pi\Sigma$ circuits mentioned above seemed to suggest (at least to us) that any $\Sigma\Pi\Sigma$ circuit computing the determinant Det_d needs to be of size at least $2^{\Omega(d)}$. Surprisingly, we show that this is not true—there do indeed exist much smaller $\Sigma\Pi\Sigma$ circuits computing Det_d. Specifically we show that over $\mathbb{Q}$, the field of rational numbers, there exists a $\Sigma\Pi\Sigma$ circuit of size $2^{O(\sqrt{d}\cdot\log d)}$ computing the determinant. To the best of our knowledge, no $\Sigma\Pi\Sigma$ circuit of size smaller than even $2^{O(d\cdot\log d)}$ was previously known. More generally, we show the following.

THEOREM 1.1. *Let $f(\mathbf{x}) \in \mathbb{Q}[\mathbf{x}]$ be an n-variate polynomial of degree $d = n^{O(1)}$ computed by an arithmetic circuit of size s. Then it can also be computed by a $\Sigma\Pi\Sigma$ circuit of size $2^{O(\sqrt{d \log n \log d \log s})}$.*

Further, if f is computable by an arithmetic branching program of size s, then it can also be computed by a $\Sigma\Pi\Sigma$ circuit of size $2^{O(\sqrt{d \log n \log s})}$.

[7] For example, it can be shown that the product of two $n \times n$ matrices can be computed with $\tilde{O}(n^{\omega})$ arithmetic operations if and only if the polynomial $M_n = \sum_{i\in[n]}\sum_{j\in[n]}\sum_{k\in[n]} x_{ij}\cdot y_{jk}\cdot z_{ki}$ can be computed by a $\Sigma\Pi\Sigma$ circuit where the top fanin s is at most $\tilde{O}(n^{\omega})$.

[8] Recall that a multivariate polynomial is said to be homogeneous if all its monomials have the same total degree. An arithmetic circuit is said to be homogeneous if the polynomial computed at every internal node of the circuit is a homogeneous polynomial. It is a folklore result (cf. the survey by Shpilka and Yehudayoff [32]) that as far as computation by polynomial-sized arithmetic circuits of unbounded depth is concerned one can assume without loss of generality that the circuit is homogeneous. Specifically, if a homogeneous polynomial f of degree d can be computed by an (unbounded depth) arithmetic circuit of size s, then it can also be computed by a *homogeneous* circuit of size $O(d^2 \cdot s)$.

[9] The results of Raz and Yehudayoff are more general.

Note that Det_d can be computed by an ABP of size $d^{O(1)}$ (cf. [4, 22]) and hence we immediately get the $\Sigma\Pi\Sigma$ circuit for it as mentioned above. We note here that, in particular, the above theorem shows that a $\Sigma\Pi\Sigma$ lower bound of $2^{\Omega(d\log n)}$ for any explicit n-variate polynomial f of degree d entails a $2^{\Omega(d\log n)}$ lower bound on the size of any arithmetic circuit computing f.

Subsequent improvement by Tavenas [33]. The proof of Theorem 1.1 is modular and consists of three steps. The first step is a reduction to depth-4 circuits via the depth reduction of Agrawal and Vinay [2], and the subsequent strengthening of Koiran [21]. At a high level, Koiran's depth reduction applies for arithmetic branching programs, which also yields a depth reduction for general arithmetic circuits with slightly worse parameters (as any circuit of size s computing a degree d polynomial can be computed by a branching program of size $(sd)^{O(\log d)}$).

Recently, Tavenas [33] has improved this step to depth reduce general arithmetic circuits to depth-4 circuits without incurring the quasi-polynomial loss in Koiran's method. This improved depth reduction can be used in our proof to show that any n-variate degree d polynomial $f \in \mathbb{Q}[\mathbf{x}]$ computable by a size s circuit can also be computed by a $\Sigma\Pi\Sigma$ circuit of size $2^{O(\sqrt{d\log n\log s})}$.

Comparison with prior work. Prior work by [2], [21] reduced the depth to four and we build upon their work to reduced the depth even further to three. Most closely related is the work by Raz [25]. While both our work and [25] have the same high-level message, namely, that strong enough lower bounds for $\Sigma\Pi\Sigma$ circuits imply more general superpolynomial lower bounds, we make here significant quantitative improvements. We state this from the perspective of obtaining lower bounds. First, Raz's result would yield only superpolynomial *formula* lower bounds, while ours can yield circuit lower bounds. Second, Raz requires the degree of the output polynomial to be rather small—$d \leq \frac{\log n}{\log\log n}$—while our results are valid for much larger d, say, $d = n^{\Omega(1)}$. Most important, Raz requires almost optimal $\Sigma\Pi\Sigma$ lower bounds of about $2^{(d\log n)\cdot(1-o(1))}$, while for us a much weaker lower bound of $2^{\omega(\sqrt{d}\log n)}$ will suffice. But it is worth noting that Raz's work deals with proving strong lower bounds for tensors, a subclass of homogeneous depth-3 circuits, whereas our result deals with heavily nonhomogeneous circuits.

$\Sigma\Pi\Sigma$ circuits have been intensely investigated—initially for their connection to tensor rank and more recently as a special case of the polynomial identity testing problem. Given this intense investigation of $\Sigma\Pi\Sigma$ circuits, it is natural to wonder why these results were not obtained before. We feel that this may be because our construction is significantly counterintuitive—the intermediate terms are of degree much higher than the degree of the output polynomial and moreover ~~we need~~ the field to have zero or large characteristic. Finally, we remark here that as a tool for proving lower bounds, our result is somewhat incomparable to [21], while the reduction of the depth further should facilitate the task of proving lower bounds, the fact that the resulting $\Sigma\Pi\Sigma$ circuit is nonhomogeneous hinders it. Overall, it is not clear to us as to which of these two kinds of circuits is a better starting point.

we suffice

Further results. In section 5 we then present some further consequences and reductions. Following [2] we show that a blackbox derandomization of polynomial identity testing for $\Sigma\Pi\Sigma$ circuits leads to a quasi-polynomial identity test for general circuits. We also explore the relation of $\Sigma\Pi\Sigma$ circuits with a subclass of circuits that arise in our proof.

2. Proof overview. Quick sketch. The depth reduction proceeds through a series of transformations to the circuit—first decreasing the depth, then replacing the multiplication (×) gates with exponentiation gates at the expense of increasing the depth slightly and then decreasing the depth once again to three. We will flesh out this quick sketch after introducing some relevant notation.

Notation. We shall use the standard notation that for positive integer a, we shall use $[a]$ is the set of positive integers less than or equal to a. To avoid adding floors and ceilings, we shall abuse notation and use $[a]$ to refer to the set of positive integers less than or equal to $\lceil a \rceil$.

Bounded depth arithmetic circuits consist of alternating layers of addition and multiplication gates. We will denote an arithmetic circuit of depth d by a sequence of d symbols wherein each symbol (either Σ or Π) denotes the nature of the gates at the corresponding layer and the leftmost symbol indicates the nature of the gate at the output layer. For example, a $\Pi\Sigma$ circuit with n input variables computes a polynomial in the following way:

$$C(\mathbf{x}) = \prod_{i\in[d]} \left(\sum_{j\in[n]} a_{ij}x_j + a_{i0} \right), \quad \text{where each } a_{ij} \in \mathbb{F} \text{ is a field element,}$$

while a $\Sigma\Pi\Sigma$ circuit computes a polynomial as in (1.1). Some of the intermediate circuits that we construct will have the feature that all the incoming edges to a multiplication gate come from a single gate g (thus computing g^e, if there are e wires entering the multiplication gate). We will refer to such circuits as *powering circuits* and use $\wedge$ instead of Π to denote a layer of multiplication gates in such circuits. So for example, a $\Sigma\wedge\Sigma$ circuit computes a polynomial in the following manner:

$$C(\mathbf{x}) = \sum_{i\in[s]} \ell_i(\mathbf{x})^{e_i}, \quad \text{where each } \ell_i \in \mathbb{F}[\mathbf{x}] \text{ is an affine form.}$$

e_i given in unary; if in binary, this seems more powerful
(over circuits of the same depth)

In doing the transformations it is useful to keep track of the fanin to various gates, especially multiplication gates. Toward this end, we extend the above notation and allow integer superscripts on Π symbols (respectively, Σ and $\wedge$ symbols), which denotes an upper bound on the fanin of any gate in the corresponding layer. For example, a $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit computes a polynomial of the following form:

$$C = \sum_{i\in[s]} \prod_{j\in[a]} Q_{ij} \quad \text{with } \deg(Q_{ij}) \le b \text{ for all } i \in [s],\ j \in [a],$$

while a $\Sigma\wedge^{[a]}\Sigma$ circuit computes a polynomial in the following manner:

$$C(\mathbf{x}) = \sum_{i\in[s]} \ell_i(\mathbf{x})^{e_i}, \quad \text{where each } e_i \le a \text{ and each } \ell_i \in \mathbb{F}[\mathbf{x}] \text{ is an affine form.}$$

The size of a circuit would be defined as the number of wires in the circuit. With this notation in place we are ready to give a more detailed overview.

Proof overview. Let $f(\mathbf{x}) \in \mathbb{Q}[\mathbf{x}]$ be a polynomial of degree d computed by an arithmetic branching program, or arithmetic circuit of size s. The first step is to obtain from C a suitable depth-4 circuit C_1 computing f. Specifically, C_1 is a $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit for suitable a, b such that $ab \approx d$. This is achieved via Koiran's [21] strengthening of the Agrawal–Vinay [2] depth reduction, or by Tavenas' [33] improvement of Koiran's result.

The second step is to use C_1 to obtain a depth-5 *powering circuit* C_2. Specifically, C_2 is a $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit of size $s_2 = \text{poly}(s_1 \cdot 2^a \cdot 2^b)$. The main ingredient of this step is a lemma of Fischer [9], or Ryser's formula [26], showing how a monomial can be computed by a $\Sigma\wedge\Sigma$ circuit. In other words, the Ryser–Fischer lemma shows how to compute a product as a sum of powers of sums. By applying this lemma to every multiplication gate, we can convert the circuit C_1 to a $\Sigma\wedge\Sigma\wedge\Sigma$ circuit C_2. It is worth noting that the Ryser–Fischer lemma is not true over fields of small characteristic, and this is the main place where the field being zero characteristic becomes crucial.

The final step is to convert the $\Sigma\wedge\Sigma\wedge\Sigma$ circuit C_2 to a $\Sigma\Pi\Sigma$ circuit C_3, and this is done by invoking the "duality trick" of Saxena [28] and factoring the resulting univariate polynomials to get a $\Sigma\Pi\Sigma$ circuit over an extension of $\mathbb{Q}$ of not too large degree. Finally, from this we derive the required $\Sigma\Pi\Sigma$ circuit over $\mathbb{Q}$. Overall in the last step we increase the degree of the intermediate polynomials substantially—to $2^{O(\sqrt{d\log n\log s})}$. Our construction ensures that all the high ($>d$) degree monomials so generated ultimately cancel out. The resulting $\Sigma\Pi\Sigma$ circuit C_3 is of size $s_3 = 2^{O(\sqrt{d\log n\log s})}$.

3. Preliminaries.

This section will deal with the preliminaries required for the rest of the paper. As usual $[n]$ denotes the set of first n positive integers. For ease of bookkeeping, we define the size of a circuit as the number of wires[10] in the circuit.

Algebraic branching programs. An algebraic branching program (ABP) is a layered graph with edges going from layer i to $i+1$. Every edge e is labeled by a linear polynomial ℓ_e. The first layer has only one vertex, called the *source*, and the last layer has only one vertex, called the *sink*. For any path $\gamma = (e_1, \dots, e_d)$ from source to sink, the weight of γ is defined as $\mathrm{wt}(\gamma) = \ell_{e_1} \dots \ell_{e_d}$. The ABP is said to compute the polynomial $\sum_\gamma \mathrm{wt}(\gamma)$, where γ runs over all source-sink paths.

~ acyclic weighted automaton over the ring of polynomials

An ABP is said to be *homogeneous* if all edge labels are homogeneous linear forms, and it naturally computes a homogeneous polynomial.

We shall say that a (possibly nonhomogeneous) polynomial is computed by a homogeneous ABP if each of its homogeneous parts can be computed by a homogenous ABP. It is a well-known fact (cf. [32]) that if a degree d polynomial can be computed by a size s (possibly nonhomogeneous) ABP, then it can be computed by a homogeneous ABP of size sd^2.

The following conversion from circuits to ABPs is attributed by Koiran [21] to Malod and Portier [23] but can also be easily deduced directly from [35].

LEMMA 3.1. *Let f be a polynomial of degree d computed by a circuit of size s. Then there is a homogeneous ABP of depth d and size $s' = 2^{O(\log s\cdot\log d)}$ computing f.*

The following bound on binomial coefficients is an easy consequence of Stirling's approximation.

LEMMA 3.2 (cf. [2, Lemma 2.2] for a proof). *For any n, k,*

$$\binom{n+k}{k} = 2^{O(k\log n)}.$$

4. The depth reduction.

Let us recall the statement of our main result for branching programs.

THEOREM 4.1. *Let $f(\mathbf{x}) \in \mathbb{Q}[\mathbf{x}]$ be an n-variate polynomial of degree $d = n^{O(1)}$ computed by an arithmetic branching program of size s. Then it can also be computed by a $\Sigma\Pi\Sigma$ circuit of size $2^{O(\sqrt{d\log n\log s})}$.*

[10] One could alternatively define the size of a circuit as the number of nodes, but defining it this way is more natural for us as it simplifies the bookkeeping—for example, in tracking the size of an exponentiation gate.

We first observe that the corresponding depth reduction for general circuits (as given in the statement of Theorem 1.1) follows immediately from the above statement via an application of Lemma 3.1—we first convert the given circuit to a slightly larger ABP and then apply the depth reduction for ABPs. The rest of this section is devoted to the proof of the above theorem. So let f be an n-variate polynomial of degree d computed by an arithmetic circuit of size s. As mentioned in section 2, the depth reduction roughly proceeds through the following steps:

- Step 1. Arithmetic circuits/branching programs $\longrightarrow \Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuits.
- Step 2. $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuits $\longrightarrow \Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuits.
- Step 3. $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuits $\longrightarrow \Sigma\Pi\Sigma$ circuits.

It suffices to start with a homogeneous polynomial computed by a homogeneous circuit, since given any circuit of size s computing an arbitrary degree d polynomial f, we can construct $(d+1)$ homogeneous circuits that compute the homogeneous components of f of size at most $O(sd^2)$ via standard homogenization tricks (see [32, section 2.2] for an exposition).

In the rest of this section we shall provide the details on how to perform each of the above steps while keeping track of the loss in size incurred at each step.

4.1. Step 1: Reduction to $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit.

The first step is a direct consequence of the depth reduction result of Koiran [21]. Although Koiran's proof works specifically with $a = b = \sqrt{d}$, the following theorem can be easily inferred from [21, Proposition 4].

THEOREM 4.2 (see [21]). *Let f be a homogeneous n-variate polynomial of degree d computed by an arithmetic branching program of size s. Then, for all a there is an equivalent homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[(d/a)]}$ circuit computing f of size $s^a + (s^2 d) \cdot \binom{n+(d/a)}{(d/a)}$.*

Choosing $a = \sqrt{\frac{d \log n}{\log s}}$ (and using the bound of Lemma 3.2) gives the following.

COROLLARY 4.3. *Let f be a homogeneous n-variate degree d polynomial computed by arithmetic branching program of size s. Then, it can be equivalently computed by homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[d/a]}$ (for $a = \sqrt{\frac{d \log n}{\log s}}$) circuit C_1 computing f of size $s_1 = \exp\left(O(\sqrt{d \log n \log s})\right)$.*

Remark 1. The recent improvement of Tavenas [33] shows a reduction from general arithmetic circuits of size s to $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ with $b \leq 15(d/a)$ with the same asymptotic size as in Theorem 4.2. This directly translates to an improvement in Theorem 1.1, to get the same asymptotic size for both ABPs and circuits.

4.2. Step 2: $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit to $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit.

The next step relies on a variant of the construction due to Fischer [9], or Ryser's formula [26], to write a monomial as a sum of powers of linear functions which we describe below.

LEMMA 4.4 (see [26, 9]). *For any n, the monomial $x_1 \cdots x_n$ can be expressed as a linear combination of 2^n powers of linear forms through the following:*

$$n! \cdot x_1 \ldots x_n \quad = \quad \sum_{S \subseteq [n]} \left(\sum_{i \in S} x_i \right) (-1)^{n-|S|}. \tag{4.1}$$

Fischer [9] used a slightly different identity that achieves the same goal, and the above is just a special case of Ryser's formula. The proof of the above lemma is a simple inclusion-exclusion argument and can also be seen in [30]. We present a proof here for completeness.

Proof of Lemma 4.4. It is clear that the coefficient of $x_1 \dots x_n$ in the right-hand side (RHS) is exactly $n!$. Hence it suffices to show that every other monomial has coefficient zero in the RHS.

Let $m \neq x_1 \dots x_n$ be any degree n monomial over $x_1, \dots, x_n$. Since $m \neq x_1 \dots x_n$ and has degree n, there must be some variable x_r such that x_r does not divide m. To show that the coefficient of m in the RHS is zero, we shall pair each set $S \subseteq [n] \setminus \{r\}$ with $S \cup \{r\}$. This is clearly a bijective map between sets that do not contain r with sets that contain r. Note that for any $S \subseteq [n] \setminus \{r\}$, the coefficient of m contributed by $\left(\sum_{i \in S} x_i\right)^n$ is exactly equal to the coefficient of m contributed by $(\sum_{i \in S \cup \{r\}} x_i)^n$. Hence it follows that the coefficient of m in the RHS of (4.1) is zero. □

Since any degree d monomial can be obtained by making appropriate substitutions in $x_1 \dots x_d$, a corollary of the above lemma is that every degree d monomial can be expressed as a sum of 2^d powers of linear forms. It is also worth noting that the above lemma is not true in low characteristic. For example, in a field $\mathbb{F}$ of characteristic 2, $(x + \alpha y + \beta)^2 = x^2 + \alpha^2 y^2 + \beta^2$ and hence the monomial $x \cdot y$ cannot be expressed as a sum of squares of affine forms. With this lemma, we can now proceed to the transformation in step 2.

LEMMA 4.5. *Let f be a homogeneous n-variate degree d polynomial computed by a homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit of size s_1. Then, there is an equivalent $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit of size $(s_1^2 a^2 b^2 n) \cdot 2^{a+b}$ computing f.*

Proof. A $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit C of size s_1 computes a polynomial of the form

$$C = \sum_{i \in [s_1]} \prod_{j \in [a]} Q_{ij} \quad \text{where } \deg(Q_{ij}) \leq b.$$

Since the size of the circuit is s_1, each of the Q_{ij}'s has at most s_1 monomials. Further, their degree is at most b and hence we may apply Lemma 4.4 to each monomial and express Q_{ij} as a $\Sigma^{[s_1 2^b]} \wedge^{[b]} \Sigma^{[b]}$ circuit of size $(s_1 \cdot 2^b)(b \cdot b)$.

Now applying Lemma 4.4 to each term $T_i = \prod_{j \in [a]} Q_{ij}$ (where some of the Q_{ij}'s could be 1), we can express it as a sum of at most 2^a ath powers of linear combinations of the Q_{ij}'s. Thus each T_i can be expressed as a $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit of size $(2^a)(a^2)(s_1 2^b)(b^2)$. Since there are at most s_1 terms T_i, we get overall a $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit of size at most $(s_1^2 a^2 b^2) \cdot 2^{a+b}$. □

Remark 2. A small observation here is that by ensuring that the bottom product gates have fan-in exactly b (by padding with ones if required), we can ensure that the lower $\wedge$ gate of the resulting $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuit has fan-in exactly b.

Combining this with corollary 4.3 we immediately get the following.

COROLLARY 4.6. *Let f be a homogeneous n-variate polynomial of degree d computed by an arithmetic branching program of size s. Then there is an equivalent $\Sigma\wedge^{[a]}\Sigma\wedge^{[d/a]}\Sigma$ circuit C_2 (for $a = \sqrt{\frac{d \log n}{\log s}}$) computing f of size s_2 at most* $\exp\left(O(\sqrt{d \log n \log s})\right)$.

4.3. Step 3: $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuits to $\Sigma\Pi\Sigma$ circuits.

The final step of the proof uses the *duality trick* of Saxena [28]. A similar version of this trick was also discovered by Shpilka and Wigderson [31] in a different context. The statement below is slightly different from the statement in [28], and the proof given below is by Forbes, Gupta, and Shpilka [10]. This alternate formulation will help simplify the proof of the main theorem.

Lemma 4.7 (see [28, 10]). *For every $m, d > 0$ and distinct $\alpha_0, \dots, \alpha_{md} \in \mathbb{Q}$, there exists $\beta_0, \dots, \beta_{md} \in \mathbb{Q}$ such that*

$$(u_1 + \cdots + u_m)^d \quad = \quad \sum_{i=0}^{md} \sum_{j=0}^{d} \beta_{ij} (u_1 + \alpha_i)^j \dots (u_m + \alpha_i)^j.$$

Proof. Let $P_{\mathbf{u}}(z)$ be defined as follows:

$$\begin{aligned} P_{\mathbf{u}}(z) &\overset{\text{def}}{=} (z + u_1) \dots (z + u_m) - z^m \\ &= z^{m-1}(u_1 + \cdots + u_m) \quad + \quad \text{(lower degree terms)} \\ \implies P_{\mathbf{u}}(z)^d &= z^{(m-1)d}(u_1 + \cdots + u_m)^d \quad + \quad \text{(lower degree terms).} \end{aligned}$$

Hence, we can compute $(u_1 + \cdots + u_m)^d$ from sufficiently many evaluations of $P_{\mathbf{u}}(z)^d$. That is, for every distinct $\alpha_0, \dots, \alpha_{md} \in \mathbb{Q}$ there exist $\beta'_0, \dots, \beta'_{md} \in \mathbb{Q}$ such that

$$\begin{aligned} (u_1 + \cdots + u_m)^d &= \sum_{i=0}^{md} \beta'_i P_{\mathbf{u}}(\alpha_i)^d \\ &= \sum_{i=0}^{md} \beta'_i \left((\alpha_i + u_1) \dots (\alpha_i + u_m) - \alpha_i^m\right)^d \\ &= \sum_{i=0}^{md} \beta'_i \sum_{j=0}^{d} \binom{d}{j} \cdot (-\alpha_i^m)^{(d-j)} \cdot \left((\alpha_i + u_1) \dots (\alpha_i + u_m)\right)^j \\ &= \sum_{i=0}^{md} \sum_{j=0}^{d} \beta_{ij} \left((\alpha_i + u_1) \dots (\alpha_i + u_m)\right)^j , \end{aligned}$$

where $\beta_{ij} = \beta'_i \cdot \binom{d}{j} \cdot (-\alpha_i^m)^{(d-j)}$. □

With the above lemma, we can proceed to Step 3. First we obtain a $\Sigma\Pi\Sigma$ circuit over $\mathbb{C}$, the field of complex numbers, and subsequently convert the $\Sigma\Pi\Sigma$ circuit over $\mathbb{C}$ to one over $\mathbb{Q}$. Using the remark following the proof of Lemma 4.5, it suffices to consider $\Sigma\wedge^{[a]}\Sigma\wedge^{[b]}\Sigma$ circuits where the lower $\wedge$ gates have fan-in exactly b. We shall denote such circuits by $\Sigma\wedge^{[a]}\Sigma\wedge^{[=b]}\Sigma$ for brevity.

Lemma 4.8. *Let f be a polynomial computed by a $\Sigma\wedge^{[a]}\Sigma\wedge^{[=b]}\Sigma$ circuit of size s_2 over $\mathbb{Q}$. Then, there is an equivalent $\Sigma\Pi\Sigma$ circuit C_3 over $\mathbb{C}$ of size $s_3 = O(s_2^3 a^3 bn)$ computing f. The circuit C_3 has top fan-in $O(s_2^2 a^2)$ and formal degree at most $O(s_2 ab)$.*

Proof. A $\Sigma\wedge^{[a]}\Sigma\wedge^{[=b]}\Sigma$ circuit C computes a polynomial of the form $C = T_1 + \cdots + T_{s_2}$, where each $T_i = \left(\ell_{i1}^b + \cdots + \ell_{is_2}^b\right)^a$ for some linear forms ℓ_{ij}'s. Applying Lemma 4.7 to each such term $T = (\ell_1^b + \cdots + \ell_{s_2}^b)^a$, we can write T as

$$\begin{aligned} T &= \sum_{i=0}^{s_2 a} \sum_{j=0}^{a} \beta_{ij} \prod_{k=1}^{s_2} (\alpha_i + \ell_k^b)^j \\ &= \sum_{i=0}^{s_2 a} \sum_{j=0}^{a} \beta_{ij} \prod_{k=1}^{s_2} (f_i(\ell_k))^j, \qquad \text{where } f_i(t) \overset{\text{def}}{=} (\alpha_i + t^b). \end{aligned}$$

Since each $f_i(t) = (\alpha_i + t^b)$ is a univariate polynomial of degree exactly b, it splits as a product of linear factors over $\mathbb{C}$, yielding a depth-3 circuit of the form

$$T = \sum_{i=1}^{(s_2a+1)\cdot a} \prod_{j=1}^{s_2} \prod_{k=1}^{ab} (\ell_j - \gamma_{ik}) . \tag{4.2}$$

Thus, f can be computed by a $\Sigma\Pi\Sigma$ circuit of top fan-in $s_2 \cdot (s_2a+1) \cdot a = O(s_2^2a^2)$ and degree $O(s_2ab)$, thereby yielding an overall size of

$$s_3 = s_2 \cdot (s_2a+1) \cdot a \cdot O(s_2ab) \cdot (n+1) = (s_2^3a^3bn). \qquad \square$$

4.3.1. Obtaining a $\boldsymbol{\Sigma\Pi\Sigma}$ circuit over rationals. The circuit thus obtained from Lemma 4.8 involves coefficients that are roots of the univariates $f_i(t) = (\alpha_i + t^b)$. The following lemma shows that if all α_i's are chosen carefully, all these coefficients come from an algebraic extension of $\mathbb{Q}$ of relatively small degree.[11]

LEMMA 4.9. *Let ω_b be a principal bth root of unity. Then, the field $\mathbb{Q}(\omega_b)$ contains the roots of every $f_i(t) = (t^b + \alpha_i)$ for every $\alpha \in \mathbb{Q}$ such that $(-\alpha)^{1/b} \in \mathbb{Q}$.*

Proof. As $(t^b + \alpha) = \prod_{i=1}^{b}(t - (-\alpha)^{1/b} \cdot \omega_b^i)$, it is clear that all coefficients are in $\mathbb{Q}(\omega_b)$ if $(-\alpha)^{1/b} \in \mathbb{Q}$. $\square$

Since we are free to choose the α_i's in Lemma 4.8, we can choose distinct α_i's so that $(-\alpha_i)^{1/b} \in \mathbb{Q}$. Hence, the coefficients in the depth-3 circuit obtained from Lemma 4.8 come from $\mathbb{K} := \mathbb{Q}(\omega_b)$. Note that $\mathbb{K}$ is an extension over $\mathbb{Q}$ of degree at most b.

The following lemma gives a generic way of converting a circuit involving coefficients from a small extension to a circuit over the base field.

LEMMA 4.10. *Let $f(\mathbf{x}) \in \mathbb{Q}[\mathbf{x}]$ be computed by a $\Sigma\Pi\Sigma$ circuit of top fanin s_3 and formal degree D with coefficients coming from a finite extension field $\mathbb{K}/\mathbb{Q}$. Then, there is an equivalent $\Sigma\Pi\Sigma$ circuit computing f of size* $\mathrm{poly}(s_3, D, [\mathbb{K} : \mathbb{Q}])$ *with coefficients coming from $\mathbb{Q}$.*

With this lemma, using the fact that the degree of the extension containing all coefficients is b and Lemma 4.8, we immediately get our main result, Theorem 4.1.

Proof of Lemma 4.10. Since $\mathbb{K}$ is a finite extension of $\mathbb{Q}$, there exists an element $\theta \in \mathbb{K}$ such that $\mathbb{K} = \mathbb{Q}(\theta)$.[12] Hence, if $m = [\mathbb{K} : \mathbb{Q}]$, then $\mathbb{K}$ is the vector space over $\mathbb{Q}$ with $\left\{1, \theta, \theta^2, \ldots, \theta^{m-1}\right\}$ as a basis and the minimum polynomial of θ has degree m. Therefore, any polynomial $g(\mathbf{x}) \in \mathbb{K}[\mathbf{x}]$ can be uniquely written as $g^{[0]} + g^{[1]}\theta + \cdots + g^{[m-1]}\theta^{m-1}$, where each $g^{[i]} \in \mathbb{Q}[\mathbf{x}]$.

If $f = T_1 + \cdots + T_{s_3}$, where each T_i is a product of linear polynomials over $\mathbb{K}$, then $f = T_1^{[0]} + \cdots + T_{s_3}^{[0]}$. Hence it suffices to show that $T_i^{[0]}$'s can be expressed as small depth-3 circuits over $\mathbb{Q}$.

Let $T = \ell_1 \ldots \ell_D \in \mathbb{K}[\mathbf{x}]$. Then,

$$\begin{aligned} T &= \prod_{i=1}^{D}(\ell_i^{[0]} + \ell_i^{[1]}\theta + \cdots + \ell_i^{[m-1]}\theta^{m-1}) \\ &= T^{[0]} + T^{[1]}\theta + \cdots + T^{[m-1]}\theta^{m-1}. \end{aligned}$$

[11] Basic facts about field extensions, their degrees, etc., can be found in Chapter 5 of [17].

[12] Follows from the primitive element theorem (Theorem 5.5.1 in [17]).

Consider the polynomial obtained from above by replacing θ by a formal variable y:

$$\begin{aligned}\tilde{T}(\mathbf{x}, y) &= \prod_{i=1}^{D}(\ell_i^{[0]} + \ell_i^{[1]}y + \cdots + \ell_i^{[m-1]}y^{m-1})\\ &= \tilde{T}_0 + \tilde{T}_1 y + \cdots + \tilde{T}_{(m-1)D}\, y^{(m-1)D}.\end{aligned}$$

Therefore, using interpolation, every $\tilde{T}_i$ can be written as a linear combination of the set $\{\tilde{T}(\mathbf{x}, \alpha_i)1 \leq i \leq (m-1)D+1\}$. Since each such evaluation is a product of D linear polynomials over $\mathbb{Q}$, we have that each $\tilde{T}_i$ can be expressed as a depth-3 circuit of top fanin $(m-1)D+1$ and formal degree D.

To obtain $T^{[0]}, \ldots, T^{[m-1]}$ from $\tilde{T}_0, \ldots, \tilde{T}_{(m-1)D}$, we can express each θ^i for $m \leq i \leq (m-1)D$ as a linear combination of $1, \ldots, \theta^{m-1}$ to obtain that each $T^{[i]}$ is the appropriate linear combination of $\tilde{T}_0, \ldots, \tilde{T}_{(m-1)D}$. Therefore, in particular, $T^{[0]}$ can be expressed as a depth-3 circuit over $\mathbb{Q}$ of top fanin $((m-1)D) \cdot ((m-1)D+1)$ and formal degree D. Hence, f can be expressed as a depth-3 circuit over $\mathbb{Q}$ of top fanin $O(s_3 m^2 D^2)$ and formal degree D. □

5. Further consequences.

5.1. Depth reduction to polynomial identity testing lift. Any depth reduction to a class $\mathcal{C}$ of circuits provides a framework for lifting a polynomial time black-box polynomial identity testing (PIT) for the class $\mathcal{C}$ to black-box PITs for general circuits with *slightly worse* running time. We now present such a lift in the context of the depth reduction in Theorem 1.1 on exactly the same lines as in [2]. We first define the problem of black-box PIT.

DEFINITION 5.1 (PIT). *The problem of* PIT *is the algorithmic question wherein, given an arithmetic circuit C as input, the task is to check if the polynomial computed by C is the zero polynomial or not.*

The problem of black-box PIT is the same algorithmic question but given only black-box or oracle access to evaluate the circuit C. Thus, black-box polynomial identity testing is equivalent to construct a hitting set *$\mathcal{H} \subseteq \mathbb{F}^n$ such that any nonzero circuit C of size s is guaranteed to evaluate to a nonzero value on some $\mathbf{a} \in \mathcal{H}$.*

The following result roughly states that any black-box PIT for a class yields a lower bound for the same class. The following is a slight rephrasing of [1, Theorem 55].

LEMMA 5.2 (see [16, 1]). *Let $\{\mathcal{C}_n\}$ be any subclass of arithmetic circuits computing n-variate degree n polynomials, and suppose there is a black-box PIT running in time $n^{O(1)}$ for the circuits of size n^2 in $\mathcal{C}_n$. Then, there is a family of multilinear polynomials $\{q_n\}$ such that q_n requires $\mathcal{C}_n$-circuits of size $2^{\Omega(n)}$. Further, q_n is computable in time $2^{O(n)}$, i.e., the set of all monomials and its coefficients can be listed out in $2^{O(n)}$ time.*

The next lemma of Kabanets and Impagliazzo [18, Theorem 7.7] states that given any family of polynomials that require exponential sized *general* circuits to compute them, one can construct a quasi-polynomial black-box PIT for *general* circuits.

LEMMA 5.3 (see [18]). *Suppose $\{q_n\}$ is a family of multilinear polynomials computable in exponential time such that q_n requires arithmetic circuits of size $2^{n^{\Omega(1)}}$. Then, one can construct a black-box PIT for general arithmetic circuits running in time $2^{(\log n)^{O(1)}}$.*

Suppose we did have a polynomial time black-box PIT for a class $\mathcal{C}_n$; then Lemma 5.2 gives a family $\{q_n\}$ that requires $2^{\Omega(n)}$-sized $\mathcal{C}_n$-circuits. Lemma 5.3, however, requires a family of polynomials $\{q_n\}$ that requires $2^{\Omega(n)}$-sized *general* circuits. If we could say that $\{q_n\}$ requiring $2^{\Omega(n)}$-sized $\mathcal{C}_n$-circuits *implies* $\{q_n\}$ requires $2^{n^{\Omega(1)}}$-sized general circuits, then we would be done. Such a statement is precisely the contrapositive of reducing a general circuit to a $\mathcal{C}_n$-circuit.

Theorem 5.4. *If there is a polynomial time black-box PIT for the class of depth-3 circuits, then there is a $2^{(\log n)^{O(1)}}$ time black-box PIT for general circuits computing a low degree polynomial.*[13]

Proof. By Lemma 5.2, a polynomial time black-box PIT for depth-3 circuits implies that there is a family of multilinear polynomials $\{q_n\}$ that require $2^{\Omega(n)}$-sized depth-3 circuits. Theorem 1.1 says that the family $\{q_n\}$ requires general circuits of size $2^{n^{\Omega(1)}}$. Using Lemma 5.3, we obtain a $2^{(\log n)^{O(1)}}$ time black-box PIT for general circuits. □

Saha, Saptharishi, and Saxena [27] showed that PIT for width-2 branching programs completely captures PIT for depth-3 circuits. A corollary for the above theorem is that polynomial time black-box PIT for width-2 ABPs implies quasi-polynomial time black-box PIT for general circuits.

Corollary 5.5. *If there is a polynomial time black-box PIT for the class of width-2 ABPs over $\mathbb{Q}$, then there is a $2^{(\log n)^{O(1)}}$ time black-box PIT for general circuits computing a low degree polynomial.*

5.2. Reduction from $\Sigma\Pi\Sigma$ circuits to $\Sigma\wedge\Sigma\wedge\Sigma$ circuits.

For the purpose of proving superpolynomial circuit lower bounds over $\mathbb{Q}$, we can equivalently work with any of $\Sigma\Pi\Sigma$ or $\Sigma\wedge\Sigma\wedge\Sigma$ circuits. We will need the following families of symmetric polynomials:

$$\mathsf{Sym}_n(x_1,\ldots,x_m) \overset{\text{def}}{=} \sum_{\substack{S\subseteq[m]\\|S|=n}} \prod_{i\in S} x_i,$$

$$\mathsf{Pow}_n(x_1,\ldots,x_m) \overset{\text{def}}{=} \sum_{j\in[m]} x_j^n.$$

Our proof relies on the following implication of Newton's identities (which can be found in [19], for example).

Lemma 5.6. *Let $\mathsf{Sym}_n(x_1,\ldots,x_m)$ and $\mathsf{Pow}_n(x_1,\ldots,x_m)$ denote the elementary symmetric and power symmetric polynomials of degree n, respectively, as defined above. Then,*

$$\mathsf{Sym}_n \quad = \quad \frac{1}{n!}\times\begin{vmatrix} \mathsf{Pow}_1 & 1 & 0 & \cdots & \\ \mathsf{Pow}_2 & \mathsf{Pow}_1 & 2 & 0 & \cdots \\ \vdots & & \ddots & \ddots & \\ \mathsf{Pow}_{n-1} & \mathsf{Pow}_{n-2} & \cdots & \mathsf{Pow}_1 & n-1 \\ \mathsf{Pow}_n & \mathsf{Pow}_{n-1} & \cdots & \mathsf{Pow}_2 & \mathsf{Pow}_1 \end{vmatrix}.$$

We now give a partial converse of Lemma 4.8.

Theorem 5.7. *Let f be an N-variate degree n polynomial over any characteristic zero field computed by a $\Sigma\Pi\Sigma$ circuit with top fan-in s and product gates with fan-in at most d. Then, there exists an equivalent $\Sigma\wedge\Sigma\wedge\Sigma$ circuit of size* $\text{poly}(N,d,s)\cdot 2^{O(\sqrt{n}\cdot\log n)}$.

[13] Here, low degree means degree of $n^{O(1)}$.

Further, all intermediate computations of the $\Sigma\wedge\Sigma\wedge\Sigma$ *circuit have degree at most* $\deg(f)$.

Proof. We are given that there exist $s \cdot d$ linear polynomials $\left\{\ell'_{ij}\right\}$ such that

$$f = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell'_{ij}.$$

As we are working over an infinite field, we can assume without loss of generality that each ℓ'_{ij} has a nonzero constant term. This is because constructing a depth-5 powering circuit for an *affine shift* of f gives a depth-5 powering circuit for f of at most polynomially larger size. Hence f has an expression of the form

$$f = \sum_{i=1}^{s} \alpha_i \prod_{j=1}^{d} (1 + \ell_{ij}).$$

Let $f^{[r]}$ denote the degree-r homogeneous component of f. Then

$$f^{[r]} = \sum_{i=1}^{s} \alpha_i \cdot \mathsf{Sym}_r(\ell_{i1}, \ldots, \ell_{id}).$$

We now focus on a summand of the form $\mathsf{Sym}_r(\ell_1, \ldots, \ell_d)$. From Lemma 5.6, there exist scalars $\beta_{\mathbf{a}}$'s such that

$$\mathsf{Sym}_r(\ell_1, \ldots, \ell_d) \;=\; \sum_{\substack{\mathbf{a}=(a_1,\ldots,a_r)\in\mathbb{Z}^r_{\geq 0} \\ \sum_i i\cdot a_i = r}} \beta_{\mathbf{a}} \cdot \prod_{i\in[r]} \mathsf{Pow}_i^{a_i}(\ell_1, \ldots, \ell_d). \tag{5.1}$$

The number of solutions of $\sum_{i\in[r]} i \cdot a_i = r$ is exactly the number of ways to partition the natural number r and hence is $2^{O(\sqrt{r})}$ by the Hardy–Ramanujan estimate for the partition function [15]. Hence the number of terms in the above summation is $2^{O(\sqrt{r})}$.

The next step is to convert the product into a powering gate, similar to Step 2 in the depth reduction which used the Ryser–Fischer construction [9]. However, we shall need a more efficient way of converting a "low support" monomial into a sum of powers. Intuitively, the Ryser–Fischer trick applied on a term of the form $x_1^n \ldots x_n^n$ would yield a $\Sigma\wedge\Sigma$ circuit of size $2^{O(n^2)}$. However, a more careful analysis of the number of distinct linear polynomials used in Lemma 4.4 yields a better bound of $2^{O(n \log n)}$. Such a bound is provided by the following lemma of Ellison [8, Theorem 1].

LEMMA 5.8 (see [8]). *Over any field of zero characteristic, any homogeneous* n*-variate degree* d *polynomial can be expressed as a linear combination of* d*th powers of linear forms. Further, the number of such powers of linear forms used is bounded by* $(d+1)^{n-1} = 2^{O(n \log d)}$.

If $y_i \overset{\text{def}}{=} \mathsf{Pow}_i(\ell_1, \ldots, \ell_d)$, then (5.1) can be written as

$$\mathsf{Sym}_r(\ell_1, \ldots, \ell_d) \quad = \quad \sum_{\substack{\mathbf{a}=(a_1,\ldots,a_r)\in\mathbb{Z}^r_{\geq 0} \\ \sum_i i\cdot a_i = r}} \beta_{\mathbf{a}} \cdot \prod_{i\in[r]} y_i^{a_i}.$$

Note that if $\sum_{i\in[r]} i \cdot a_i = r$, then at most $O(\sqrt{r})$ of the a_i's are nonzero. Hence, each of the monomials (in the y_i's) in the RHS of the above equation has support at most

$O(\sqrt{r})$ and degree at most r. Hence, applying Lemma 5.8 to each of the monomials gives a representation of the form

$$\mathsf{Sym}_r(\ell_1,\ldots,\ell_d) \quad = \quad \sum_{\substack{\mathbf{a}\in\mathbb{Z}^r_{\geq 0}\\ \sum_i i\cdot a_i = r}} \beta_{\mathbf{a}} \cdot \left(\sum_{i=1}^{2^{O(\sqrt{r}\log r)}} \gamma_i \left(\sum_{j=1}^{r} \alpha_{ij} y_j \right)^{\sum a_j} \right)$$

for some scalars γ_i, α_{ij}, which is a $\Sigma\wedge\Sigma$ circuit in the y_i's of size $2^{O(\sqrt{r}\log r)}$. Substituting each y_i by $\mathsf{Pow}_i(\ell_1,\ldots,\ell_d)$, we get a $\Sigma\wedge\Sigma\wedge\Sigma$ circuit for $\mathsf{Sym}_r(\ell_1,\ldots,\ell_d)$ of size bounded by $\mathrm{poly}(N,d)\cdot 2^{O(\sqrt{r}\log r)}$. Since each $f^{[r]}$ is a linear combination of s such terms, and $f = f^{[0]} + \cdots + f^{[n]}$, we have that f can be computed by a $\Sigma\wedge\Sigma\wedge\Sigma$ circuit of size $\mathrm{poly}(N,d,s)\cdot 2^{O(\sqrt{n}\log n)}$. □

6. Conclusion. Recently, Tavenas [33] has improved the depth reduction to depth-4 circuits (Step 1 of our result) by showing that any n-variate homogeneous polynomial of degree d that is computable by an arithmetic circuit of size s can be equivalently computed by a $\Sigma\Pi^{[15\sqrt{d}]}\Sigma\Pi^{[\sqrt{d}]}$ of size $\exp(O(\sqrt{d\log n\log s}))$. He further observed that plugging this improvement into Step 1 of our result yields an improved reduction to depth 3 as well.

We saw that powering circuits are useful as an intermediate step in doing depth reduction of general circuits. We feel that proving lower bounds for powering circuits will also be useful as a good intermediate step toward proving lower bounds for general circuits. Toward this we recall a problem posed in [5] in the form of a more specific conjecture.

CONJECTURE 6.1. *Over any field $\mathbb{F}$ of characteristic 0, any depth-Δ powering circuit with* $\mathrm{poly}(d)$*-bounded degree computing the monomial $x_1\cdots x_d$ must be of size at least $2^{d^{\Omega(1/\Delta)}}$.*

Also, Step 1 and Step 2 of the depth reduction reduce any circuit to a depth-5 powering circuit with formal degree bounded by the degree of the polynomial. It is only Step 3 that results in the blow-up in degree. Hence, proving a $2^{\omega(\sqrt{d}\log d)}$ lower bound for low (formal) degree depth-5 powering circuits computing Perm_d would yield superpolynomial formula lower bounds. Proving such a lower bound for homogeneous depth-5 powering circuits might be an easier task than proving lower bounds for nonhomogeneous depth-3 circuits (where we have no degree bound). We pose the following problem as the next potential step toward proving superpolynomial formula lower bounds for Perm_d.

PROBLEM 6.2. *Lower bounds for structured $\Sigma\wedge\Sigma\wedge\Sigma$ circuits. Find an explicit n-variate degree d polynomial f such that any expression of the form*

$$f \quad = \quad \sum_{i=1}^{s} \left(\sum_{j=1}^{t} \ell_{ij}^{\sqrt{d}} \right)^{\sqrt{d}} \qquad \textit{with } t = \mathrm{poly}(n,d)$$

must have $s = n^{\omega(\sqrt{d})}$.

We already know of an $n^{\Omega(\sqrt{d})}$ lower bound [20, 11] on s, independent of how large t is. Presumably using the internal structure of the polynomials, say, the fact that each of the inner terms has a *low partial derivative space*, we should be able to prove better lower bounds. We believe that any technique to solve Problem 6.2 would be very insightful toward the grander goal of proving superpolynomial circuit lower bounds.

Summary. $\Sigma\Pi\Sigma$ circuits have been intensely investigated. Some recent work led to both structural results on $\Sigma\Pi\Sigma$ identities as well as deterministic blackbox identity testing algorithms for various subclasses of $\Sigma\Pi\Sigma$ circuits. We also note here that the Sylvester–Gallai configurations that arose in this line of work have led to some beautiful and productive series of works [3, 7] in understanding such geometric configurations culminating in essentially optimal upper and lower bounds on the dimension of such configurations. With these developments, we are inclined toward interpreting Theorem 1.1 optimistically. Proving lower bounds for $\Sigma\Pi\Sigma$ circuits is yet another promising route to obtaining lower bounds for arbitrary arithmetic circuits.

Acknowledgments. We are thankful to Avi Wigderson for several useful comments and suggestions on the initial draft of this write-up. We would also like to thank Klim Efremenko for pointing out some typos in the initial draft.

REFERENCES

[1] M. Agrawal, *Proving lower bounds via pseudo-random generators*, in Proceedings of Foundations of Software Technology and Theoretical Computer Science Science (FSTTCS), 2005, pp. 92–105.

[2] M. Agrawal and V. Vinay, *Arithmetic circuits: A chasm at depth four*, in Proceedings of Foundations of Computer Science (FOCS), 2008, pp. 67–75.

[3] B. Barak, Z. Dvir, A. Yehudayoff, and A. Wigderson, *Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes*, in Proceedings of Symposium on Theory of Computing (STOC), 2011, pp. 519–528.

[4] S. J. Berkowitz, *On computing the determinant in small parallel time using a small number of processors*, Inform. Process. Lett., 18 (1984), pp. 147–150.

[5] X. Chen, N. Kayal, and A. Wigderson, *Partial Derivatives in Arithmetic Complexity*, Found. Trends Theor. Comput. Sci. 6, now Publishers, Hanover, MA, 2011.

[6] L. Csanky, *Fast parallel inversion algorithm*, SIAM J. Comput., 5 (1976), pp. 618–623.

[7] Z. Dvir, S. Saraf, and A. Wigderson, *Improved rank bounds for design matrices and a new proof of Kelly's theorem*, Forum Math. Sigma, preprint, arXiv:1211.0330 [math.CO], 2012.

[8] W. Ellison, *A 'Waring's problem' for homogeneous forms*, Proc. Cambridge Philos. Soc., 65 (1969), pp. 663–672.

[9] I. Fischer, *Sums of like powers of multivariate linear forms*, Math. Mag., 67 (1994), pp. 59–61.

[10] M. Forbes, A. Gupta, and A. Shpilka, *private communication*, 2013.

[11] H. Fournier, N. Limaye, G. Malod, and S. Srinivasan, *Lower bounds for depth* 4 *formulas computing iterated matrix multiplication*, in Proceedings of Symposium on Theory of Computing (STOC), 2014, pp. 128–135.

[12] D. Grigoriev and M. Karpinski, *An exponential lower bound for depth* 3 *arithmetic circuits*, in Proceedings of Symposium on Theory of Computing (STOC), 1998, pp. 577–582.

[13] D. Grigoriev and A. A. Razborov, *Exponential lower bounds for depth* 3 *arithmetic circuits in algebras of functions over finite fields*, Appl. Algebra Engg. Comm. Comput., 10 (2000), pp. 465–487.

[14] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, *Approaching the chasm at depth four*, J. ACM, 61 (2014), pp. 33:1–33:16.

[15] G. H. Hardy and S. Ramanujan, *Asymptotic formula? in combinatory analysis*, Proc. Lond. Math. Soc., s2-17 (1918), pp. 75–115.

[16] J. Heintz and C.-P. Schnorr, *Testing polynomials which are easy to compute (Extended abstract)*, in Proceedings of Symposium on Theory of Computing (STOC), 1980, pp. 262–272.

[17] I. Herstein, *Topics in Algebra*, 2nd ed., Wiley, New York, 1975.

[18] V. Kabanets and R. Impagliazzo, *Derandomizing polynomial identity tests means proving circuit lower bounds*, Comput. Complexity, 13 (2004), pp. 1–46.

[19] D. Kalman, *A Matrix Proof of Newton's Identities*, Math. Mag., 73 (2000), pp. 313–315.

[20] N. Kayal, C. Saha, and R. Saptharishi, *A super-polynomial lower bound for regular arithmetic formulas*, in Proceedings of Symposium on Theory of Computing (STOC), 2014, pp. 146–153.

[21] P. Koiran, *Arithmetic circuits: The chasm at depth four gets wider*, Theoret. Comput. Sci., 448 (2012), pp. 56–65.

[22] M. Mahajan and V. Vinay, *A combinatorial algorithm for the determinant*, in Procedings of Symposium on Discrete Algorithms (SODA), 1997, pp. 730–738.

[23] G. Malod and N. Portier, *Characterizing valiant's algebraic complexity classes*, J. Complexity, 24 (2008), pp. 16–38.

[24] N. Nisan and A. Wigderson, *Lower bounds on arithmetic circuits via partial derivatives*, Comput. Complexity, 6 (1997), pp. 217–234.

[25] R. Raz, *Tensor-rank and lower bounds for arithmetic formulas*, J. ACM, 60 (2013), 40.

[26] H. J. Ryser, *Combinatorial Mathematics*, Carus Math. Monogr. 14, Math. Association of America, Washington, DC, 1963.

[27] C. Saha, R. Saptharishi, and N. Saxena, *The power of depth* 2 *circuits over algebras*, in Proceedings of Foundations of Software Technology and Theoretical Computer Science Science (FSTTCS), 2009, pp. 371–382.

[28] N. Saxena, *Diagonal circuit identity testing and lower bounds*, in Proceedings of International Colloquium on Automata, Languages, and Programming (ICALP), 2008, pp. 60–71.

[29] A. Shpilka, *Lower Bounds for Small Depth Arithmetic and Boolean Circuits*, Ph.D. thesis, Hebrew University, Jerusalem, 2001.

[30] A. Shpilka, *Affine projections of symmetric polynomials*, J. Comput. System Sci., 65 (2002), pp. 639–659.

[31] A. Shpilka and A. Wigderson, *Depth*-3 *arithmetic circuits over fields of characteristic zero*, Comput. Complexity, 10 (2001), pp. 1–27.

[32] A. Shpilka and A. Yehudayoff, *Arithmetic circuits: A survey of recent results and open questions*, Found. Trends Theor. Comput. Sci., 5 (2010), pp. 207–388.

[33] S. Tavenas, *Improved bounds for reduction to depth* 4 *and depth* 3, in Proceedings of Mathematical Foundations of Computer Science, 2013, pp. 813–824.

[34] L. G. Valiant, *Completeness classes in algebra*, in Proceedings of Symposium on Theory of Computing (STOC), 1979, pp. 249–261.

[35] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff, *Fast parallel computation of polynomials using few processors*, SIAM J. Comput., 12 (1983), pp. 641–644.

[36] A. Wigderson, *P, NP and Mathematics—A computational complexity perspective*, in Proceedings of the ICM 06 (Madrid), vol. 1, EMS Publishing House, Zurich, 2007, pp. 665–712.