Innovations in Computer Science 2011

Shallow Circuits with High-Powered Inputs

Pascal Koiran
LIP*, École Normale Supérieure de Lyon, Université de Lyon
Department of Computer Science, University of Toronto†
pascal.koiran@gmail.com

Abstract: A polynomial identity testing algorithm must determine whether an input polynomial (given for instance by an arithmetic circuit) is identically equal to 0. In this paper, we show that a deterministic black-box identity testing algorithm for (high-degree) univariate polynomials would imply a lower bound on the arithmetic complexity of the permanent. The lower bounds that are known to follow from derandomization of (low-degree) multivariate identity testing are weaker.
To obtain a lower bound for the permanent it would be sufficient to derandomize identity testing for polynomials of a very specific norm: sums of products of sparse polynomials with sparse coefficients. This observation leads to new versions of the Shub-Smale τ-conjecture on integer roots of univariate polynomials. In particular, we show that a lower bound for the permanent would follow if one could give a polynomial upper bound on the number of real roots of sums of products of sparse polynomials (Descartes' rule of signs gives such a bound for sparse polynomials and products thereof). In fact the same lower bound would follow even if one could only prove a slightly superpolynomial upper bound on the number of real roots. This is a consequence of a new result on reduction to depth 4 for arithmetic circuits which we establish in a companion paper. We also show that an even weaker bound on the number of real roots would suffice to obtain a lower bound on the size of depth 4 circuits computing the permanent.
These results suggest the intriguing possibility that tools from real analysis might be brought to bear on a longstanding open problem: what is the arithmetic complexity of the permanent polynomial?

Keywords: algebraic complexity, arithmetic circuits, permanent, lower bounds, polynomial identity testing, sparse polynomials, Descarte's rule.

1 Introduction

A polynomial identity testing algorithm must determine whether an input polynomial (given for instance by an arithmetic circuit) is identically equal to 0. If randomization is allowed, this problem can be solved efficiently thanks to the well-known Schwarz-Zippel lemma. Following Kabanets and Impagliazzo [12], it has become increasingly clear in recent years that efficient deterministic algorithms for polynomial identity testing would imply strong lower bounds (the connection between arithmetic circuit lower bounds and derandomization of polynomial identity testing was foreshadowed in a 30 years old paper by Heintz and Schnorr [1]). This approach to lower bounds was advocated in particular by Agrawal [1].

In this paper we show that an efficient black-box deterministic identity testing algorithm for univariate polynomials of a very specific form (namely, sums of products of sparse polynomials with sparse coefficients) would imply that the permanent does not belong to VP^0. This is the class of polynomial families computable by constant-free arithmetic circuits of polynomial size and polynomially bounded formal degree. It plays roughly the same role for constant-free circuits as the class VP in Valiant's algebraic version of the P versus NP problem (in Valiant's original setting, arithmetic circuits can use arbitrary constants from the underlying field [9,28]).

Compared to [1,12], one originality of the present paper is to show that lower bound for multivariate polynomials such as the permanent would follow from *univariate* identity testing algorithms. Most of the recent work on identity testing (surveyed in [2,25]) has been focused on low-degree multivariate polynomials.[1] Nevertheless, we believe that the univariate approach

* UMR 5668 ENS Lyon, CNRS, UCBL, INRIA.

† A part of this work was done during a visit to the Fields Institute.

[1] Two exceptions are [7,17].

is worth exploring for at least two reasons.

First, it would lead to stronger lower bounds. Indeed, we show that black-box derandomization of identity testing implies a lower bound for the permanent, whereas [1,Section 6.2] would only yield lower bounds for polynomials with coefficients computable in PSPACE (this complexity class was independently defined in [20], where it is called VPSPACE; further results on this class and other space-bounded classes in Valiant's model can be found in [19,22,24]). The lower bound obtained from [12] would be even weaker, but could be obtained from a non-black-box identity testing algorithm.

A second, possibly even more important advantage of the univariate approach is that it leads to new (and hopefully more tractable) versions of Shub and Smale's τ-conjecture. According to the τ-conjecture, the number of integer roots of a univariate polynomial $f \in \mathbb{Z}[X]$ should be bounded by a polynomial function of its arithmetic circuit size (the inputs to the circuit are the constant 1, or the variable X). It was shown by Bürgisser [10] that the τ-conjecture implies a lower bound for the permanent. Our main "hardness from derandomization" result can be viewed as an improvement of Bürgisser's result. Indeed, it follows immediately from our result that to obtain a lower bound for the permanent, one just has to bound the number of integer roots for sums of products of sparse polynomials with sparse coefficients (rather than for arbitrary arithmetic circuits). Our strongest version of the τ-conjecture raises the intriguing possibility that tools from real analysis might be brought to bear on this problem (a bound on the number of real roots of a polynomial is a fortiori a bound on its number of integer roots). It is known that this approach cannot work for the original τ-conjecture because the number of real roots of a univariate polynomial can grow exponentially as a function of its arithmetic circuit size: Chebyshev polynomials provide such an example [27]. A similar example was provided earlier by Borodin and Cook [8] (but they did not provide an analysis of the size of constants used by the corresponding arithmetic circuit). We conjecture that this behavior is not possible for sums of products of sparse polynomials.

1.1 Main ideas

A hitting set $\mathcal{H}$ for a set $\mathcal{F}$ of polynomials is a (finite) set of points such that there exists for any non-identically zero polynomial $f \in \mathcal{F}$ at least one point $a \in \mathcal{H}$ such that $f(a) \neq 0$. Hitting sets are sometimes called "correct test sequences" [11]. It is well-known that deterministic constructions of hitting sets and black-box deterministic identity testing are two equivalent problems: any hitting set for $\mathcal{F}$ yields an obvious black-box identity testing algorithm (declare that $f \equiv 0$ iff f evaluates to 0 on all the points of $\mathcal{H}$); conversely, assuming that $\mathcal{F}$ contains the identically zero polynomial, the set of points queried by a black box algorithm on the input $f \equiv 0$ must be a hitting set for $\mathcal{F}$.

The connection between black-box identity testing and lower bounds is especially apparent for univariate polynomials [11]. Namely, let $\mathcal{H}$ be a hitting set for $\mathcal{F}$. The polynomial

$$P = \prod_{a \in \mathcal{H}} (X - a) \tag{1}$$

cannot belong to $\mathcal{F}$ since it is nonzero and vanishes on $\mathcal{H}$. The same remark applies to all nonzero multiples of P. If $\mathcal{F}$ is viewed as some kind of "complexity class", we have therefore obtained a lower bound against $\mathcal{F}$ by exhibiting a polynomial P which does not belong to $\mathcal{F}$.

In the low-degree multivariate setting the polynomial which plays the same role is not given by such a simple formula as (1). Its coefficients can be obtained by solving an exponential size system of linear equations. This can be done in PSPACE, explaining why the lower bound in [1] would be for polynomials with coefficients computable in PSPACE. By contrast one can show that the coefficients in exponential-size products such as (1) are in the counting hierarchy, a subclass of PSPACE. This is the reason why we can obtain a lower bound for a polynomial in VNP (namely, the permanent) rather than in VPSPACE as in [1, Section 6.2].

It remains to explain why we only have to derandomize identity testing for sums of products of sparse polynomials in order to obtain a lower bound. This class of polynomials comes into the picture thanks to the recent depth reduction theorem of Agrawal and Vinay [3]: any multilinear polynomial in n variables which has an arithmetic circuit of size $2^{o(n)}$ also has a depth-4 arithmetic circuit of size $2^{o(n)}$. Sums of products of sparse polynomials are very far from being multilinear (they are univariate polynomials of possibly very high degree). They are nonetheless connected to depth-4 circuits by a simple transformation: if we replace the input variables in a depth-4 circuit by powers of a single variable X, we obtain a SPS polynomial

$f(X)$.

At this point, we should stress that we do *not* claim that univariate arithmetic circuits can be efficiently converted into SPS polynomials (this would be a kind of high-degree analogue of Agrawal and Vinay's depth reduction theorem). On the contrary, we conjecture that such a transformation is in general impossible, and that Chebyshev polynomials provide a counterexample (because, as pointed out earlier, they have too many real roots). Nevertheless, to obtain our results we represent efficiently (in Theorem 6) certain exponential size products by sums of products of sparse polynomials. *This is possible only under the assumption that the permanent is easy.* This assumption (and the resulting representation) is of course very likely to be false, but there is no harm in making it since the ultimate goal is a proof by contradiction that the permanent is hard.

1.2 Organization of the paper

In the next section we present our model of computation for the permanent (constant-free arithmetic circuits) as well as the corresponding complexity classes. Then we recall some definitions and results about the counting hierarchy (as explained above, this class plays a crucial role in the derivation of a lower bound for the permanent). Finally, we present the result by Agrawal and Vinay on reduction to depth four for arithmetic circuits, as well as a new result along the same lines [15].

In Section 3 we define precisely the notion of sum of products of sparse polynomials with sparse coefficients, and explain the connection to depth-4 circuits.

In Section 4 we present the notion of algebraic number generator. This is basically just a sequence of efficiently computable polynomials in $\mathbb{Z}[X]$. We wish to use them to construct hitting sets, by taking the sets of all roots of the polynomials in an initial segment of this sequence. In Section 5 we prove our main result: if a polynomial-size initial segment provides a hitting set against sums of products of sparse polynomials with sparse coefficients, then the permanent is not in VP^0. In fact, using our new result on reduction to depth four [15] we can show that the same lower bound would follow even if the hitting sets are of slightly superpolynomial size.

In Section 6 we present three new versions of the τ-conjecture, including a "real τ-conjecture". A proof of any of these conjectures would yield a lower bound for the permanent. We show that a fairly weak version of the real τ-conjecture would suffice to obtain a lower bound on the size of depth 4 circuits computing the permanent. We conclude the paper with a few remarks on some tools that might be useful to attack these conjectures.

2 Preliminaries

2.1 Complexity of arithmetic computations

We recall that an arithmetic circuit contains addition, subtraction and multiplication gates. We usually assume that these gates have arity 2, except when dealing with constant-depth circuits as in e.g. Theorem 3. The input gates are labelled by variables or constants. A circuit where the only constants are from the set $\{0, -1, 1\}$ is said to be constant-free (in such a circuit one can even assume that -1 is the only constant, and that there are no subtraction gates). A constant-free circuit represents a polynomial in $\mathbb{Z}[X_1, \ldots, X_n]$, where $X_1, \ldots, X_n$ are the variables labelling the input gates.

In this paper we investigate the complexity of computing the permanent polynomial with constant-free arithmetic circuits. This model of computation was systematically studied by Malod [23]. In particular, he defined a class VP^0 of polynomial families that are "easy to compute" by constant-free arithmetic circuits. First we need to recall the notion of formal degree:

(i) The formal degree of an input gate is equal to 1.
(ii) The formal degree of an addition or subtraction gate is the maximum of the formal degrees of its two incoming gates, and the formal degree of a multiplication gate is the sum of these two formal degrees.

Finally, the formal degree of a circuit is equal to the formal degree of its output gate. This is obviously an upper bound on the degree of the polynomial computed by the circuit.

Definition 1 *A sequence (f_n) of polynomials belongs to VP^0 if there exists a polynomial $p(n)$ and a sequence (C_n) of constant-free arithmetic circuits such that C_n computes f_n and is of size (number of gates) and formal degree at most $p(n)$.*

The size constraint implies in particular that f_n depends on polynomially many variables. The constraint on the formal degree forbids the computation of polynomials of high degree such as e.g. X^{2^n}; it also forbids the computation of large constants such as 2^{2^n}.

A central question in the constant-free setting is whether the permanent family belongs to VP^0. A related question is whether $\tau(\mathrm{PER}_n)$, the constant-free arithmetic circuit of the $n \times n$ permanent, is polynomially bounded in n. Obviously, if $\mathrm{PER} \in \mathsf{VP}^0$ then $\tau(\mathrm{PER}_n)$ is polynomially bounded in n, but (as pointed out in e.g. [9]) it is not clear whether the converse holds true. In this paper we focus on the first question (see section [7] for further comments).

Another important complexity class in the constant-free setting is the class VNP^0 of easily definable families. It is obtained from VP^0 in the natural way:

Definition 2 *A sequence $(f_n(X_1, \ldots, X_{u(n)}))$ belongs to VNP^0 if there exists a sequence $(g_n(X_1, \ldots, X_{v(n)}))$ in VP^0 such that $f_n(X_1, \ldots, X_{u(n)})$ is equal to:*

$$\sum_{\overline{\epsilon} \in \{0,1\}^{v(n)-u(n)}} g_n(X_1, \ldots, X_{u(n)}, \overline{\epsilon}).$$

For instance, the permanent family is in VNP^0. If this family in fact belongs to VP^0 then the same is true of every VNP^0 family up to constant multiplicative factors. Indeed, we have the following result (Theorem 4.3 of [16]):

Theorem 1 *Assume that the permanent family is in VP^0. For every family (f_n) in VNP^0 there exists a polynomially bounded function $p(n)$ such that the family $(2^{p(n)} f_n)$ is in VP^0.*

The occurence of the factor $2^{p(n)}$ in this theorem is due to the fact that the completeness proof of the permanent uses the constant $1/2$. As in [16] one could avoid this factor by working with the Hamiltonian polynomial instead of the permanent.

The next lemma is Valiant's criterion. The present formulation is basically that of [16, Theorem 2.3] but this lemma essentially goes back to [28](see also [9, Proposition 2.20]).

Lemma 1(Valiant's criterion) *Suppose that $n \mapsto p(n)$ is a polynomially bounded function, and that $f : \mathbb{N} \times \mathbb{N} \to \mathbb{Z}$ is such that the map $1^n 0 j \mapsto f(j,n)$ is in the complexity class $\mathsf{GapP/poly}$. Let $f_n(X_1, \ldots, X_{p(n)})$ be the multilinear polynomial*

$$\sum_{j \in \{0,1\}^{p(n)}} f(j,n) X_1^{j_1} \cdots X_{p(n)}^{j_{p(n)}}, \qquad (2)$$

where j_k denotes the bit of j of weight 2^{k-1}. Then the polynomial family (f_n) is in VNP^0.

Note that we use a unary encoding for n but a binary encoding for j. We recall the definition of $\mathsf{GapP/poly}$ (and a few other boolean complexity classes) in Section 2.2. In this paper we only need to apply Valiant's criterion to boolean-valued functions ($f(j,n) \in \{0,1\}$ for all j and n) such that the map $1^n 0 j \mapsto f(j,n)$ is in $\mathsf{P/poly}$.

2.2 The counting hierarchy

A connection between the counting hierarchy and algebraic complexity theory was discovered in [4]. This connection was further explored in [10] and [18]. For instance, it was shown in [10] that the polynomials $\prod_{i=0}^{2^n}(X - i)$ have polynomial-size circuits if the the same is true for the permanent family.

We first recall the definition of the two counting classes $\sharp\mathsf{P}$ and GapP.

Definition 3 *The class $\sharp\mathsf{P}$ is the set of functions $f : \{0,1\}^* \to \mathbb{N}$ such that there exist a language $A \in \mathsf{P}$ and a polynomial $p(n)$ satisfying*

$$f(x) = \#\{y \in \{0,1\}^{p(|x|)} : (x,y) \in A\}.$$

A function $f : \{0,1\}^* \to \mathbb{Z}$ is in GapP if it is the difference of two $\sharp\mathsf{P}$ functions.

The counting hierarchy introduced in [29] is a class of languages rather than functions. It is defined via the majority operator $\mathbf{C}$ as follows.

Definition 4 *If K is a complexity class, the class $\mathbf{C}.K$ is the set of languages A such that there exist a language $B \in K$ and a polynomial $p(n)$ satisfying the following condition: $x \in A$ iff*

$$\#\{y \in \{0,1\}^{p(|x|)} : (x,y) \in B\} \geqslant 2^{p(|x|)-1}.$$

The i-th level $\mathsf{C_iP}$ of the counting hierarchy is defined recursively by $\mathsf{C_0P} = \mathsf{P}$ and $\mathsf{C_{i+1}P} = \mathbf{C}.\mathsf{C_iP}$. The counting hierarchy CH is the union of the levels $\mathsf{C_iP}$ for all $i \geqslant 0$.

The counting hierarchy contains all the polynomial hierarchy PH and is contained in PSPACE.

The arithmetic circuit classes defined in Section 2.1 are nonuniform. As a result, we will actually work with nonuniform versions of the counting classes defined above. We use the standard Karp-Lipton notation [13]:

Definition 5 *If K is a complexity class, the class K/poly is the set of languages A such that there exist a language $B \in K$, a polynomial $p(n)$ and a family $(a_n)_{n\geqslant 0}$ of words (the "advice") satisfying*

- *for all $n \geqslant 0$, $|a_n| \leqslant p(n)$;*
- *for all word x, $x \in A \iff (x, a(|x|)) \in B$.*

Note that the advice only depends on the size of x.

The next lemma [10, Lemmas 2.6 and 2.13] provides a first link between arithmetic complexity and the counting hierarchy.

Lemma 2 *If the permanent family is in VP^0 then $\mathsf{CH/poly} = \mathsf{P/poly}$.*

In particular, Lemma 2 was used to show that large sums and products are computable in the counting hierarchy [10, Theorem 3.10].

In the remainder of this section we summarize some relevant results from [18].

Definition 6 *Let (f_n) be a family of polynomials in $\mathbb{Z}[X]$ such that the degree of f_n and the bitsize of its coefficients are smaller than $2^{p(n)}$ for some polynomial p.*

The coefficient sequence of (f_n) is the (double) sequence of integers $a(n,\alpha)$ defined by the relation

$$f_n(x) = \sum_{\alpha=0}^{2^{p(n)}-1} a(n,\alpha)x^\alpha.$$

The coefficient sequence is said to be definable in $\mathsf{CH/poly}$ if the language $\mathrm{Bit}(a) = \{(1^n, \alpha, j, b);$ the j-th bit of $a(n,\alpha)$ is equal to $b\}$ is in $\mathsf{CH/poly}$.

Note that in the above definition of $\mathrm{Bit}(a)$, the input n is given in unary but α and j are in binary (this is the same convention as in [18]; by contrast, in [10] all inputs are in binary).

Definition 7 *Let (f_n) be a family of polynomials as in Definition 6. We say that this family can be evaluated in $\mathsf{CH/poly}$ if the language of all tuples $(1^n, i, j, b)$ such that $0 \leqslant i < 2^{p(n)}$ and the j-th bit of $f_n(i)$ is equal to b is in $\mathsf{CH/poly}$.*

The following result establishes a connection between these two definitions. It is stated (and proved) in the proof of the main theorem (Theorem 3.5) of [18].

Theorem 2 *Let (f_n) be a family of polynomials as in Definition 6. If (f_n) can be evaluated in $\mathsf{CH/poly}$ at integer points, the coefficient sequence of (f_n) is definable in $\mathsf{CH/poly}$.*

In [18] we actually prove a multivariate version of this result, but the univariate case will be sufficient for our purposes.

2.3 Sums of products of dense polynomials

Agrawal and Vinay have shown that polynomials of degree $d = O(m)$ in m variables which admit nontrivial arithmetic circuits also admit nontrivial arithmetic circuits of depth four [3]. Here, "nontrivial" means of size $2^{o(d+d\log\frac{m}{d})}$. The resulting depth 4 circuits are $\sum\prod\sum\prod$ arithmetic formulas: the output gate (at depth 4) and the gates at depth 2 are addition gates, and the other gates are multiplication gates. This theorem shows that for problems such as arithmetic circuit lower bounds or black-box derandomization of identity testing, the case of depth four circuits is in a certain sense the general case.

We will need to apply reduction to depth four to multilinear polynomials only. In this case their result (Corollary 2.5 in [3]) reads as follows:

Theorem 3 (Reduction to depth four) *A multilinear polynomial in m variables which has an arithmetic circuit of size $2^{o(m)}$ also has a depth 4 arithmetic circuit of size $2^{o(m)}$.*

But what if we start from arithmetic circuits of size smaller than $2^{o(m)}$ (for instance, of size polynomial in m) ? It is reasonable to expect that the size of the corresponding depth four circuits will be reduced accordingly, but such a result cannot be found in [3]. We can however prove the following result [15].

Theorem 4 *Let (f_n) be a VP^0 family of polynomials of degree $d_n = \deg(f_n)$. This family can be computed by a family (Γ_n) of depth four circuits with $n^{O(\log d_n)}$ addition gates and $n^{O(\sqrt{d_n}\log d_n)}$ multiplication gates. The family (f_n) can also be computed by a family (F_n) of depth four arithmetic formulas of size $n^{O(\sqrt{d_n}\log d_n)}$. The inputs to Γ_n and F_n are variables*

of f_n or relative integers of polynomial bit size; their multiplication gates are of fan-in $O(\sqrt{d_n})$.

For instance, if the permanent is in VP^0 it can be computed by depth four arithmetic formulas of size $n^{O(\sqrt{n}\log n)}$. Compared to [3], there are mainly two new elements in Theorem 4:

(i) The size bounds for the depth-four circuits (Γ_n) and (F_n).
(ii) The bit size bound for the inputs of these circuits.

Our main results rely on these two new elements. In particular, we use (i) to show that constructing hitting sets of slightly superpolynomial size will still imply that the permanent is not in VP^0. To the author's knowledge, an analysis of the size of constants created in the depth-reduction procedure of [3] has not been carried out yet.

We can formulate Theorem 4 in more traditional mathematical language.

Corollary 1 *Let (f_n) be a VP^0 family of polynomials of degree $d_n = \deg(f_n)$. Each f_n can be represented by an expression of the form $\sum_{i=1}^{k}\prod_{j=1}^{m} f_{ij}$ where $k = n^{O(\sqrt{d_n}\log d_n)}$ and $m = O(\sqrt{d_n})$. The f_{ij} are polynomials of degree $O(\sqrt{d_n})$ and their coefficients are relative integers of polynomial bit size. Moreover, the sum of the number of monomials in all the f_{ij} is $n^{O(\sqrt{d_n}\log d_n)}$, and there are only $n^{O(\log d_n)}$ distinct f_{ij}.*

Proof Sketch. Each multiplication gate at depth 1 in the depth four circuit of Theorem 4 computes a monomial. Each addition gate at depth 2 computes a f_{ij}. A multiplication gate at depth 3 computes an expression of the form $\prod_j f_{ij}$. The output gate computes the final sum.

In fact, several multiplication gates at depth 1 may contribute to the same monomial of a f_{ij} and the monomial will be obtained as the sum of the outputs of these multiplication gates.[2] Taking this sum preserves the polynomial size bound on coefficients since there are only $n^{O(\sqrt{d_n}\log d_n)}$ multiplication gates. □

[2]This is bound to happen since as a polynomial of degree $O(\sqrt{d_n})$ in $n^{O(1)}$ variables f_{ij} can have at most $n^{O(\sqrt{d_n})}$ monomials, but there are many more multiplication gates.

A polynomial is sparse if it has few monomials compared to the maximal number of monomials possible given its degree and number of variables (recall that for a polynomial in n variables of degree d, this number is $\binom{n+d}{d}$). There is no reason for the f_{ij} to be sparse in general (but they have few terms compared to the maximum possible for f_n). As explained in the next section, if we replace the variables of the f_{ij} by a quickly growing sequence of powers of a single variable X, we obtain truly sparse univariate polynomials.

3 Sums of products of sparse polynomials

A sums of products of sparse polynomials is an expression of the form $\sum_i \prod_j f_{ij}$ where each $f_{ij} \in \mathbb{Z}[X]$ is a sparse univariate polynomial. Here "sparse" means as usual that we only represent the nonzero monomials of each f_{ij}. As a result one can represent concisely polynomials of very high degree. We define the size of such an expression as the sum of the number of monomials in all the f_{ij}. Note that this measure of size does not take into account the size of the coefficients of the f_{ij}, or their degrees. These relevant parameters are taken into account in the following definition.

Definition 8 *We denote by $\mathrm{SPS}_{s,e}$ the set of all polynomials in $\mathbb{Z}[X]$ which can be represented by an expression of the form $\sum_i \prod_j f_{ij}$ so that:*

- *The size of the expression as defined above is at most s.*
- *Each coefficient of each f_{ij} can be written as the difference of two nonnegative integers with at most s nonzero digits in their binary representations.*
- *These coefficients are of absolute value at most 2^e, and the f_{ij} are of degree at most e.*

Remark 1 *The polynomials f_{ij} in this definition can be thought of as "sparse polynomial with sparse coefficients". The integer s serves as a sparsity parameter for the number of monomials as well as for the number of digits in their coefficients. A typical choice for these parameters is $s = 2^{o(n)}$ and $e = 2^{O(n)}$, where n represents an input size (see for instance Theorem 6 in Section 5).*

We will show in Section 5 that constructing polynomial size hitting sets for sums of products of sparse polynomials implies the lower bound $\mathrm{PER}\notin\mathsf{VP}^0$. Here "polynomial size" means polynomial in $s + \log e$. It

is quite natural to insist on a size bound which is polynomial in s and $\log e$: s is an arithmetic circuit size bound, and $\log e$ can also be interpreted as an arithmetic cost since each power x^α in an f_{ij} can be computed from x in $O(\log e)$ operations by repeated squaring. Likewise, we can write each coefficient of each f_{ij} as the difference of two nonnegative integers as in Definition 8, and each of the $\leqslant s$ powers of 2 occuring in a nonnegative integer can be computed from the constant 2 in $O(\log e)$ operations. Each coefficient can therefore be computed in $O(s \log e)$ operations. As a result, a polynomial in $\mathrm{SPS}_{s,e}$ can be evaluated from the constant 1 and the variable X in a number of arithmetic operations which is polynomial in $s + \log e$.

The size of a SPS polynomial as we have defined it is essentially the size of a depth three arithmetic circuit (or more precisely of a depth three arithmetic formula) computing the polynomial. In this depth three formula each input gate carries a monomial; each addition gate at level 1 computes a f_{ij}; each multiplication gate at level 2 computes a product of the form $\prod_j f_{ij}$; and the output gate at level 3 computes the final sum.

We can further refine this representation of SPS polynomials by arithmetic formulas. Namely, instead of viewing the monomial aX^β as an atomic object which is fed to an input gate, we can decompose it as a sum of terms of the form $\pm 2^\alpha X^\beta$; and each term can be further decomposed as a product of factors of the form $\pm 2^{2^i}$ and X^{2^j}. The resulting object is a depth four formula where each input gate carries an expression of the form $\pm 2^{2^i}$ or x^{2^j} (note the symmetry between variables and constants in this representation). This connection between depth four formulas and SPS polynomials plays a crucial role in our results. In particular, we will use the following result in Section 5.

Proposition 1 *Let $(f_n(\overline{x}, \overline{z}))$ be a VP^0 family of multilinear polynomials, with $\overline{x}$ and $\overline{z}$ two tuples of variables of length $c \cdot n$ each (for some constant c). We define a univariate polynomial $f'_n(x)$ from f_n by the following substitution: $f'_n(x)$ is equal to*

$$f_n(x^{2^0}, x^{2^1}, \ldots, x^{2^{c \cdot n - 1}}, 2^{2^0}, \ldots, 2^{2^{c \cdot n - 1}}). \qquad (3)$$

The polynomials f'_n belong to $\mathrm{SPS}_{s,e}$ where $s = n^{O(\sqrt{n} \log n)}$ and $e = 2^{O(n)}$.

More precisely, each f'_n can be represented by an expression of the form $\sum_{i=1}^{k} \prod_{j=1}^{m} f'_{ij}$ where $k = n^{O(\sqrt{n} \log n)}$ and $m = O(\sqrt{n})$. The f'_{ij} are polynomials of degree $2^{O(n)}$ and have at most $n^{O(\sqrt{n})}$ nonzero monomials. Each coefficient of a monomial can be written as the difference of two non-negative integers of bit size $2^{O(n)}$ with at most $n^{O(\sqrt{n})}$ nonzero digits. Moreover, the sum of the number of monomials in all the f'_{ij} is $n^{O(\sqrt{n} \log n)}$, and there are only $n^{O(\log n)}$ distinct f'_{ij}.

Proof Sketch. This is a fairly straightforward consequence of Corollary 1. In particular, we have at most $n^{O(\sqrt{n})}$ monomials in f'_{ij} because this is also an upper bound on the number of monomials in the corresponding polynomials f_{ij} of Corollary 1. The effect of multiplication by the powers of two in (3) is to shift the coefficients of the f_{ij} without increasing their bit size, and we need to add (and subtract) $n^{O(\sqrt{n})}$ shifted coefficients to obtain a coefficient of a f'_{ij}. □

4 Algebraic number generators

As explained in Section 1.2, we wish to construct hitting sets by taking the sets of all roots of the polynomials in an initial segment of an efficiently computable sequence of polynomials. The following definition makes the notion of "efficiently computable" precise (compare with the notion of *hitting set generator* in [1, Section 6.2]).

Definition 9 *An algebraic number generator is a sequence $(f_i)_{i \geqslant 1}$ of nonzero univariate polynomials $f_i(X) = \sum_\alpha a(\alpha, i) X^\alpha$ such that for some integer constant $c \geqslant 1$:*

1. *The exponents α range from 0 to i^c;*
2. *$a(\alpha, i)$ is a sequence of integers of absolute value $\leqslant 2^{i^c}$;*
3. *The language $L(f)$ of all tuples (α, i, j, b) such that the j-th bit of $a(\alpha, i)$ is equal to b is in $\mathsf{CH/poly}$.*

In the above definition we work with the complexity class $\mathsf{CH/poly}$ because this is the largest complexity class for which our proofs go through. As shown in the next example, the language $L(f)$ can often be located in a much smaller complexity class.

Example 1 *Each of the three sequences $f_i = x - i$, $(x^i - 1)$ or $x^i - 2^i x + i^2 + 1$ is an algebraic number generator. Notice that in these three examples we can compute the coefficients of the f_i in polynomial time rather than in $\mathsf{CH/poly}$, i.e., there is no need for counting and the construction of the f_i is uniform.*

Theorem 5 *Let (f_i) be an algebraic number generator. From this sequence we define a family of univari-*

ate polynomials g_n by the formula:

$$g_n(x) = \prod_{i=1}^{2^n} f_i(x).$$

The coefficient sequence $b(n,\alpha)$ of g_n, defined by $g_n(x) = \sum_\alpha b(n,\alpha)x^\alpha$, is definable in $\mathsf{CH/poly}$.

Proof. The family (g_n) can be evaluated in $\mathsf{CH/poly}$ at integer points. This follows from the fact that integer sequences definable in $\mathsf{CH/poly}$ are stable under products and summations [10, Theorem 3.10]. The result then follows from Theorem 2. □

We illustrate this result on two examples.

Example 2 *For $f_i = x - i$ we have $g_n(x) = \prod_{i=1}^{2^n}(x-i)$. This is the Pochhammer-Wilkinson polynomial of order 2^n. It was shown in [10, proof of Main Theorem 1.2] that the coefficient sequence of Pochhammer-Wilkinson polynomials is definable in* CH.

Example 3 *For $f_i = x^i - 1$ we have $g_n(x) = \prod_{i=1}^{2^n}(x^i - 1)$. This product can be written as*

$$g_n(x) = \prod_{\bar\epsilon} h_n(x, \bar\epsilon) \tag{4}$$

where the auxiliary family h_n is defined by:

$$h_n(x, \epsilon_1, \ldots, \epsilon_n) = x \prod_{j=1}^{n} [(1-\epsilon_j) + \epsilon_j x^{2^{j-1}}] - 1.$$

Note that the powers $x^{2^{j-1}}$ in the above formula can be computed efficiently by repeated squaring. The family (h_n) therefore belongs to the class VP^0_{nb} of polynomials that can be evaluated in a polynomial number of arithmetic operations in the constant-free unbounded-degree model. It then follows from (4) that g_n belongs to the class $\mathsf{V\Pi P}^0$ (by definition, the families of this class are obtained as in (4) from a VP^0_{nb} family by taking an exponential-size product over a VP^0_{nb} family). It is shown in [18, Theorem 3.7] that the class $\mathsf{V\Pi P}^0$ would collapse to VP^0_{nb} if VNP^0 collapses to VP^0. The proof of this theorem is based on definability of coefficients in $\mathsf{CH/poly}$ for $\mathsf{V\Pi P}^0$ families (in our particular example there is again no need for nonuniformity since the family (f_i) is uniform).

5 From a hitting set to a lower bound

In this section we prove our main result: constructing hitting sets for the class $\mathrm{SPS}_{s,e}$ of sums of products of sparse polynomials with sparse coefficients implies a lower bound for the permanent (recall that the class $\mathrm{SPS}_{s,e}$ is defined in Section 3).

We begin with a lemma showing that under the assumption $\mathrm{PER} \in \mathsf{VP}^0$, polynomials with coefficients definable in $\mathsf{CH/poly}$ can be efficiently represented by sums of products of sparse polynomials. This result is an adaptation of [18, Lemma 3.2], which was itself a scaled up version of [10, Theorem 4.1(2)]. The main new ingredient is reduction to depth four as presented in Section 2.3.

Lemma 3 *Let $g_n(x) = \sum_\alpha a(n,\alpha)x^\alpha$ where the integers α range from 0 to $2^{c\cdot n}-1$, $a(n,\alpha)$ is a sequence of integers of absolute value $< 2^{2^{c\cdot n}}$ definable in* $\mathsf{CH/poly}$, *and c is an integer constant (independent of n).*

If $\mathrm{PER} \in \mathsf{VP}^0$ there is a polynomially bounded function $p(n)$ such that $2^{p(n)}g_n \in \mathrm{SPS}_{s,e}$ where $s = n^{O(\sqrt{n}\log n)}$ and $e = 2^{O(n)}$.

Proof. Expand a in binary:

$$a(n,\alpha) = \sum_{i=0}^{2^{c\cdot n}-1} a_i(n,\alpha)2^i.$$

Let h_n be the following multilinear polynomial:

$$h_n(x_1, x_2, \ldots, x_{c\cdot n}, z_1, \ldots, z_{c\cdot n}) = \sum_{i=0}^{2^{c\cdot n}-1}\sum_{\alpha=0}^{2^{c\cdot n}-1} a_i(n,\alpha)\bar z^{\bar i}\bar x^{\bar\alpha}.$$

In this formula, $\bar z^{\bar i}\bar x^{\bar\alpha}$ denotes the monomial $z_1^{i_1}\cdots z_{c\cdot n}^{i_{c\cdot n}} x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_{c\cdot n}^{\alpha_{c\cdot n}}$ and the exponents i_j and α_j denote the binary digits of the integers i and α. The univariate polynomial $g_n(x)$ is then equal to:

$$h_n(x^{2^0}, x^{2^1}, \ldots, x^{2^{c\cdot n-1}}, 2^{2^0}, \ldots, 2^{2^{c\cdot n-1}}). \tag{5}$$

Assume that the permanent family is in VP^0. by Lemma 2 the nonuniform counting hierarchy collapses, therefore computing the i-th bit $a_i(n,\alpha)$ of $a(n,\alpha)$ on input $(1^n,\alpha,i)$ is in $\mathsf{GapP/poly}$ (and even in $\mathsf{P/poly}$). By Lemma 1, $(h_n) \in \mathsf{VNP}^0$. By Theorem 1 there exists a polynomially bounded function $p(n)$ such that the family $f_n = 2^{p(n)}h_n$ is in VP^0. Applying Proposition 1 to (f_n) shows that the polynomials $f'_n = 2^{p(n)}g_n$ are in $\mathrm{SPS}_{s,e}$ for $s = n^{O(\sqrt{n}\log n)}$ and $e = 2^{O(n)}$. □

Next we we show that the product of the first 2^n polynomials of an algebraic number generator can be represented by a sum of products of sparse polynomials of subexponential size, assuming again that the permanent is in VP^0.

Theorem 6 *Let (f_i) be an algebraic number generator and $g_n(x) = \prod_{i=1}^{2^n} f_i(x)$. If* PER $\in \mathsf{VP}^0$ *there is a polynomially bounded function $p(n)$ such that $2^{p(n)} g_n \in \mathrm{SPS}_{s,e}$ where $s = n^{O(\sqrt{n}\log n)}$ and $e = 2^{O(n)}$. Here $\mathrm{SPS}_{s,e}$ is the class of sums of products of sparse polynomials from Definition 8.*

Proof. We wish to apply Lemma 3 to the polynomial $g_n(x) = \prod_{i=1}^{2^n} f_i(X)$. Each polynomial f_i in this product is of degree less than 2^{cn} (except possibly f_n, which may be of degree up to 2^{cn}). Hence g_n is of degree less than $2^{(c+1)n}$. As to the coefficient size, we have $||g_n||_1 \leqslant \prod_i ||f_i||_1$ where $||.||_1$ denotes the sum of the absolute values of the coefficients of a polynomial. For each i we have $||f_i||_1 < (2^{cn}+1)\cdot 2^{2^{cn}} \leqslant 2^{2^{(c+1)n}}$ so that $||g_n||_1 \leqslant 2^{2^{(c+2)n}}$. Finally, definability of coefficients in $\mathsf{CH/poly}$ is provided by Theorem 5. □

We can finally prove our main result.

Theorem 7 (Lower Bound from Hitting Sets) *Let (f_i) be an algebraic number generator and H_m the set of all roots of the polynomials f_i for all $i \leqslant m$. Let q and r be two functions such that $H_{q(s)+r(e)}$ is a hitting set for $\mathrm{SPS}_{s,e}$. The permanent is not in VP^0 if $r(e) = e^{o(1)}$ and q satisfies the following condition: for some constant $c < 1$ and s large enough, $q(s) \leqslant 2^{(\log s)^{1+c}}$.*

The conditions on q and r cover in particular the case of hitting sets of size polynomial in s and $\log e$. This special case was treated in an earlier version of this paper.[3] Note also that any set of more than $s \cdot e$ complex numbers is a hitting set since any polynomial in $\mathrm{SPS}_{s,e}$ is of degree at most $s \cdot e$.

Proof of Theorem 7 Let $g_n(x) = \prod_{i=1}^{2^n} f_i(x)$ be the polynomial of Theorem 6. Assume by contradiction that:

(i) There exists functions q and r such that $H_{q(s)+r(e)}$ is a hitting set for $\mathrm{SPS}_{s,e}$, where q and r satisfy the conditions in the statement of the theorem.
(ii) The permanent family is in VP^0.

From our second assumption and Theorem 6 we know that $2^{p(n)} g_n$ is in $\mathrm{SPS}_{s,e}$ for $s = n^{O(\sqrt{n}\log n)}$, $e = 2^{O(n)}$ and some polynomially bounded function $p(n)$. The conditions on q and r imply that for these values of s and e we have $q(s) + r(e) = 2^{o(n)}$. Hence by (i), for n large enough H_{2^n} is a hitting set for g_n. This is a contradiction since g_n vanishes on the hitting set H_{2^n} but is not identically 0. □

[3] http://arxiv.org/abs/1004.4960v2

6 Hitting sets from real analysis?

In this section we present our new versions of the τ-conjecture. Each of the three conjectures implies that the permanent is not in VP^0.

Conjecture 1(τ-conjecture for SPS polynomials) *There is a polynomial p such that any nonzero polynomial in $\mathrm{SPS}_{s,e}$ has at most $p(s + \log e)$ integer roots.*

This conjecture implies that PER$\not\in\mathsf{VP}^0$ (apply Theorem 7 to the algebraic number generator $f_i(x) = x - i$). Conjecture 1 follows from the τ-conjecture of Shub and Smale on integer roots of polynomials [26,27] since, as explained after Definition 8, polynomials in $\mathrm{SPS}_{s,e}$ can be evaluated by constant-free arithmetic circuits of size polynomial in s and $\log e$. It was already shown in [10] that the τ-conjecture implies a lower bound for the permanent. The point of Conjecture 1 is that to obtain such a lower bound we no longer have to bound the number of integer roots of arbitrary arithmetic circuits: we need only do this for sums of products of sparse polynomials. This looks like a much more manageable class of circuits, but the question is of course still wide open. Another related benefit of SPS polynomials in this context is that techniques from real analysis might become applicable. Before explaining this in more detail we formulate a somewhat stronger conjecture. The idea is that the parameter e in Conjecture 1 as well as the sparsity hypothesis on the integer coefficients might be irrelevant. This leads to:

Conjecture 2 (strong form of τ-conjecture for SPS polynomials)
Consider a nonzero polynomial of the form

$$f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X),$$

where each $f_{ij} \in \mathbb{Z}[X]$ has at most t monomials. The number of integer roots of f is bounded by a polynomial function of kmt.

Note that the size of f as defined in Section 3 is bounded by kmt. Therefore, Conjecture 2 is indeed stronger than Conjecture 1. Finally, we formulate an even stronger conjecture.

Conjecture 3(real τ-conjecture)
Consider a nonzero polynomial of the form

$$f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X), \tag{6}$$

where each $f_{ij} \in \mathbb{R}[X]$ has at most t monomials. The number of real roots of f is bounded by a polynomial function of kmt.

One could also formulate a weak version of the real τ-conjecture where the parameters s and e would play the same role as in Conjecture 1. Also, instead of a bound on the number of real roots which is polynomial in kmt one could seek a bound $q(kmt)$ which is slightly superpolynomial in the sense of Theorem 7: for some constant $c < 1$ and s large enough, we have $q(s) \leqslant 2^{(\log s)^{1+c}}$. By Theorem 7, such a bound would still be strong enough to conclude that the permanent is not in VP^0 (consider again the algebraic number generator $f_i(x) = x - i$, and the function $r(e) = 1$). This goal might still be difficult to achieve, so it would be of great interest to establish upper bounds on the number of real roots that are even weaker but still strong enough to imply interesting lower bounds. For instance:

Proposition 2 *Assume that for nonzero polynomials of the form (6) the number of real roots is less than $q(kmt)$, where the function q satisfies the condition $q(s) = 2^{s^{o(1)}}$. Then the permanent is not computable by polynomial size depth 4 circuits using polynomial size integer constants.*

Proof. Assume that the permanent is computable by polynomial size depth 4 circuits using polynomial size integer constants. In particular, the permanent is in VP^0. By completeness of the permanent we have the following strengthening of Theorem 1: for every family (h_n) in VNP^0 there exists a polynomially bounded function $p(n)$ such that the family $(2^{p(n)}h_n)$ is computable by polynomial size depth 4 circuits (using polynomial size integer constants).

We consider again the algebraic number generator $f_i(x) = x - i$ and the polynomial $g_n(x) = \prod_{i=1}^{2^n} f_i(x)$ of Theorem 6. We claim that g_n can be expressed as a SPS polynomial of size polynomial in n. This yields a contradiction since the assumption in the statement of the Proposition implies that g_n has $2^{n^{o(1)}}$ real roots but in reality g_n has 2^n integer roots.

The proof of the claim is similar to the proof of Lemma 3 and Theorem 6. In particular, we have for g_n the same representation as in formula (5) of Lemma 3. But now, the above-mentioned strengthening of Theorem 1 shows that h_n is computable by a depth 4 circuit of polynomial size. We obtain the SPS polynomial for g_n by plugging powers of x into this circuit.

□

At present there isn't a lot of evidence for or against Conjecture 3. We do know that the conjecture holds true when $k = 1$: by Descartes' rule each polynomial f_{1j} has at most $2t - 2$ nonzero real roots, so f has at most $2m(t-1)+1$ real roots. Also some indirect evidence is provided by the few known examples of polynomials with short arithmetic circuits but many real roots [8,27]: these examples are definitely not given as sums of products of sparse polynomials. The case $k = 2$ already looks nontrivial. In the general case we can expand f as a sum of at most kt^m monomials, so we have at most $2kt^m - 1$ real roots. A refutation of the conjecture would be interesting from the point of view of real algebra and geometry as it would yield examples of "sparse like" polynomials with many real roots. Of course, a proof of the conjecture would be even more interesting as it would yield a lower bound for the permanent.

7 Final Remarks

We have shown that constructing hitting sets for sums of products of sparse polynomials with sparse coefficients will show that $\mathrm{PER} \notin \mathsf{VP}^0$. It should be possible to obtain a variation of this result where the conclusion is that $\tau(\mathrm{PER}_n)$, the constant-free arithmetic circuit complexity of the permanent, is not polynomial in n. To obtain this stronger conclusion, a stronger hypothesis should be necessary. It seems natural to expect that the role payed by sparse polynomials with sparse coefficients will now played by sparse polynomials with coefficients of "small" τ-complexity (this is a larger class of polynomials since sparse coefficients are certainly of small τ-complexity).

Most importantly, one should try to prove or disprove the real τ-conjecture. A solution in the case $k = 2$ (a sum of two products of sparse polynomials) would already be quite interesting. We note that the search for good upper bounds on the number of solutions of sparse multivariate systems is a topic of current interest in real algebraic geometry. The theory of fewnomials [14] provides finiteness results and sometimes quantitative estimates on the number of real roots in very general "sparse like" situations. The

general estimates from [14], at least when applied in a straightforward manner, do not seem strong enough to imply the real τ-conjecture. Nevertheless, one can hope that the methods developed in [14] as well as in more recent work such as [5,6,21] will turn out to be useful.

References

[1] M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proc. FSTTCS 2005*. Invited paper.

[2] M. Agrawal and R. Saptharishi. Classifying Polynomials and Identity Testing. *Current Trends in Science*, 2009.

[3] M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proc. 49th IEEE Symposium on Foundations of Computer Science*, pages 67–75, 2008.

[4] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. Bro-Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009. Conference version in CCC 2006.

[5] B. Bertrand, F. Bihan, and F. Sottile. Polynomial systems with few real zeroes. *Mathematische Zeitschrift*, 253(2):361–385, 2006.

[6] F. Bihan and F. Sottile. New fewnomial upper bounds from Gale dual polynomial systems. *Moscow Mathematical Journal*, 7(3), 2007.

[7] M. Bläser, M. Hardt, R. J. Lipton, and N. K. Vishnoi. Deterministically testing sparse polynomial identities of unbounded degree. *Information Processing Letters*, 109(3):187–192, 2009.

[8] A. Borodin and S. Cook. On the number additions to compute specific polynomials. *SIAM Journal on Computing*, 5(1):146–157, 1976.

[9] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Number 7 in Algorithms and Computation in Mathematics. Springer, 2000.

[10] P. Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18:81–103, 2009. Conference version in STACS 2007.

[11] J. Heintz and C.-P. Schnorr. Testing polynomials which are easy to compute. In *Logic and Algorithmic (an International Symposium held in honour of Ernst Specker)*, pages 237–254. Monographie n^o 30 de L'Enseignement Mathématique, 1982. Preliminary version in *Proc. 12th ACM Symposium on Theory of Computing*, pages 262-272, 1980.

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

[13] R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28:191–209, 1982.

[14] A. G. Khovanskii. *Fewnomials*, volume 88 of *Translations of Mathematical Monographs*. American Mathematical Society, 1991.

[15] P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. http://arxiv. org/ abs/ 1006.4700.

[16] P. Koiran. Valiant's model and the cost of computing integers. *Computational Complexity*, 13:131–146, 2004.

[17] P. Koiran. A hitting set construction, with application to arithmetic circuit lower bounds. http://arxiv.org/abs/0907.5575, 2009.

[18] P. Koiran and S. Perifel. Interpolation in Valiant's theory, 2007. To appear in *Computational Complexity*. Available from http://perso.ens-lyon. fr/ pascal. koiran /publications.html.

[19] P. Koiran and S. Perifel. VPSACE and a transfer theorem over the complex field. In *Proc. 32nd International Symposium on Mathematical Foundations of Computer Science*, volume 4162 of *Lecture Notes in Computer Science*, pages 359–370, 2007.

[20] P. Koiran and S. Perifel. VPSACE and a transfer theorem over the reals. *Computational Complexity*, 18:551–575, 2009. Conference version in STACS 2007.

[21] T.-Y. Li, J. Maurice Rojas, and X. Wang. Counting real connected components of trinomial curve intersections and m-nomial hypersurfaces. *Discrete and Computational Geometry*, 30(3):379–414, 2003.

[22] M. Mahajan and R. Rao. Small-space analogues of Valiant's classes. In *Proc. 17th International Symposium on Fundamentals of Computation Theory*, pages 250–261. Springer, 2009.

[23] G. Malod. *Polynômes et coefficients*. PhD thesis, Université Claude Bernard - Lyon 1, 2003.

[24] B. Poizat. A la recherche de la définition de la complexité d'espace pour le calcul des polynômes à la manière de Valiant. *Journal of Symbolic Logic*, 73(4):1179–1201, 2008.

[25] N. Saxena. Progress on Polynomial Identity Testing. *Bull. EATCS*, 99:49–79, 2009.

[26] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP". *Duke Mathematical Journal*, 81(1):47–54, 1995.

[27] S. Smale. Mathematical problems for the next century. *Mathematical Intelligencer*, 20(2):7–15, 1998.

[28] L. G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM Symposium on Theory of Computing*, pages 249–261, 1979.

[29] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Inf.*, 23(3):325–356, 1986.