Arithmetic Circuits: A Chasm at Depth Four

Manindra Agrawal
Indian Institute of Technology, Kanpur*

V Vinay
Geodesic Information Systems Ltd, Bangalore, and
Chennai Mathematical Institute, Chennai

August 12, 2008

Abstract

We show that proving exponential lower bounds on depth four arithmetic circuits imply exponential lower bounds for unrestricted depth arithmetic circuits. In other words, for exponential sized circuits additional depth beyond four does not help.

We then show that a complete black-box derandomization of Identity Testing problem for depth four circuits with multiplication gates of small fanin implies a nearly complete derandomization of general Identity Testing.

1 Introduction

The permanent, by virtue of being complete for $\#P$ [21], occupies a central position in the study of the complexity of counting problems. Its illustrious sibling, the determinant is comparatively easy, being complete for $GapL$, a complexity class housed within NC^2 [4, 19, 24].

The difference between the computation complexity of the permanent and the determinant has been among the most intriguing mathematical questions of our times. While we know determinant is easy, it has been difficult to prove any non-trivial lower bounds against the permanent.

In reality, a variety of lower bounds have been proved in restricted settings. Jerrum and Snir [8], and more recently improved by Raz and Yehudayoff [16], show that any monotone circuit computing the permanent requires exponential size. Nisan and Wigderson [14] show that any depth three circuit computing $2d$th symmetric polynomial requires $(\frac{n}{4d})^{\Omega(d)}$ size. Shpilka and Wigderson [18] show that any depth three circuit computing the permanent and determinant over the rationals requires quadratic size. Grigoriev and Razborov [6, 7] show that any depth three arithmetic circuit over a finite field computing the permanent or the determinant requires exponential size. Raz [15] shows every multilinear forumla computing permanent and determinant requires superpolynomial size. All of these proofs already require mathematical intricate machinery.

*This research was partially supported by J. C. Bose fellowship FLW/DST/CS/20060225

Another path to potential lower bounds was discovered by Kabanets and Impagliazzo [9]. Ever since they showed a remarkable connection between efficient polynomial identity testing (PIT) and arithmetic ciruit lower bounds, identity testing has invited closer scrutiny. For example, Kayal and Saxena [12] and Saxena [17], Karnin and Shpilka [11] show how certain restricted depth three circuits have deterministic polynomial time identity tester. Also, Dvir, Shpilka, and Yehudayoff [5] show how the Kabanets and Impagliazzo results can be extended to bounded-depth circuits.

Interestingly, a number of the above results are restricted to depth three circuits. Why not four or five or six? The reason, and we show it in this paper, is that crossing the chasm at depth four is as hard as the general case!

Depth Four Chasm: If a polynomial $P(x_1, ..., x_n)$ of degree d with $d = O(n)$ can be computed by an arithmetic circuit of size $2^{o(d+d\log\frac{n}{d})}$, it can also be computed by a depth four arithmetic circuit of size $2^{o(d+d\log\frac{n}{d})}$ with multiplication gates of fanin $o(d)$.

Notice that polynomial P can trivially be computed by an arithmetic circuit of size $2^{O(d+d\log\frac{n}{d})}$ (and depth two). The main result have implications to identity testing as well.

Identity Testing Chasm: If there is a complete black-box derandomization of Identity Testing for depth four circuits with multiplication gates of small fanin, then the general Identity Testing problem can be deterministically solved in $n^{O(\log n)}$ time.

We wish to point out that the Boolean setting behaves quite differently as compared to the arithmetic setting due to the existence of two additional axioms: (1) $fg + g = g$ and (2) $g^2 = g$. These axioms result in strong cancellative properties and they actually demonstrate that degree in a Boolean setting is not a "natural" primary resource in the sense that it can be traded for time. For example, insisting on polynomial size and polynomial degree circuits puts the languages in LOGCFL but if the constraint on degree is removed, one captures all of P. Intuitively, degree is not a very good resource measure as we may get small sized circuits for a language if we lift the constraint on its degree. In contrast, in arithmetic settings higher degree terms can never cancel lower degree ones and it follows that given a circuit to compute a degree d polynomial it can be replaced by another circuit whose all intermediate polynomials are of degree $\leq d$ at the cost of polyonmial increase in size [22].

This is not to mean that depth reduction results such as ours are not possible in Boolean settings. Indeed, Valiant[23] shows how monotone circuit of size n and depth $O(\log n)$ can be converted to a depth three monotone circuit of size $2^{O(n/\log\log n)}$.

2 The Chasm at Depth Four

It is easy to see that depth four circuits are more powerful than depth three. For example, consider the problem of computing determinant over a finite field F. We know, by [7], that depth three circuits computing determinant over F require exponential size. We now observe that determinant over F can be computed by depth four arithmetic circuits of size $2^{o(n)}$.

We will start with a problem well know to be computationally equivalent to the determinant: matrix powering [3]. Matrix powering is the problem of powering an $n \times n$ matrix to the

n^{th} power, where each entry of the matrix is either $-1, 0$, or 1.

The proof is simple. We break the matrix chain of n matrices into $\sqrt{n}$ equal sections. In each section, we can compute the ij^{th} entry of the resulting matrix as a sum of products; each product being a multiplication of $\sqrt{n}$ entries. It is easy to see that the number of such products, and hence the fan-in into the PLUS gate, is bounded by $n^{\sqrt{n}}$.

At the end of this phase, we are left with $\sqrt{n}$ matrices; one for each section. The ij^{th} entry of the resulting matrix, can similarly be written as sum of products. Again, a product would be $\sqrt{n}$ long and the sum would be over all possible $n^{\sqrt{n}}$ products.

Overall, this results in a depth 4 circuit of size $n^{O(\sqrt{n})}$ for matrix powering, and hence for the determinant.

Theorem 2.1 *The determinant of a $n \times n$ matrix with integer n bit entries can be computed by depth 4 arithmetic circuits of size $n^{O(\sqrt{n})}$.*

We now generalize the above observation to any arithmetic circuit of subexponential size. In this paper, we use subexponential size to mean circuits of size $2^{o(n)}$.

Let $P(x_1, \ldots, x_n)$ be a polynomial of total degree d. We restrict our attention to the case when $d = O(n)$[1]. P can be written as sum of at most $\binom{n+d}{d}$ products. Hence it can always be computed by a depth two circuit of size $2^{O(d+d\log\frac{n}{d})}$ because:

[1]If the degree is $\omega(n)$ then the bounds we get are weaker, and in any case the permanent has sublinear degree.

Lemma 2.2 *For any n and $k(n)$ such that $k(n) = O(n)$:*

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

Proof. If $k(n) = \Theta(n)$, then $\binom{n+k(n)}{k(n)} \leq 2^{n+k(n)} = 2^{O(k(n))}$. If $k(n) = o(n)$ then, using Stirling's formula for factorial:

$$\begin{aligned}
\binom{n+k(n)}{k(n)} &= \frac{(n+k(n))!}{n!k(n)!} \\
&= O\left(\frac{(n+k(n))^{n+k(n)}}{n^n k(n)^{k(n)}}\right) \\
&= O\left(\left(1+\frac{k(n)}{n}\right)^n \cdot \left(1+\frac{n}{k(n)}\right)^{k(n)}\right) \\
&= O\left(e^{k(n)} \cdot \left(2\frac{n}{k(n)}\right)^{k(n)}\right) \\
&= 2^{O(k(n)+k(n)\log\frac{n}{k(n)})}.
\end{aligned}$$

□

Let P be computed by arithmetic circuit C of size M with $M = 2^{o(d+d\log\frac{n}{d})}$. We can assume, without loss of generality, that all the intermediate polynomials computed inside the circuit C have degree bounded by d [22]. In [20, 2] it is shown that C can be transformed to a circuit D of degree d, size $M^{O(1)}$ and depth $O(\log d)$ with multiplication gates of fan-in two. We do a careful analysis of the transformation in [2] to obtain a circuit D with more structural properties. In particular, we will be interested in getting good bounds on the degree of the gates.

The circuit D we construct will be a strictly alternating circuit of size $S = M^{O(1)}$, where M is the size of the original circuit. The addition gates of D have unbounded fan-in, while multiplication gates of D have fan-in bounded by 6.

The degree of polynomials computed at each gate satisfies the following properties:

- the output gate degree is d,
- degree of any child of an addition gate is the same as the degree of the gate,
- all children of a multiplication gate have degree at most half of the degree of the gate.

It follows that the depth of the circuit D is at most $2 \log d$. We now indicate how to construct such a circuit.

Construction

As a first step, we ensure the C is a layered circuit with alternating levels of PLUS and MULT gates. Also, we will ensure the fan-in at every multiplication gate is 2. Finally, we rearrange the children of the multiplication gate so that the degree of the left child is smaller than or equal to the degree of the right child.

A *proof tree* rooted at gate g of circuit C is a subcircuit obtained as follows:

- start with the subcircuit in C that has gate g at the top and computes the polynomial associated with gate g,
- for every addition gate in this subcircuit, retain only one input to the gate while deleting the remaining input lines,
- for any multiplicate gate in the subcircuit, retain both the inputs to it.

It is easy to see that a proof tree rooted at g computes a monomial of the polynomial computed at g and this polynomial equals the sum of all such monomials.

For every gate g in C, define $[g]$ to stand for the polynomial computed at gate g. For every pair of gates g and h in C, let $[g, h] = \sum_T p(T, h)$ where T runs over all the proof trees rooted at g in which h occurs in the rightmost path of the tree, and $p(T, h)$ denotes the polynomial computed by the proof tree T *when gate h is replaced by constant* 1. If h does not occur in the rightmost path of any proof tree at g then $[g, h]$ is the zero polynomial. Gates of circuit D will be $[g]$, $[g, h]$, and $[x_i]$ for gates g and h of C and variables x_i ($[x_i]$ represent the variable x_i). The connections between these gates are described below.

It is easy to see that $[g] = \sum_{i=1}^{n} [g, x_i][x_i]$.

Also, if g is a PLUS gate with children $g_1, \ldots, g_k$, then $[g, h] = \sum_{i=1}^{k} [g_i, h]$.

Let g be a MULT gate with children g_L and g_R as left and right children respectively. Then, if the right most path from g to h has only PLUS gates then $[g, h] = [g_L]$. Otherwise, for a fixed right most path from g to h, there must exist a unique intermediate MULT gate, say p (with children p_L and p_R) along the right most path connecting g and h such that

$$deg(p_R) \leq \frac{1}{2}(deg(g) + deg(h)) \leq deg(p).$$

Of course, several right most paths could exist between g and h and we have no way of pinpointing only them. Therefore we sum over all possible gates p, satisfying the above condition. Then, $[g, h] = \sum_p [g, p][p_L][p_R, h]$.

Let us now analyze the three terms in the product. Clearly, $deg([g]) = deg(g)$ and $deg([g, h]) = deg(g) - deg(h)$.

1. $deg([g, p]) = deg(g) - deg(p) \leq \frac{1}{2}(deg(g) - deg(h))$.
2. $deg([p_L]) \leq deg(p) \leq \frac{1}{2} deg(g)$. Again, $deg(p_L) \leq deg(p_L) + deg(p_R) - deg(h) \leq deg(g) - deg(h)$.

3. $deg([p_R, h]) = deg(p_R) - deg(h) \leq \frac{1}{2}(deg(g) - deg(h))$.

We want all the children of $[g, h]$ to be at most half its degree, $deg([g, h])$. The problem is with the child $[p_L]$, whose degree need not be bounded above by $\frac{1}{2}deg([g, h])$. To get around this, we apply the depth reduction algorithm once more to $[p_L]$. We have $[p_L] = \sum_{i=1}^{n}[p_L, x_i][x_i]$ and p_L is a PLUS gate. Let $p_L = \sum_j p_L^j$ with each p_L^j being a MULT gate. Then, $[p_L, x_i] = \sum_j [p_L^j, x_i]$. Applying the above analysis on $[p_L^j, x_i]$, we get that $[p_L^j, x_i] = \sum_q [p_L^j, q][q_L][q_R, x_i]$ for certain gates q, q_L and q_R (q_L and q_R are children of q and degree of q satisfies the bounds as above). By our analysis, for the troublesome left child we now have $deg(q_L) \leq \frac{1}{2}p_L \leq \frac{1}{2}deg([g, h])$. Of course, the bound holds easily for the q and q_R as well and therefore, we have:

$$[g, h] = \sum_p \sum_{i=1}^{n} \sum_j \sum_q [g, p][p_L^j, q][q_L][q_R, x_i][x_i][p_R, h]$$

where p, q satisfy the appropriate degree constraints. This completes the description of circuit D.

By introducing dummy PLUS gates in the circuit, we can ensure that PLUS and MULT gates alternate in D. Thus we get a fan-in 6 multiplication circuit D with depth at most $2 \log d$ (of which at most $\log d$ layers are of MULT gates) and size $M^{O(1)}$. All the properties that we had listed of circuit D are satisfied. Let S be the size of the circuit D, $S = M^{O(1)}$.

We construct a depth 4 circuit E from D. Choose any ℓ such that $\ell \leq \frac{d + d \log \frac{n}{d}}{\log S}$ and $\ell = \omega(1)$. Set $t = \frac{1}{2}\log_6 \ell$. Cut the circuit D in two halves: the top one has exactly t MULT layers with the last layer being of MULT gates and the rest of layers belong to the bottom half. Let g_1, ..., g_k ($k \leq S$) be the output gates in the bottom layer. We can view the top layer as computing a polynomial in k new variables, say, y_1, ..., y_k. Let this polyonmial be $P_0(y_1, \ldots, y_k)$. Let the polynomial computed at the gate g_i be $P_i(x_1, \ldots, x_n)$ for $1 \leq i \leq k$. The polynomial computed by the circuit D equals

$$P_0(P_1(x_1, \ldots, x_n), \ldots, P_k(x_1, \ldots, x_n)).$$

We now obtain an upper bound on the degrees of all these polynomials. Since the top layer has exactly t MULT layers and each MULT gate has fanin bounded by 6, the degree of P_0 is bounded by 6^t. Since the degree goes down by at least a factor of two across MULT layers, the degree of P_i is bounded by $\frac{d}{2^t}$ for $1 \leq i \leq k$.

Express each P_i, $0 \leq i \leq k$ as a sum of products, thus each requiring a depth two circuit to compute. Patching together these circuits, we get a depth four circuit computing the polynomial computed by D. Let this circuit be E. Let us calculate the size of E.

Lemma 2.3 *Polynomial P_0 can be written as a sum of at most $\binom{S+6^t}{6^t}$ products, each of fanin $\leq 6^t$. Polynomials P_i, $1 \leq i \leq k$, can be written as a sum of at most $\binom{n+\frac{d}{2^t}}{\frac{d}{2^t}}$ products, each of fanin $\leq \frac{d}{2^t}$.*

Proof. The number of monomials on n variables and degree k is $\binom{n+k}{k}$. The lemma now follows from the degree bound on each polynomial and the number of variables they are defined on. □

Therefore, the size of circuit E is bounded by

$$\begin{aligned}
&\binom{S+6^t}{6^t} + S \cdot \binom{n+\frac{d}{2^t}}{\frac{d}{2^t}} \\
&= \binom{S+\sqrt{\ell}}{\sqrt{\ell}} + S \cdot \binom{n+\frac{d}{\ell^\delta}}{\frac{d}{\ell^\delta}} \quad (\delta = \tfrac{1}{2}\log_6 2) \\
&\leq (2S)^{\sqrt{\ell}} + S \cdot 2^{O(\frac{d}{\ell^\delta} + \frac{d}{\ell^\delta}(\log \frac{n}{d} + \log \ell^\delta))} \\
&\qquad (\text{since } \sqrt{\ell} \leq S \text{ and } \tfrac{d}{\ell^\delta} = o(n)) \\
&= 2^{O(\sqrt{(d+d\log\frac{n}{d})\log S})} + S \cdot 2^{o(d+d\log\frac{n}{d})} \\
&= 2^{o(d+d\log\frac{n}{d})}.
\end{aligned}$$

Therefore, we have the following theorem.

Theorem 2.4 *Let $P(x_1, \ldots, x_n)$ be a polynomial of degree $d = O(n)$ over the field F. If there exists an arithmetic circuit of size $2^{o(d+d\log\frac{n}{d})}$ for P then there exists a depth 4 arithmetic circuit of size $2^{o(d+d\log\frac{n}{d})}$ for P. Further, the fanin of second layer* MULT *gates is bounded by $\ell(n)$ where ℓ is any sufficiently slowly growing function in $\omega(1)$ and the fanin of bottom layer* MULT *gates is bounded by $o(d)$.*

For multilinear polynomials, we have the following corollary.

Corollary 2.5 *Let $P(x_1, \ldots, x_n)$ be a multilinear polynomial of over the field F. If there exists an arithmetic circuit of size $2^{o(n)}$ for P, then there exists a depth 4 arithmetic circuit of size $2^{o(n)}$ for P.*

When the multilinear polynomial is specialized to the permanent we get,

Corollary 2.6 *If every depth 4 arithmetic circuit for Permanent require exponential size, then every arithmetic circuit for Permanent requires exponential size.*

3 Black-box Derandomization of Identity Testing

An arithmetic circuit of size n is a *low degree circuit* if the polynomial computed by the circuit has degree $\leq n$. *Low degree Identity Testing* is the problem of testing if a given low degree circuit is zero. In this section, we relate the black-box derandomization of depth four Identity Testing to low degree Identity Testing. A black-box derandomization of low degree Identity Testing problem can be defined as follows (it is a restriction of the definition given in [1] to low degree circuits).

Definition 3.1 *Let F be a field. Let $\mathcal{C}$ be a class of low degree arithmetic circuits over F. Function $f : \mathbb{N} \mapsto (F[y])^*$ is a* s(n)-pseudorandom generator against class $\mathcal{C}$ of arithmetic circuits *if*

- *$f(n) = (p_1^n(y), p_2^n(y), \ldots, p_n^n(y)), p_j^n(y) \in F[y]$ is computable in time polynomial in $s(n)$ and each p_j^n is of degree bounded by $s(n)$.*
- *For any arithmetic circuit $C \in \mathcal{C}$ of size n computing a polynomial of n variables over F, $C(x_1, x_2, \ldots, x_n) = 0$ iff $C(p_1^n(y), p_2^n(y), \ldots, p_n^n(y)) = 0$.*

Given an $s(n)$-pseudorandom generator f against $\mathcal{C}$, one can solve the Identity Testing problem (for circuits from the class $\mathcal{C}$) deterministically in time $s^{O(1)}(n)$ by simply plugging in the polynomial p_j^n for x_j and evaluating the resulting (univariate) polynomial. A complete derandomization is obtained when $s(n)$ is a polynomial in n. We call such generators *optimal* pseudorandom generators.

Remark. The above definition of pseudorandom generators, at a first glace, may appear

different from the one in boolean settings. Borrowing from boolean settings, one can define a $s(n)$-*hitting set generator* against arithmetic circuits via function $g : \mathbb{N} \times \mathbb{N} \mapsto F^*$, $g(n,t) = (a_1^t, a_2^t, \ldots, a_n^t)$ such that for any circuit C of size n on n inputs, C computes a non-zero polyonmial iff there exists a t, $1 \leq t \leq s(n)$ such that $C(a_1^t, \ldots, a_n^t) \neq 0$. It is, however, straightforward to see that the two definitions are equivalent: let $p_i^n(y)$ be the polyonmial of degree $\leq s(n)$ such that $p_i^n(t) = a_i^t$ for all $1 \leq t \leq s(n)$. This gives a pseudorandom generator of our definition. Conversely, let f be a $s(n)$-pseudorandom generator of our definition. Define $g(n,t) = (p_1^n(t), \ldots, p_n^n(t))$ for $1 \leq t \leq 1 + ns(n)$. Then g is a $(1+ns(n))$-hitting set generator. (If for circuit C of degree n, $C(p_1^n(y), \ldots, p_n^n(y)) \neq 0$ then $C(g(n,t)) \neq 0$ for some $t \leq 1 + ns(n)$ since $C(p_1^n(y), \ldots, p_n^n(y))$ is a non-zero polynomial of degree $\leq n \cdot s(n)$.

Theorem 3.2 *Consider the class of depth* 4 *arithmetic circuits over F with fanin of second layer* MULT *gates bounded by $O(\ell(n))$ (for any unbounded function ℓ) and the fanin of bottom layer* MULT *gates bounded by $O(\log n)$. If there is an optimal pseudorandom generator against this class of circuits then the low degree Identity Testing problem over F can be solved deterministically in time $n^{O(\log n)}$.*

Proof. Let f be an optimal pseudorandom generator against the class of depth 4 circuits over F defined above. It was shown in [1] that such a pseudorandom generator yields a family of multilinear polynomials $\{q_m\}_{m \geq 1}$ such that q_m is over m variables, is computable in time $2^{O(m)}$, and requires depth 4 circuits of size $2^{\Omega(m)}$, with fanins of second and bottom layer MULT gates bounded by $O(\ell(2^m))$ and $O(m)$ respectively, to compute. By Theorem 2.4, polynomial q_m requires exponential sized circuits (of any depth) to compute. Now, we can construct an algorithm that derandomizes low degree Identity Testing over F in time $n^{O(\log n)}$ using the polynomial q as shown by the lemma below. □

Lemma 3.3 *Let $\{q_m\}_{m \geq 1}$ be a multilinear polynomial family over F computable in exponential time and that cannot be computed by subexponential sized arithmetic circuits. Then the low degree Identity Testing problem over F can be solved in time $n^{O(\log n)}$.*

Proof. The proof is along the lines of the proof of Lemma 7.6 in [9]. Let C be any circuit over F of size n computing a polynomial of degree $\leq n$. We wish to test if C compute the zero polynomial. Let S_1, S_2, ..., S_n be subsets of $[1, c \log n]$ (for a suitable constant c) such that $|S_i| = d \log n$ (for a suitable $d < c$) and $|S_i \cap S_j| \leq \log n$ (for $i \neq j$). This family of sets is the *Nisan-Wigderson design* [13] and can be efficiently constructed. For a tuple of variables $(x_1, x_2, \ldots, x_{c \log n})$, denote by $(x_1, x_2, \ldots, x_{c \log n})_{S_i}$ the tuple obtained by retaining only those variables whose indices occur in S_i (the variables are always arranged in increasing order of index). Without loss of generality, we can assume that C has n inputs $z_1, \ldots, z_n$. Replace z_i by $p_i = q_{d \log n}(x_1, x_2, \ldots, x_{c \log n})_{S_i}$ for each i. We now claim that if C is zero after substitution then it is zero without substitution as well.

Suppose not. So $C(z_1, \ldots, z_n) \neq 0$ and $C(p_1, \ldots, p_n) = 0$. Then there must exist an index j such that $C(p_1, \ldots, p_j, z_{j+1}, \ldots, z_n) = 0$ and $C(p_1, \ldots, p_{j-1}, z_j, \ldots, z_n) \neq 0$. Randomly

fix values of variables $z_{j+1}, \ldots, z_n$ as well as x_i's not occuring in the polynomial p_j in the last circuit. The circuit will still compute a non-zero polynomial with high probability. Fix value to the above variables that keep the circuit non-zero. Now replace each p_i, $i < j$, by a sum of product form. Since all but $\log n$ variables of p_i are fixed, the size of this form is bounded by n. After replacement, we get a circuit of size $\leq n^2$ over variables $(x_1, \ldots, x_{c \log n})_{S_j}$ and z_j that is non-zero but becomes zero on substituting z_j by p_j. Hence $z_j - p_j$ divides the polynomial computed by the new circuit. We now use the multivariate polynomial factorization algorithm [10] to compute this factor. The circuit computing the factor has size n^e for some constant e independent of d. This gives us a circuit of size $n^e + n^2$ that computes polynomial p_j which is $q_{d \log n}$. Choosing a suitable d yields a contradiction on the hardness of $q_{d \log n}$.

Therefore, if C was non-zero to start with, it continues to be non-zero even after the substitution. Now express C as sum of products using brute-force. Since C after substitution computes a degree $O(n \log n)$ polyonmial over $O(\log n)$ variables, it will have at most $n^{O(\log n)}$ terms. This gives an $n^{O(\log n)}$ time algorithm for testing if C is a zero. □

Theorem 3.2 is suboptimal. It is an interesting open question to improve it to obtain polynomial time algorithm instead of $n^{O(\log n)}$.

References

[1] M. Agrawal. Proving lower bounds via pesudo-random generators. In *Proceedings of the FST&TCS*, pages 96–105, 2005.

[2] E. Allender, J. Jiao, M. Mahajan, and V Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theoretical Computer Science*, 209:47–86, 1998.

[3] S. A. Cook. A Taxonomy of Problems with Fast Parallel Algorithms. *Information and Control* 64(1-3):2–21, 1985.

[4] C. Damm. DET=$L^{\#l}$. Technical Report Informatik-preprint 8, Fachbereich Informatik der Humboldt Universität zu Berlin, 1991.

[5] Z. Dvir, A. Shpilka, A. Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. In *Symposium of Theory of Computing* 741–748, 2008.

[6] D. Grigoriev and A. Razborov. An Exponential Lower Bound for Depth 3 Arithmetic Circuits. In *Symposium on Theory of Computing* 577–582, 1998.

[7] D. Grigoriev and A. Razborov. Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 10(6):467–487, 2000.

[8] M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982.

[9] Valentine Kabanets and Russell Impagliazzo. Derandomizing polyonmial identity tests means proving circuit lower bounds. *Computational Complexity*, 13:1–46, 2004.

[10] E. Kaltofen. Factorization of polynomials given by straight-line programs. In S. Micali, editor, *Randomness and Computation*, 375–412. JAI Press, 1989. Volume 5 of Advances in Computing Research.

[11] Z. S. Karnin, A. Shpilka. Black Box Polynomial Identity Testing of Generalized Depth-3 Arithmetic Circuits with Bounded Top Fan-In. In *IEEE Conference on Computational Complexity* 280–291, 2008.

[12] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.

[13] N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Sys. Sci.*, 49(2):149–167, 1994.

[14] N. Nisan and A. Wigderon. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6:217–234, 1996.

[15] Ran Raz. Multi-linear formulas for permanent and determinant and of super-polynomial size. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 633–641, 2004.

[16] Ran Raz, Amir Yehudayoff. Multilinear Formulas, Maximal-Partition Discrepancy and Mixed-Sources Extractors. ECCC TR07-085, 2007.

[17] Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, LNCS 5125, pages 60–71, 2008.

[18] A. Shpilka and A. Wigderson. Depth-3 arithmetic formulae over fields of characteristic zero. In *Proceedings of the Conference on Computational Complexity*, pages 79–96, 1999.

[19] S. Toda. Counting problems computationally equivalent to the determinant. manuscript, 1991.

[20] L. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynnomials using few processors. *SIAM Journal on Computing*, 12:641–644, 1983.

[21] L. G. Valiant. The Complexity of Computing the Permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[22] L. G. Valiant. Negation can be exponentially powerful. *Theor. Comput. Sci.*, 12:303–314, 1980.

[23] L. G. Valiant. Exponential Lower Bounds for Restricted Monotone Circuits. In *Symposium on Theory of Computing*, 110–117, 1983.

[24] V Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *IEEE Proceedings of the Structure in Complexity Theory Conference*, pages 270–284. 1991.