The Limited Power of Powering: Polynomial Identity Testing and a Depth-four Lower Bound for the Permanent

Bruno Grenet[13], Pascal Koiran[13], Natacha Portier[13⋆], Yann Strozecki[23]

[1] LIP, UMR 5668, ÉNS de Lyon – CNRS – UCBL – INRIA
École Normale Supérieure de Lyon, Université de Lyon
[Bruno.Grenet,Pascal.Koiran,Natacha.Portier]@ens-lyon.fr
[2] Équipe de Logique Mathématique, Université Paris VII
Strozecki@logique.jussieu.fr
[3] Department of Computer Science, University of Toronto

Abstract. Polynomial identity testing and arithmetic circuit lower bounds are two central questions in algebraic complexity theory. It is an intriguing fact that these questions are actually related. One of the authors of the present paper has recently proposed a "real τ-conjecture" which is inspired by this connection. The real τ-conjecture states that the number of real roots of a sum of products of sparse univariate polynomials should be polynomially bounded. It implies a superpolynomial lower bound on the size of arithmetic circuits computing the permanent polynomial.
In this paper we show that the real-τ conjecture holds true for a restricted class of sums of products of sparse polynomials. This result yields lower bounds for a restricted class of depth-4 circuits: we show that polynomial size circuits from this class cannot compute the permanent, and we also give a deterministic polynomial identity testing algorithm for the same class of circuits.

1 Introduction

The τ-conjecture [15,16] states that a univariate polynomial with integer coefficients defined by an arithmetic circuit has a number of integer roots polynomial in the size of the circuit. A real version of this conjecture was recently presented in [11]. The real τ-conjecture states that the number of real roots of a sum of products of sparse univariate polynomials should be polynomially bounded as a function of the size of the corresponding expression. More precisely, consider a polynomial of the form

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

where $f_{ij} \in \mathbb{R}[X]$ has at most t monomials. The conjecture asserts that the number of real roots of f is bounded by a polynomial function of kmt. It was shown in [11] that this conjecture implies a superpolynomial lower bound on the arithmetic circuit complexity of the permanent polynomial (a central goal of algebraic complexity theory ever since

⋆ This material is based on work supported in part by the European Community under contract PIOF-GA-2009-236197 of the 7th PCRD.

Valiant's seminal work [17]). In this paper we show that the conjecture holds true in a special case. We focus on the case where the number of distinct sparse polynomials is small (but each polynomial may be repeated many times). We therefore consider expressions of the form

$$\sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}}(X). \tag{1}$$

We obtain a $O(t^{m(2^{k-1}-1)})$ upper bound on the number of real roots of such a polynomial, where t is the maximum number of monomials in the f_j. In particular, the bound is polynomial in t when the "top fan-in" k and the number m of sparse polynomials in the expression are both constant. Note also that the bound is independent of the magnitude of the integers α_{ij}.

From this upper bound we obtain a lower bound on the complexity of the permanent for a restricted class of arithmetic circuits. The circuits that we consider are again of form (1), but now X should be interpreted as the tuple of inputs to the circuit rather than as a single real variable. Roughly speaking, we show a superpolynomial lower bound on the complexity of the permanent in the case where k and m are again fixed. Note that this is a lower bound for a restricted class of depth-4 circuits: the output gate at depth 4 has fan-in bounded by the constant k, and the gates at depth 2 are only allowed to compute a constant (m) number of distinct polynomials f_j.

Our third main result is a deterministic identity testing algorithm, again for polynomials of the same form. When k and m are fixed, we can test if the polynomial in (1) is identically equal to 0 in time polynomial in t and in $\max_{ij} \alpha_{ij}$. Note that if k, m *and the exponents* α_{ij} are all bounded by a constant then the number of monomials in such a polynomial is $t^{O(1)}$ and our three main results become trivial. These results are therefore interesting only in the case where the α_{ij} may be large, and can be interpreted as limits on the power of powering.

1.1 Connection to Previous Work

The idea of deriving lower bounds on arithmetic circuit complexity from upper bounds on the number of real roots goes back at least to a 1976 paper by Borodin and Cook [5]. Their results were independently improved by Grigoriev and Risler (see [7], chapter 12). For a long time, it seemed that the lower bounds that can be obtained by this method had to be rather small since the number of real roots of a polynomial can be exponential in its arithmetic circuit size. Nevertheless, as explained above it was recently shown in [11] that superpolynomial lower bounds on the complexity of the permanent on general arithmetic circuits can be derived from a suitable upper bound on the number of roots of sums of products of sparse polynomials. This is related to the fact that for low degree polynomials, arithmetic circuits of depth 4 are almost equivalent to general arithmetic circuits [2,10].

The study of polynomial identity testing (PIT) also has a long history. The Schwartz-Zippel lemma [14] yields a randomized algorithm for PIT. A connection between deterministic PIT and arithmetic circuit lower bounds was pointed out as early as 1980 by Heintz and Schnorr [8], but a more in-depth study of this connection began only much later [9]. The recent literature contains deterministic PIT algorithms for various restricted models (see e.g. the two surveys [1,13]). One model which is similar to ours was recently studied in [4]. It follows from Theorem 1 in [4] that there is a polynomial time deterministic black-box PIT algorithm for polynomials of the form (1) if, instead of bounding k and m as in our algorithm, we bound the transcendence degree r of the polynomials f_j. Obviously we have $r \leq m$, so from this point of view their result is more general.[†] On the other hand their running time is polynomial in the degree of the f_j, whereas we can handle polynomials of exponential degree in polynomial time. Note also that [4] does not provide any lower bound result.

1.2 Our approach

The proof of our bound on the number of real roots has the same high-level structure as that of Descartes' rule of signs.

Proposition 1. *A univariate polynomial $f \in \mathbb{R}[X]$ with $t \geq 1$ monomials has at most $t-1$ positive real roots.*

The number of negative roots of f is also bounded by $t-1$ (consider $f(-X)$), hence there are at most $2t-1$ real roots (including 0). There is also a refined version of Proposition 1 where the number of monomials t is replaced by the number of sign changes in the sequence of coefficients of f. The cruder version will be sufficient for our purposes.

We briefly recall an inductive proof of Proposition 1. For $t = 1$, there is no non-zero root. For $t > 1$, let $a_\alpha X^\alpha$ be the monomial of lowest degree. We can assume that $\alpha = 0$ (if not, we can divide f by X^α since this operation does not change the number of positive roots). Consider now the derivative f'. It has $t-1$ monomials, and at most $t-2$ positive real roots by induction hypothesis. Moreover, by Rolle's theorem there is a positive root of f' between 2 consecutive positive roots of f. We conclude that f has at most $(t-2)+1 = t-1$ positive roots.

In (1) we have a sum of k terms instead of t monomials, but the basic strategy remains the same: we divide by the first term and take the derivative. This has the effect of removing a term, but it also has the effect (unlike Descartes' rule) of increasing the complexity of the remaining $k-1$ terms. This results in a larger bound (and a longer proof).

From this upper bound we obtain our permanent lower bound by applying the proof method which was put forward in [11]. More precisely, assume that the permanent has an efficient representation of the form (1). We show that the same must be true for the univariate polynomial $\prod_{i=1}^{2^n}(X-i)$ using a result of Bürgisser [6]. This yields a contradiction with our upper bound on the number of real roots.

Our third result is a polynomial identity testing algorithm. Using a standard substitution technique, we can assume that the polynomials f_j in (1) are univariate. We note that the resulting f_j may be of exponential degree even if the original multivariate f_j are of low degree. The construction of hitting sets is is a classical approach to deterministic

[†] As pointed out by the authors of [4], their result already seems nontrivial for a constant m.

identity testing. Recall that a hitting set for a class $\mathcal{F}$ of polynomials is a set of points H such that for any non-identically zero polynomial $f \in \mathcal{F}$ we have a point $x \in H$ such that $f(x) \neq 0$. Clearly, a hitting set yields a black-box identity testing algorithm (it is not hard to see that the converse is also true). Moreover, for any class $\mathcal{F}$ of univariate polynomials, an upper bound $z(\mathcal{F})$ on the number of real roots of each non-zero polynomial in $\mathcal{F}$ yields a hitting set (any set of $z(\mathcal{F}) + 1$ real numbers will do). From our upper bound result we therefore have polynomial size hitting sets for polynomials of the form (1) when k and m are fixed. Unfortunately, the resulting black-box algorithm does not run in polynomial time: evaluating a polynomial at a point of the hitting set may not be feasible in polynomial time since (as explained above) the f_j may be of very high degree. We therefore use a different strategy. Roughly speaking, we "run" the proof of our upper bound theorem on an input of form (1). This requires explicit knowledge of this representation, and the resulting algorithm is non-black-box. As explained in Section 1.1, for the case where the f_j are low-degree multivariate polynomials an efficient black-box algorithm was recently given in [4].

Organization of the paper. In Section 2 we prove an upper bound on the number of real roots of polynomials of the form (1), see Theorems 1 and 2. In fact, we obtain an upper bound for a more general class of polynomials which we call $\mathsf{SPS}(k, m, t, h)$. This generalization is needed for the inductive proof to go through. From this upper bound, we derive in Section 3 a lower bound on the computational power of (multivariate) circuits of the same form. We give in Section 4 a deterministic identity testing algorithm, again for polynomials of form (1).

2 The real roots of a sum of products of sparse polynomials

2.1 Definitions

In this section, we define precisely the polynomials we are working with. We then explain how to transform those polynomials in a way which reduces the number of terms but does not increase too much the number of roots. This method has some similarities with the proof of Lemma 2 in [12] and it leads to a bound on the number of roots of the polynomials we study.

We say that a polynomial is *t-sparse* if it has at most t monomials.

Definition 1. *Let $\mathsf{SPS}(k, m, t, h)$ denote the class of polynomials $\phi \in \mathbb{R}[X]$ defined by*

$$\phi(X) = \sum_{i=1}^{k} g_i(X) \prod_{j=1}^{m} f_j^{\alpha_{ij}}(X)$$

where

- $g_1, \ldots, g_k$ *are h-sparse polynomials over* $\mathbb{R}$*;*
- $f_1, \ldots, f_m$ *are t-sparse non-zero polynomials over* $\mathbb{R}$*;*
- $\alpha_{11}, \ldots, \alpha_{km}$ *are non-negative integers.*

We define $P_i = \prod_{j=1}^m f_j^{\alpha_{ij}}$ *and* $T_i = g_i P_i$ *for all* i*. We also define* $\pi = \prod_{j=1}^m f_j$*. Finally, we define* $\mathsf{SPS}(k, m, t)$ *as the subclass of* $\mathsf{SPS}(k, m, t, h)$ *in which all the* g_i *are equal to the constant 1.*

Note that $\mathsf{SPS}(k, m, t)$ is just the class of polynomials of form (1), and is included in $\mathsf{SPS}(k, m, t, 1)$. We want to give a bound for the number of real roots of the polynomials in this class, and more generally in $\mathsf{SPS}(k, m, t, h)$. To this end, from a polynomial $\phi \in \mathsf{SPS}(k, m, t, h)$, we build a new polynomial $\tilde{\phi} \in \mathsf{SPS}(k-1, m, t, \tilde{h})$ for some $\tilde{h}$ such that a bound on the number of real roots of $\tilde{\phi}$ yields a bound for ϕ.

Lemma 1. *Let* $\phi \in \mathsf{SPS}(k, m, t, h)$*. If* g_1 *is not identically zero, we write* $\tilde{\phi} = g_1 T_1 \pi (\phi/T_1)'$ *otherwise* $\tilde{\phi} = \phi$*. There exists* $\tilde{h}$ *such that* $\tilde{\phi} \in \mathsf{SPS}(k-1, m, t, \tilde{h})$*.*

Proof. If g_1 is identically zero, the theorem holds with $\tilde{h} = h$. Assume now that g_1 is not identically zero and let

$$\psi(X) = \phi(X)/T_1(X) = 1 + \frac{1}{T_1(X)} \cdot \sum_{i=2}^{k} T_i(X).$$

Then

$$\psi' = \frac{\sum_{i=2}^{k} (T_1 T_i' - T_1' T_i)}{T_1^2}.$$

Notice that $T_i' = g_i' P_i + g_i P_i'$ and

$$P_i' = \sum_{j=1}^{m} \alpha_{ij} f_j' f_j^{\alpha_{ij}-1} \cdot \prod_{l \neq j} f_l^{\alpha_{il}} = P_i \cdot \sum_{j=1}^{m} \alpha_{ij} f_j'/f_j.$$

Therefore

$$\begin{aligned}
\psi' &= \frac{1}{T_1^2} \cdot \sum_{i=2}^{k} (g_1 P_1 g_i' P_i + g_1 P_1 g_i P_i' - g_1' P_1 g_i P_i - g_1 P_1' g_i P_i) \\
&= \frac{1}{T_1^2} \cdot \sum_{i=2}^{k} (g_1 g_i' P_1 P_i + g_1 g_i P_1 P_i \sum_j \alpha_{ij} f_j'/f_j \\
&\qquad\qquad - g_1' g_i P_1 P_i - g_1 g_i P_1 P_i \sum_j \alpha_{1j} f_j'/f_j) \\
&= \frac{1}{g_1 T_1} \cdot \sum_{i=2}^{k} P_i \left(g_1 g_i' - g_1' g_i + g_1 g_i \sum_j (\alpha_{ij} - \alpha_{1j}) f_j'/f_j \right).
\end{aligned}$$

We now multiply ψ' by $\pi = \prod_j f_j$ and get

$$\pi\psi' = \frac{1}{g_1 T_1} \cdot \sum_{i=2}^{k} P_i \left(\pi \cdot (g_1 g_i' - g_1' g_i) + g_1 g_i \sum_j (\alpha_{ij} - \alpha_{1j}) f_j' \prod_{l \neq j} f_l \right).$$

Thus $g_1 T_1 \pi \psi'$ is a polynomial of the class $\mathsf{SPS}(k-1, m, t, \tilde{h})$ for some $\tilde{h}$. Let us write

$$\tilde{\phi} = g_1 T_1 \pi \psi' = \sum_{i=2}^{k} P_i \tilde{g}_i.$$

The integer $\tilde{h}$ denotes the maximum number of monomials in $\tilde{g}_i$ for $2 \leq i \leq k$. □

Definition 2. *Let $(\phi_n)_{1 \leq n \leq k}$ be the sequence defined by $\phi_1 = \phi$ and for $n \geq 1$, $\phi_{n+1} = \widetilde{\phi_n}$. Let also, for $1 \leq i \leq k$, $(g_i^{(n)})_{1 \leq n \leq i}$ be defined by $g_i^{(1)} = g_i$ and $g_i^{(n+1)} = \widetilde{g_i^{(n)}}$ for $i > n$. In other words*

$$\phi_n = \sum_{i=n}^{k} g_i^{(n)} \prod_{j=1}^{m} f_j^{\alpha_{ij}}.$$

We also define the sequence $(h_n)_{1 \leq n \leq k}$ by $h_1 = 1$ and $h_{n+1} = \tilde{h}_n$. That is, each $g_i^{(n)}$ is h_n-sparse.

2.2 A generalization of Descartes' rule

In Definition 2 we defined a sequence of polynomials (ϕ_n) and a sequence of integers (h_n). In this section we first prove that the number of real roots of ϕ_n is bounded by the number of real roots of ϕ_{n+1} up to a multiplicative constant. Then, we give an upper bound on h_n and we combine these ingredients to obtain a bound on the number of real roots of a polynomial in $\mathsf{SPS}(k, m, t)$. This bound (in Theorem 1 at the end of the section) is polynomial in t.

We denote by $r(P)$ the number of distinct real roots of a rational function P. In order to obtain a bound on $r(\phi)$ from a bound on $r(\tilde{\phi})$, we need the following lemma.

Lemma 2. *Let $P \in \mathsf{SPS}(1, m, t, h)$. If P is not identically zero then*

$$r(P) \leq 2h + 2m(t-1) - 1.$$

Proof. By definition, $P = g \cdot \prod_j f_j^{\alpha_j}$. The number of non-zero real roots of P is therefore bounded by the sum of the number of non-zero real roots of g and of the f_j's. Since g is h sparse, we know from Descartes' rule that is has at most $2(h-1)$ non-zero real roots. Likewise, each f_j has at most $2(t-1)$ real roots. As a result, P has at most $2(h-1) + 2m(t-1)$ non-zero real roots. Since 0 can also be a root, we add 1 to this bound to obtain the final result. □

Lemma 3. *Let $\phi \in \mathsf{SPS}(k, m, t, h)$. Then*

$$r(\phi) \leq r(\tilde{\phi}) + 4h + 4m(t-1) - 1.$$

Proof. If g_1 is zero in the definition of ϕ, then $\tilde{\phi} = \phi$ which proves the lemma.

Recall from the proof of Lemma 1 the notation $\psi = \phi / T_1$. If g_1 is not identically zero, by definition we have $\tilde{\phi} = g_1 T_1 \pi \psi'$, so the number $r(\tilde{\phi})$ of real roots of the polynomial $\tilde{\phi}$ is an upper bound on the number of real roots of ψ'.

Since $\phi = T_1\psi$, we have $r(\phi) \leq r(T_1)+r(\psi)$. Moreover, between two consecutive roots of the rational function ψ, we have a root of ψ' or a root of the denominator T_1. As a result, $r(\psi) \leq r(\psi')+r(T_1)+1$. It follows that $r(\phi) \leq r(\psi')+2r(T_1)+1 \leq r(\tilde{\phi})+2r(T_1)+1$. Moreover, the polynomial $T_1 = g_1 \cdot \prod_j f_j^{\alpha_{1j}}$ is in $\mathsf{SPS}(1, m, t, h)$. Thus by Lemma 2, T_1 has at most $2h + 2m(t-1) - 1$ real roots. We conclude that ϕ has at most

$$r(\tilde{\phi}) + 2 \cdot (2h + 2m(t-1) - 1) + 1 = r(\tilde{\phi}) + 4h + 4m(t-1) - 1$$

real roots. □

Proposition 2. *Let $\phi \in \mathsf{SPS}(k, m, t, 1)$. Then*

$$r(\phi) \leq 2h_k + 4\sum_{i=1}^{k-1} h_i + 2m(2k-1)(t-1) - k.$$

Proof. Lemma 3 gives the following recurrence:

$$r(\phi_n) \leq r(\phi_{n+1}) + 4h_n + 4m(t-1) - 1.$$

Thus, we get

$$r(\phi) \leq r(\phi_k) + 4\sum_{i=1}^{k-1} h_i + (k-1)(4m(t-1) - 1). \tag{2}$$

Since $\phi_k \in \mathsf{SPS}(1, m, t, h_k)$, Lemma 2 bounds its number of real roots:

$$r(\phi_k) \leq 2h_k + 2m(t-1) - 1. \tag{3}$$

The bound is a combination of (2) and (3). □

Proposition 2 shows that in order to bound $r(\phi)$, we need a bound on h_n.

Proposition 3. *For all n, h_n is bounded by $((m+2)t^m)^{2^{n-1}-1}$.*

Proof. As showed in the proof of Lemma 1, $\tilde{\phi} = \sum_{i=2}^{k} \tilde{g}_i P_i$ where each $\tilde{g}_i$ is $\tilde{h}$-sparse. More precisely,

$$\tilde{g}_i = (g_1 g_i' - g_1' g_i)\prod_{j=1}^{m} f_j + g_1 g_i \sum_{j=1}^{m} (\alpha_{ij} - \alpha_{1j}) f_j' \prod_{l \neq j} f_l.$$

Thus $\tilde{g}_i$ is a sum of $(m+2)$ terms, and each term is a product of m t-sparse polynomials by two h-sparse polynomials. Thus $\tilde{h} \leq (m+2)t^m h^2$.

This gives the following recurrence relation on h_n:

$$\begin{cases} h_1 & = 1 \\ h_{n+1} & \leq (m+2)t^m h_n^2 \end{cases}$$

Therefore, $h_n \leq ((m+2)t^m)^{2^{n-1}-1}$. □

Now, we combine Propositions 2 and 3 to obtain our first bound on the number of roots of a polynomial in $\mathsf{SPS}(k,m,t)$.

Theorem 1. *Let $\phi \in \mathsf{SPS}(k,m,t)$: we have $\phi = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}}$ where for all i and j, f_j is t-sparse and $\alpha_{ij} \geq 0$. Then $r(\phi) \leq C \times ((m+2)t^m)^{2^{k-1}-1}$ for some universal constant C.*

Proof. It follows from Propositions 2 and 3 that the number of real roots of a polynomial $\phi \in \mathsf{SPS}(k,m,t,1)$ is

$$r(\phi) \leq 2((m+2)t^m)^{2^{k-1}-1} + 4\sum_{i=1}^{k-1}((m+2)t^m)^{2^{i-1}-1} + 2m(2k-1)(t-1) - k.$$

To simplify this expression, note that

$$\sum_{i=1}^{k-1}((m+2)t^m)^{2^{i-1}-1} \leq (k-1)((m+2)t^m)^{2^{k-2}-1}.$$

It is then clear that the function $((m+2)t^m)^{2^{k-1}-1}$ dominates the two smallest terms in the bound on $r(\phi)$. The result follows since $\mathsf{SPS}(k,m,t) \subseteq \mathsf{SPS}(k,m,t,1)$. ☐

2.3 A tighter analysis

This section is devoted to an improved bound for h_n, the number of monomials in the polynomials $g_i^{(n)}$. That automatically sharpens the bound we give for the number of real roots of a polynomial in $\mathsf{SPS}(k,m,t)$.

Let P be a polynomial, and let $S(P)$ be its support, that is the set of integers i such that X^i has a nonzero coefficient in P. Let A be a set of integers, we write $A-\mathbf{1}$ for the set $\{i-1 \mid i \in A\}$. If A and B are two sets, we write $A+B$ for the set $\{i+j \mid i \in A, j \in B\}$ and we write $n \times A$ for the sum of n copies of the set A. Remark that the sum is commutative and that $A + (B - \mathbf{1}) = (A - \mathbf{1}) + B$. We shall use some easy properties of the supports of polynomials. The proof is left to the reader.

Lemma 4. *Let P and Q be two polynomials, then*

1. $S(P') \subseteq S(P) - \mathbf{1}$;
2. $S(P+Q) \subseteq S(P) \cup S(Q)$;
3. $S(PQ) \subseteq S(P) + S(Q)$.

Now consider a polynomial $\phi \in \mathsf{SPS}(k,m,t)$ as in the previous section. Recall that $\phi_n = \sum_{i=n}^{k} g_i^{(n)} P_i$ is the polynomial obtained from ϕ after n steps of the transformation in the first section. Let S be the set $(\sum_j S(f_j)) - \mathbf{1}$. We prove by induction on n that for all $i > n$, $g_i^{(n)}$ satisfies $S(g_i^{(n)}) \subseteq (2^n - 1) \times S$. To this end, we prove the following lemma.

Lemma 5. *Let* $\phi \in \mathsf{SPS}(k, m, t, h)$*, and* $\tilde{\phi} \in \mathsf{SPS}(k-1, m, t, \tilde{h})$ *as defined in Lemma 1. Then*

$$\bigcup_{i=2}^{k} S(\tilde{g}_i) \subseteq 2 \times \left(\bigcup_{i=1}^{k} S(g_i) \right) + S.$$

Proof. To simplify notations, let us define $S_g = \bigcup_i S(g_i)$ and $S_{\tilde{g}} = \bigcup_i S(\tilde{g}_i)$. We aim to show that $S_{\tilde{g}} \subseteq 2 \times S_g + S$.

Recall that

$$\tilde{g}_i = \pi \cdot (g_n g_i' - g_n' g_i) + g_n g_i \sum_j (\alpha_{ij} - \alpha_{nj}) f_j' \prod_{l \neq j} f_l.$$

Applying Lemma 4(2) yields

$$S(\tilde{g}_i) \subseteq S(\pi g_n g_i') \cup S(\pi g_n' g_i) \cup S\left(g_n g_i \sum_j (\alpha_{ij} - \alpha_{nj}) f_j' \prod_{l \neq j} f_l \right).$$

By Lemma 4(3), we have

$$S(\pi g_n g_i') \subseteq S(\pi) + S(g_n) + S(g_i').$$

Moreover, $S(g_n) \subseteq S_g$ and $S(g_i') \subseteq \bigcup_i (S(g_i) - \mathbf{1}) = S_g - \mathbf{1}$. Thus

$$S(\pi g_n g_i') \subseteq S(\pi) + S_g + (S_g - \mathbf{1}).$$

Since $-\mathbf{1}$ commutes with $+$, we obtain:

$$S(\pi g_n g_i') \subseteq (S(\pi) - \mathbf{1}) + 2 \times S_g.$$

Now, $S(\pi) - \mathbf{1} = S$ by definition, and $S(\pi g_n g_i') \subseteq S + 2 \times S_g$. The proof is the same for $S(\pi g_n' g_i) \subseteq S + 2 \times S_g$.

Finally, it holds that

$$S\left(g_n g_i \sum_j (\alpha_{ij} - \alpha_{nj}) f_j' \prod_{l \neq j} f_l \right) \subseteq 2 \times S_g + \bigcup_j S(f_j' \prod_{l \neq j} f_l).$$

Furthermore,

$$\bigcup_j S(f_j' \prod_{l \neq j} f_l) \subseteq \bigcup_j \left((S(f_j) - \mathbf{1}) + \sum_{l \neq j} S(f_l) \right) = S.$$

Therefore we have

$$S\left(g_n g_i \sum_j (\alpha_{ij} - \alpha_{nj}) f_j' \prod_{l \neq j} f_l \right) \subseteq S + 2 \times S_g.$$

We proved that for every $i > n$, $S(\tilde{g}_i) \subseteq S + 2 \times S_g$. This is enough to conclude that

$$S_{\tilde{g}} \subseteq S + 2 \times S_g.$$

□

Proposition 4. *Let $\phi \in \mathsf{SPS}(k,m,t)$ and let ϕ_n be defined as in Definition 2. Then for $1 \leq n \leq i \leq k$,*

$$S(g_i^{(n)}) \subseteq (2^{n-1}-1) \times S.$$

Proof. We actually show by induction on n that $\bigcup_{i\geq n} S(g_i^{(n)}) \subseteq (2^{n-1}-1) \times S$. For $n = 1$, it is clear since the $g_i^{(1)}$ have degree 0. By definition $g_i^{(n+1)} = \widetilde{g_i^{(n)}}$, thus Lemma 5 proves the induction step. □

We need the following combinatorial lemma to improve the bound of Theorem 1.

Lemma 6. *Let S be a set of integers and $p > 0$. Then*

$$|p \times S| \leq \binom{p+|S|}{p} \leq \left[e \times \left(1 + \frac{|S|}{p}\right)\right]^p.$$

Proof. We want to count the number of different sums of p terms from S. This is bounded from above by the number of non-decreasing sequences of elements from S of length p (where elements can be repeated). To count such non-decreasing sequences, we can assume without loss of generality that $S = \{1,\dots,N\}$ where $N = |S|$. To a non-decreasing sequence $(s_1,\dots,s_p)$, we associate the sequence $(t_1,\dots,t_p)$ defined by $t_i = s_i + i - 1$ for $1 \leq i \leq p$. We claim that this defines a bijection between non-decreasing sequences of length p in $\{1,\dots N\}$ and increasing sequences of length p in $\{1,\dots,N+p\}$. Its inverse is indeed defined by mapping $(t_1,\dots,t_p)$ to $(t_1, t_2-1,\dots,t_p-p-1)$. Now increasing sequences of length p in $\{1,\dots,N+p\}$ are subsets of size p of this set. Thus there are $\binom{N+p}{p}$ such sequences.

A well known bound on the binomial coefficient $\binom{n}{k}$ is $(en/k)^k$. Thus $\binom{N+p}{p} \leq (e(1+N/p))^p$. □

Proposition 4 and Lemma 6 improve the bound on h_n given in Section 2.2. Consequently, we obtain a tighter bound on the number of real roots of a $\mathsf{SPS}(k,m,t)$ polynomial.

Theorem 2. *Let $\phi \in \mathsf{SPS}(k,m,t)$. Then ϕ has at most*

$$C \times \left[e \times \left(1 + \frac{t^m}{2^{k-1}-1}\right)\right]^{2^{k-1}-1}$$

real roots, where C is a universal constant.

Proof. As in Section 2.2, we combine Proposition 2 with the bound we have just obtained for h_n. Recall that

$$r(\phi) \leq 2h_k + 4\sum_{i=1}^{k-1} h_i + 2m(2k-1)(t-1) - k.$$

Moreover the polynomials f_j in a $\mathsf{SPS}(k,m,t)$ polynomial are t-sparse, thus $|S| = \left|(\sum_j S(f_j)) - \mathbf{1}\right| \leq t^m$. We can combine Proposition 4 and Lemma 6 with S and $p = 2^{k-1}-1$ to obtain $h_k \leq \left[e \times \left(1 + \frac{t^m}{2^{k-1}-1}\right)\right]^{2^{k-1}-1}$. Since it dominates the other terms of the sum when t grows, this proves the theorem. □

The bound of Lemma 6 is reached for a set S of "far from each other" integers. More precisely, if the integers in S form a increasing sequence (s_n), such that for all n, $ps_n < s_{n+1}$, then $|p \times S| = \binom{p+S}{p}$. Indeed, two different sums of p integers of S cannot have the same value in this case. If this condition is not satisfied, one can build a set S, whose two different sums of p terms have the same value.

In the proof of Theorem 2, S is built from the supports of the f_j's. In this case, the preceding discussion shows that if the degrees of the f_j's are not very far from each other, we can improve our bound. In particular, it can be shown that if the monomials of the f_j's are clustered, and each cluster has a constant diameter, then t^m can be replaced by the number of cluster in the statement of the theorem.

3 Lower bounds

In this section we introduce a subclass $\mathsf{mSPS}(k,m)$ of the class of "easy to compute" multivariate polynomial families, and we use the results of Section 2.2 to show that it does not contain the permanent family. The polynomials in a $\mathsf{mSPS}(k,m)$ family have the same structure as the univariate polynomials in the class $\mathsf{SPS}(k,m,t)$ from Definition 1. In this section, polynomial families are denoted by their general term in brackets: The polynomial P_n is the n-th polynomial of the family (P_n). When there is no ambiguity on the number of variables, we denote by $\vec{X}$ the tuple of variables of a polynomial P_n.

Definition 3. *We say that a sequence of polynomials (P_n) is in $\mathsf{mSPS}(k,m)$ if there is a polynomial Q such that for all n:*

(i) *P_n depends on at most $Q(n)$ variables.*
(ii) $P_n(\vec{X}) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{jn}^{\alpha_{ij}}(\vec{X})$
(iii) *The bitsize of α_{ij} is bounded by $Q(n)$.*
(iv) *For all $1 \leq j \leq m$, the polynomial f_{jn} has a constant free circuit of size $Q(n)$ and is $Q(n)$-sparse.*

Remark 1. If $(P_n) \in \mathsf{mSPS}(k,m)$ then each P_n has a constant free circuit of size polynomial in n. Indeed from the constant free circuits of the polynomials f_{jn} we can build a constant free circuit for P_n. We have to take the α_{ij}-th power of f_{jn}, which can be done with a circuit of size polynomial in the bitsize of α_{ij} thanks to fast exponentiation. The size of the final circuit is up to a constant the sum of the sizes of these powering circuits and of the circuits giving f_{jn}, which is thus polynomial in n.

Definition 4. *The Pochhammer-Wilkinson polynomial of order 2^n is defined by* $\mathrm{PW}_n = \prod_{i=1}^{2^n}(X - i)$.

Definition 5. *The Permanent over n^2 variables is defined by* $\mathrm{PER}_n = \sum_{\sigma\in\Sigma_n}\prod_{i=1}^{n} X_{i\sigma(i)}$ *where Σ_n is the set of permutations of* $\{1,\dots,n\}$.

We now give a lower bound on the Permanent, using its completeness for VNP [17], a result of Bürgisser on the Pochhammer-Wilkinson polynomials [6] and our bound on the roots of the polynomials in $\mathsf{SPS}(k,m,t)$.

Theorem 3. *The family of polynomials* (PER_n) *is not in* $\mathsf{mSPS}(k,m)$ *for any k and m, i.e., there is no representation of the permanent family of the form*

$$\mathrm{PER}_n(\vec{X}) = \sum_{i=1}^{k}\prod_{j=1}^{m} f_{jn}^{\alpha_{ij}}(\vec{X})$$

where the bitsize of the α_{ij}, the sparsity of the polynomials f_{jn} and their constant-free arithmetic circuit complexity are all bounded by a polynomial function $Q(n)$.

Proof. Assume by contradiction that $(\mathrm{PER}_n) \in \mathsf{mSPS}(k,m)$. By the previous remark, this implies that PER_n can be computed by polynomial size constant free arithmetic circuits. As in the proofs of Theorem 4.1 and 1.2 in [6], it follows from this property that there is a family $(G_n(X_0,\dots,X_n))$ in VNP such that

$$\mathrm{PW}_n(X) = G_n(X^{2^0}, X^{2^1}, \dots, X^{2^n}). \tag{4}$$

Since the permanent is complete for VNP, we have a polynomial h such that

$$\mathrm{PER}_{h(n)}(z_1,\dots,z_{h(n)^2}) = G_n(X_0,\dots,X_n) \tag{5}$$

where the z_i's are either variables of G_n or constants. By hypothesis $(\mathrm{PER}_n) \in \mathsf{mSPS}(k,m)$. Let Q be the corresponding polynomial from Definition 3. From this definition and from (4) and (5) we have

$$\mathrm{PW}_n(X) = \sum_{i=1}^{k}\prod_{j=1}^{m} f_{jn}(X)^{\alpha_{ij}}$$

where $f_{jn}(X)$ is $Q(h(n))$-sparse. This shows that the polynomial PW_n is in $\mathsf{SPS}(k,m,R(n))$ where $R(n) = Q(h(n))$.

We have proved in Theorem 1 that polynomials in $\mathsf{SPS}(k,m,R(n))$ have at most $r(n) = C \times ((m+2)R(n))^m)^{2^{k-1}-1}$ real roots. On the other hand, by construction the polynomial PW_n has 2^n roots, which is larger than $r(n)$ for all large enough n. This yields a contradiction and completes the proof of the theorem. □

Remark 2. It is possible to relax condition (iv) in Definition 3. We can replace it by the less restrictive condition:

(iv') the polynomial f_{jn} is $Q(n)$-sparse,

i.e., we allow polynomials f_{jn} with arbitrary complex coefficients. Theorem 3 still applies to this larger version of the class $\mathsf{mSPS}(k,m)$, but for the proof to go through we need to assume the Generalized Riemann Hypothesis. The only change is at the beginning of the proof: Assuming that the permanent family belongs to the (redefined) class $\mathsf{mSPS}(k,m)$, we can conclude that this family can be computed by polynomial size arithmetic circuits with arbitrary constants. To see this, note that any non-multilinear monomial in any f_{jn} can be deleted since it cannot contribute to the final result (the permanent is multilinear). And since f_{jn} is sparse, there is a polynomial size arithmetic circuit with arbitrary constants to compute its multilinear monomials. The remainder of the proof is essentially unchanged. But to deal with arithmetic circuits with arbitrary constants (from the complex field) instead of constant-free arithmetic circuits, we shall use Corollary 4.2 of [6] instead of Theorems 1.2 and 4.1. This means that we have to assume GRH as in this corollary. It is an intriguing question whether this assumption can be removed from Corollary 4.2 of [6] and from this lower bound result.

4 Polynomial Identity Testing

This section is devoted to a proof that Identity Testing can be done in deterministic polynomial time on the polynomials studied in the previous sections. Recall from Definition 2 that for $\phi = \sum_{i=1}^{k} P_i \in \mathsf{SPS}(k,m,t)$, (ϕ_n) is defined by $\phi_n = \sum_{i=n}^{k} g_i^{(n)} P_i$.

Lemma 7. *Let $\phi \in \mathsf{SPS}(k,m,t)$ and (ϕ_n) as in Definition 2. Then for $l < k$, $\phi_l \equiv 0$ if and only if $\phi_{l+1} \equiv 0$ and ϕ_l has a smaller degree than $g_l^{(l)} P_l$.*

Proof. If for all i, $g_i^{(l)}$ is identically zero, then the lemma holds. If there is at least one which is not identically zero, assume that it is $g_l^{(l)}$ up to a reindexing of the terms.

Let $T_l = g_l^{(l)} P_l$, recall that $\phi_{l+1} = g_l T_l \pi(\phi_l/T_l)'$. If $\phi_l \equiv 0$, then $\phi_{l+1} \equiv 0$. Moreover, we have assumed that $T_l \not\equiv 0$ and it it thus of larger degree than ϕ_l which is identically 0.

Assume now that $\phi_{l+1} \equiv 0$, that is $g_l T_l \pi(\phi_l/T_l)' \equiv 0$. By hypothesis, T_l and π are not identically zero, therefore $(\phi_l/T_l)' \equiv 0$. Thus there is $\lambda \in \mathbb{R}$ such that $\phi_l = \lambda T_l$. Since by hypothesis ϕ_l and T_l have different degrees, $\lambda = 0$ and $\phi_l \equiv 0$. □

To solve PIT, we will need to explicitly compute the sequence of polynomials ϕ_l. Thus, the algorithm is not black-box: it must have access to a representation of the input polynomial under form (1).

Theorem 4. *Let k and m be two integers and $\phi \in \mathsf{SPS}(k,m,t)$: we have $\phi = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}}$ where for all i and j, f_j is t-sparse and $\alpha_{ij} \geq 0$. Then one can test if ϕ is identically zero in time polynomial in t, in the size of the sparse representation of the f_j's and in the α_{ij}'s.*

Proof. Let (ϕ_n) be the sequence defined from ϕ as in Definition 2. Lemma 7 implies that ϕ is identically zero if and only if ϕ_k is identically zero and that for all $l < k$, $\phi_l = \sum_{i=l}^{k} g_i^{(l)} P_i$

has a strictly smaller degree than $g_l^{(l)}P_l$. We also assume that $g_l^{(l)}P_l$ is of highest degree amongst the $g_i^{(l)}P_i$ (always true up to a reordering of these terms).

One can compute the sparse polynomials $g_i^{(l)}$, for all i and l in time polynomial in the size of the f_j's if k and m are fixed. For each l, one can test if the degree of $g_l^{(l)}P_l$ and of ϕ_l differ. One only has to compute the highest degree monomials of each $g_i^{(l)}P_i$ for $i \geq l$. One can do that in time polynomial in the α_{ij} (not their bitsize) and the size of the f_j's.

Finally, $\phi_k = g_k^{(k)}P_k$ therefore it is identically zero if and only if $g_k^{(k)}$ is identically zero and we have computed it explicitly. □

This algorithm is polynomial in the α_{ij}'s, though ideally we would like it to be polynomial in their bitsize.

Proposition 5. *Assume that we have access to an oracle which decides whether*

$$\sum_{i=1}^{k}\prod_{j=1}^{m} a_{ij}^{\alpha_{ij}} = 0. \tag{6}$$

Let $\phi = \sum_{i=1}^{k}\prod_{j=1}^{m} f_j^{\alpha_{ij}}$ as in Theorem 4. Then one can decide deterministically whether ϕ is identically zero in time polynomial in the sparsity of the f_j's and in the bitsize of the a_{ij}'s and α_{ij}'s.

Proof. The only dependency in the α_{ij}'s in the proof of Theorem 4 is the computation of the coefficient of the highest degree monomials of the $g_i^{(l)}P_i$. With the oracle for (6), we skip this step and achieve a polynomial dependency in the bitsize of the α_{ij}'s. □

A direct computation of the constant on the left-hand side of (6) is not possible since it involves numbers of exponential bitsize (the exponents α_{ij} are given in binary notation). The test to 0 can be made by computing modulo random primes, but this is ruled out since we want a deterministic algorithm. Note also that this test is a PIT problem for polynomials in $\mathsf{SPS}(k, m, t)$ where the f_j's are constant polynomials. For general arithmetic circuits, it is likewise known that PIT reduces to the case of circuits without any variable occurrence ([3], Proposition 2.2).

The polynomial identity test from Theorem 4 can also be applied to the class of multivariate polynomial families $\mathsf{mSPS}(k, m)$ introduced in the previous section. Indeed, let $P(X_1, \ldots, X_n) = \sum_i \prod_j f_j^{\alpha_{ij}}$ belongs to some $\mathsf{mSPS}(k, m)$ family, and suppose we know a bound d on its degree. We turn P into a univariate polynomial Q by the classical substitution (sometimes attributed to Kronecker) $X_i \mapsto X^{(d+1)^i}$. We write $Q(X) = \sum_i \prod_j g_j^{\alpha_{ij}}$, where each univariate polynomial g_j is the image of f_j by the substitution. It is a folklore result that $P \equiv 0$ if and only if $Q \equiv 0$, thus we can apply the PIT algorithm of Theorem 4 on Q.

Let s be the size of the representation of P, meaning that P depends on at most s variables, the f_j's have a constant free circuit of size at most s and are s-sparse, and the

α_{ij} are at most equal to s. (Note that we do not bound their bitsizes but their values as it is needed for our PIT algorithm.) Then the degree of the f_j's is at most 2^s, and $d \leq 2^{\mathsf{poly}(s)}$ where $\mathsf{poly}(s)$ denotes some polynomial function of s. The g_j's therefore have a degree at most $2^{s\mathsf{poly}(s)} \times 2^s = 2^{s\mathsf{poly}(s)+s}$. This proves that Q satisfies the hypothesis of Theorem 4.

5 Conclusion

We have shown that the real τ-conjecture from [11] holds true for a restricted class of polynomials, and from this result we have obtained an identity testing algorithm and a lower bound for the permanent. Other simple cases of the conjecture remain open. In the general case, we can expand a sum of product of sparse polynomials as a sum of at most kt^m monomials. There are therefore at most $2kt^m - 1$ real roots. As pointed out in [11], the case $k = 2$ is already open: is there a polynomial bound on the number of real roots in this case? Even simpler versions of this question are open. For instance, we can ask whether the number of real roots of an expression of the form $f_1 \cdots f_m + 1$ is polynomial in m and t. A bare bones version of this problem was pointed out by Arkadev Chattopadhyay (personal communication): taking $m = 2$, we can ask what is the maximum number of real roots of an expression of the form $f_1 f_2 + 1$. Expansion as a sum of monomials yields a $O(t^2)$ upper bound, but for all we know the true bound could be $O(t)$.

References

1. M. Agrawal and R. Saptharishi. Classifying Polynomials and Identity Testing. In *Current Trends in Science*, pages 149–162. Indian Academy of Sciences, 2009.
2. M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75, 2008.
3. 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.
4. M. Beecken, J. Mittmann, and N. Saxena. Algebraic independence and blackbox identity testing. *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*, 2011. Arxiv preprint arXiv:1102.2789.
5. A. Borodin and S. Cook. On the number additions to compute specific polynomials. *SIAM Journal on Computing*, 5(1):146–157, 1976.
6. P. Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18(1):81–103, 2009.
7. P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory.* Springer, 1997.
8. 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.
9. V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1):1–46, 2004.
10. P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Arxiv preprint arXiv:1006.4700*, 2010.

11. P. Koiran. Shallow circuits with high-powered inputs. *Proceedings of the Second Symposium on Innovations in Computer Science*, 2011.
12. T.Y. Li, J.M. 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.
13. N. Saxena. Progress on Polynomial Identity Testing. *Bull. EATCS*, 99:49–79, 2009.
14. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomials identities. *Journal of the ACM*, 27:701–717, 1980.
15. 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.
16. S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998.
17. L.G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 249–261, 1979.