On fixed-polynomial size circuit lower bounds for uniform polynomials in the sense of Valiant

Hervé Fournier * Sylvain Perifel † Rémi de Verclos ‡

May 11, 2014

Abstract

We consider the problem of fixed-polynomial lower bounds on the size of arithmetic circuits computing uniform families of polynomials. Assuming the Generalised Riemann Hypothesis (GRH), we show that for all $k$, there exist polynomials with coefficients in $\text{MA}$ having no arithmetic circuits of size $O(n^k)$ over $\mathbb{C}$ (allowing any complex constant). We also build a family of polynomials that can be evaluated in $\text{AM}$ having no arithmetic circuits of size $O(n^k)$. Then we investigate the link between fixed-polynomial size circuit bounds in the Boolean and arithmetic settings. In characteristic zero, it is proved that $\text{NP} \not\subseteq \text{size}(n^k)$, or $\text{MA} \subseteq \text{size}(n^k)$, or $\text{NP} = \text{MA}$ imply lower bounds on the circuit size of uniform polynomials in $n$ variables from the class $\text{VNP}$ over $\mathbb{C}$, assuming GRH. In positive characteristic $p$, uniform polynomials in $\text{VNP}$ have circuits of fixed-polynomial size if and only if both $\text{VP} = \text{VNP}$ over $\mathbb{F}_p$ and $\text{Mod}_p\text{P}$ has circuits of fixed-polynomial size.

1 Introduction

Baur and Strassen [3] proved in 1983 that the number of arithmetic operations needed to compute the polynomials $x_1^n + \ldots + x_n^n$ is $\Omega(n \log n)$. This is still the best lower bound on uniform polynomials on $n$ variables and of degree $n^{O(1)}$, if uniformity means having circuits computed in polynomial time.

If no uniformity condition is required, lower bounds for polynomials have been known since Lipton [13]. For example, Schnorr [18], improving on [13] and Strassen [20], showed for any $k$ a lower bound $\Omega(n^k)$ on the complexity of a family $(P_n)$ of univariate polynomials of degree polynomial in $n$ – even allowing arbitrary complex constants in the circuits. The starting point of Schnorr’s method is to remark that the coefficients of a polynomial computed by a circuit using constants $\alpha = (\alpha_1, \ldots, \alpha_p)$ is given by a polynomial mapping in $\alpha$. Hence, finding hard polynomials reduces to finding a point outside the image of the mapping associated to some circuit which is universal for a given size. This method has been studied and extended by Raz [16].

In the Boolean setting, this kind of fixed-polynomial lower bounds has already drawn a lot of attention, from Kannan’s result [10] proving that for all $k$, $\Sigma_2^P$ does not have circuits

*Univ Paris Diderot, Sorbonne Paris Cité, Institut de Mathématiques de Jussieu, UMR 7586 CNRS, F-75205 Paris, France. Email: fournier@math.univ-paris-diderot.fr.
†Univ Paris Diderot, Sorbonne Paris Cité, LIAFA, UMR 7089 CNRS, F-75205 Paris, France. Email: sylvain.perifel@liafa.univ-paris-diderot.fr.
‡ENS Lyon, F-69342 Lyon, France. Email: remi.de_joannis_de_verclos@ens-lyon.fr.
of size $n^k$, to [5], delineating the frontier of Boolean classes which are known to have fixed-polynomial size circuits lower bounds. It might seem easy to prove similar lower bounds in the algebraic world, but the fact that arbitrary constants from the underlying field (e.g. $\mathbb{C}$) are allowed prevents from readily adapting Boolean techniques.

Different notions of uniformity can be thought of, either in terms of the circuits computing the polynomials, or in terms of the complexity of computing the coefficients. For instance, an inspection of the proof of Schnorr’s result mentioned above shows that the coefficients of the polynomials can be computed in exponential time. But this complexity is generally considered too high to qualify these polynomials as uniform.

The first problem we tackle is the existence of hard polynomials (i.e. without small circuits over $\mathbb{C}$) but with coefficients that are “easy to compute”. The search for a uniform family of polynomials with no circuits of size $n^k$ was pursued recently by Jansen and Santhanam [8]. They show in particular that there exist polynomials with coefficients in MA (thus, uniform in some sense) but not computable by arithmetic circuits of size $n^k$ over $\mathbb{Z}$. Assuming the Generalised Riemann Hypothesis (GRH), we extend their result to the case of circuits over the complex field. GRH is used to eliminate the complex constants in the circuits, by considering solutions over $\mathbb{F}_p$ of systems of polynomial equations, for a small prime $p$, instead of solutions over $\mathbb{C}$. In fact, the family of polynomials built by Jansen and Santhanam is also uniform in the following way: it can be evaluated at integer points in MA. Along this line, we obtain families of polynomials without arithmetic circuits of size $n^k$ over $\mathbb{C}$ and that can be evaluated in AM. The arbitrary complex constants prevents us to readily adapt Jansen and Santhanam’s method and we need to use in addition the AM protocol of Koiran [11] in order to decide whether a system of polynomial equations has a solution over $\mathbb{C}$.

Another interesting and robust notion of uniformity is provided by Valiant’s algebraic class VNP, capturing the complexity of the permanent. The usual definition is non-uniform, but a natural uniformity condition can be required and gives two equivalent characterisations: in terms of the uniformity of circuits and in terms of the complexity of the coefficients. This is one of the notions we shall study in this paper and which is also used by Raz [10] (where the term explicit is used to denote uniform families of VNP polynomials). The second problem we study is therefore to give an $\Omega(n^k)$ lower bound on the complexity of an $n$-variate polynomial in the uniform version of the class VNP. Note that from Valiant’s criterion, it corresponds to the coefficients being in GapP, so it is a special case of coefficients that are easy to compute. Even though MA may seem a small class in comparison with GapP (in particular due to Toda’s theorem $\text{PH} \subseteq \text{P}^\text{#P}$), the result obtained above does not yield lower bounds for the uniform version of VNP.

We show how fixed-polynomial circuit size lower bound on uniform VNP is connected to various questions in Boolean complexity. For instance, the hypothesis that NP does not have circuits of size $n^k$ for all $k$, or the hypothesis that MA has circuits of size $n^k$ for some $k$, both imply the lower bound on the uniform version of VNP assuming GRH. Concerning the question on finite fields, we show an equivalence between lower bounds on uniform VNP and standard problems in Boolean and algebraic complexity.

The paper is organised as follows. Definitions, in particular of the uniform versions of Valiant’s classes, are given in Section 2. Hard families of polynomials with easy to compute coefficients, or that are easy to evaluate, are built in Section 3. Finally, conditional lower

1Even though this result is not stated explicitly in their paper, it is immediate to adapt their proof to our context.
bounds on uniform $\text{VNP}$ are presented in the last section.

## 2 Preliminaries

### Arithmetic circuits

An arithmetic circuit over a field $K$ is a directed acyclic graph whose vertices have indegree 0 or 2 and where a single vertex (called the output) has outdegree 0. Vertices of indegree 0 are called inputs and are labelled either by a variable $x_i$ or by a constant $\alpha \in K$. Vertices of indegree 2 are called gates and are labelled by $+$ or $\times$.

The polynomial computed by a vertex is defined recursively as follows: the polynomial computed by an input is its label; a $+$ gate (resp. $\times$ gate), having incoming edges from vertices computing the polynomials $f$ and $g$, computes the polynomial $f + g$ (resp. $fg$). The polynomial computed by a circuit is the polynomial computed by its output gate.

A circuit is called constant-free if the only constant appearing at the inputs is $-1$. The formal degree of a circuit is defined by induction in the following way: the formal degree of a leaf is 1, and the formal degree of a sum (resp. product) is the maximum (resp. sum) of the formal degree of the incoming subtrees (thus constants “count as variables” and there is no possibility of cancellation).

We are interested in sequences of arithmetic circuits $(C_n)_{n \in \mathbb{N}}$, computing sequences of polynomials $(P_n)_{n \in \mathbb{N}}$ (we shall usually drop the subscript “$n \in \mathbb{N}$”).

**Definition 1.** Let $K$ be a field. If $s : \mathbb{N} \to \mathbb{N}$ is a function, a family $(P_n)$ of polynomials over $K$ is in $\text{asize}_K(s(n))$ if it is computed by a family of arithmetic circuits of size $O(s(n))$ over $K$.

Similarly, $\text{size}(s(n))$ denotes the set of (Boolean) languages decided by Boolean circuits of size $O(s(n))$.

### Counting classes

A function $f : \{0,1\}^* \to \mathbb{N}$ is in $\#P$ if there exists a polynomial $p(n)$ and a language $A \in \text{P}$ such that for all $x \in \{0,1\}^*$

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

A function $g : \{0,1\}^* \to \mathbb{Z}$ is in $\text{GapP}$ if there exist two functions $f,f' \in \#P$ such that $g = f - f'$. The class $\text{C}_s\text{P}$ is the set of languages $A = \{x, g(x) = 0\}$ for some function $g \in \text{GapP}$. The class $\oplus\text{P}$ is the set of languages $A = \{x, f(x) \text{ is odd}\}$ for some function $f \in \#P$. We refer the reader to [6] for more details on counting classes.

### Valiant’s classes and their uniform counterpart

Let us first recall the usual definition of Valiant’s classes.

**Definition 2** (Valiant’s classes). Let $K$ be a field. A family $(P_n)$ of polynomials over $K$ is in the class $\text{VP}_K$ if the degree of $P_n$ is polynomial in $n$ and $(P_n)$ is computed by a family $(C_n)$ of polynomial-size arithmetic circuits over $K$. 

3
A family \((Q_n(x))\) of polynomials over \(K\) is in the class \(\text{VNP}_K\) if there exists a family \((P_n(x, y)) \in \text{VP}_K\) such that
\[
Q_n(x) = \sum_{y \in \{0, 1\}^{|y|}} P_n(x, y).
\]

The size of \(x\) and \(y\) is limited by the circuits for \(P_n\) and is therefore polynomial. Note that the only difference between \(\text{VP}_K\) and \(\text{size}_K(\text{poly})\) is the constraint on the degree of \(P_n\). If the underlying field \(K\) is clear, we shall drop the subscript “\(K\)” and speak only of \(\text{VP}\) and \(\text{VNP}\). Based on these usual definitions, we now define uniform versions of Valiant’s classes.

**Definition 3** (Uniform Valiant’s classes). Let \(K\) be a field. A family of circuits \((C_n)\) is called uniform if the (usual, Boolean) encoding of \(x\) polynomials \((\text{constant-free} \text{ arithmetic circuits of polynomial formal degree})\.

A family of polynomials \((Q_n(x))\) over \(K\) is in the class \(\text{unif-VP}_K\) if it is computed by a uniform family of constant-free arithmetic circuits of polynomial formal degree.

A family of polynomials \((Q_n(x))\) over \(K\) is in the class \(\text{unif-VNP}_K\) if \(Q_n\) has \(n\) variables \(x = x_1, \ldots, x_n\) and there exists a family \((P_n(x, y)) \in \text{unif-VP}_K\) such that
\[
Q_n(x) = \sum_{y \in \{0, 1\}^{|y|}} P_n(x, y).
\]

The uniformity condition implies that the size of the circuit \(C_n\) in the definition of \(\text{unif-VP}\) is polynomial in \(n\). Note that \(\text{unif-VP}_K\) and \(\text{unif-VNP}_K\) only depend on the characteristic of the field \(K\) (indeed, since no constant from \(K\) is allowed in the circuits, these classes are equal to the ones defined over the prime subfield of \(K\)).

In the definition of \(\text{unif-VNP}\), we have chosen to impose that \(Q_n\) has \(n\) variables because this enables us to give a very succinct and clear statement of our questions. This is not what is done in the usual non-uniform definition where the number of variables is only limited by the (polynomial) size of the circuit.

The well-known “Valiant’s criterion” is easily adapted to the uniform case in order to obtain the following alternative characterisation of \(\text{unif-VNP}\).

**Proposition 1** (Valiant’s criterion). In characteristic zero, a family \((P_n)\) is in \(\text{unif-VNP}\) iff \(P_n\) has \(n\) variables, a polynomial degree and its coefficients are computable in \(\text{GapP}\); that is, the function mapping \((c_1, \ldots, c_n)\) to the coefficient of \(X_1^{c_1} \cdots X_n^{c_n}\) in \(P_n\) is in \(\text{GapP}\).

The same holds in characteristic \(p > 0\) with coefficients in \(\text{GapP} \mod p\).

Over a field \(K\), a polynomial \(P(x_1, \ldots, x_n)\) is said to be a projection of a polynomial \(Q(y_1, \ldots, y_m)\) if \(P(x_1, \ldots, x_n) = Q(a_1, \ldots, a_m)\) for some choice of \(a_1, \ldots, a_m \in \{x_1, \ldots, x_n\} \cup K\). A family \((P_n)\) reduces to \((Q_n)\) (via projections) if \(P_n\) is a projection of \(Q_{q(n)}\) for some polynomially bounded function \(q\).

The Hamiltonian Circuit polynomials are defined by
\[
\text{HC}_n(x_1, \ldots, x_{n+1}) = \sum_{\sigma} \prod_{i=1}^{n} x_{i, \sigma(i)},
\]
where the sum is on all cycles \(\sigma \in S_n\) (i.e. on all the Hamiltonian cycles of the complete graph over \(\{1, \ldots, n\}\)). The family \((\text{HC}_n)\) is known to be \(\text{VNP}\)-complete over any field [21] (for projections).

\[\text{This is equivalent to the fact that for all } v \in F_p, \text{ the set of monomials having coefficient } v \text{ is in } \text{Mod}_p\text{.}\]
Elimination of complex constants in circuits

The weight of a polynomial $P \in \mathbb{C}[X_1, \ldots, X_n]$ is the sum of the absolute values of its coefficients. We denote it by $\omega(P)$. It is well known that $\omega$ is a norm of algebra, that is: for $P, Q \in \mathbb{C}[X_1, \ldots, X_n]$ and $\alpha \in \mathbb{C}$, it holds that $\omega(PQ) \leq \omega(P)\omega(Q)$, $\omega(P + Q) \leq \omega(P) + \omega(Q)$ and $\omega(\alpha P) = |\alpha|\omega(P)$.

The following result gives a bound on the weight of a polynomial computed by a circuit.

**Lemma 1.** Let $P$ be a polynomial computed by an arithmetic circuit of size $s$ and formal degree $d$ with constants of absolute value bounded by $M \geq 2$, then $\omega(P) \leq M^{s \cdot d}$.

**Proof.** We prove it by induction on the structure of the circuit $C$ which computes $P$. The inequality is clear if the output of $C$ is a constant or a variable since $\omega(P) \leq M$, $s \geq 1$ and $d \geq 1$ in this case. If the output of $P$ is a $+$ gate then $P$ is the sum of the value of two polynomials $P_1$ and $P_2$ calculated by subcircuits of $C$ of formal degree at most $d$ and size at most $s - 1$. By induction hypothesis, we have $\omega(P_1) \leq M^{d(s-1)}$ and $\omega(P_2) \leq M^{d(s-1)}$. We have $\omega(P) \leq \omega(P_1) + \omega(P_2)$ so $\omega(P) \leq 2 \cdot M^{d(s-1)} \leq M^{d(s-1)+1} \leq M^{da}$. If the output of $C$ in a $\times$ gate, $P$ is the product some polynomials $P_1$ and $P_2$ each calculated by circuits of size at most $s - 1$ and degrees $d_1$ and $d_2$ respectively such that $d_1 + d_2 = d$. Then $\omega(P) \leq \omega(P_1)\omega(P_2) \leq M^{(s-1)d_1}M^{(s-1)d_2} = M^{(s-1)d} \leq M^{sd}$.

For $a \in \mathbb{N}$, we denote by $\pi(a)$ the number of prime numbers smaller than or equal to $a$. For a system $S$ of polynomial equations with integer coefficients, we denote by $\pi_S(a)$ the number of prime numbers $p \leq a$ such that $S$ has a solution over $\mathbb{F}_p$. The following lemma will be useful for eliminating constants from $\mathbb{C}$. (Note that the similar but weaker statement first shown by Koiran [11] as a step in his proof of Theorem 4 would be enough for our purpose.)

**Lemma 2** ([Bürgisser [4] p. 64]). Let $S$ be a system of polynomial equations

$$P_1(x) = 0, \ldots, P_n(x) = 0$$

with coefficients in $\mathbb{Z}$ and with the following parameters: $n$ unknowns, and for all $i$, degree of $P_i$ at most $d$ and $\omega(P_i) \leq w$. If the system $S$ has a solution over $\mathbb{C}$ then under GRH,

$$\pi_S(a) \geq \frac{\pi(a)}{d^{O(a)}} - \sqrt{a \log(2a)}.$$

At last, we need a consequence of VNP having small arithmetic circuits over the complex field.

**Lemma 3.** Assume GRH. If $VP = VNP$ over $\mathbb{C}$, then $CH = MA$.

**Proof.** Assume $VP = VNP$ over $\mathbb{C}$. From the work on Boolean parts of Valiant’s classes [4 Chapter 4], this implies $P/poly = PP/poly = CH/poly$, therefore $MA = CH$ [14].

## 3 Hard polynomials with coefficients in MA

We begin with lower bounds on polynomials with coefficients in PH before bringing them down to MA.
**Hard polynomials with coefficients in PH**

We first need to recall a couple of results. The first one is an upper bound on the complexity of the following problem called HN:

**Input** A system \( S = \{P_1 = 0, \ldots, P_m = 0\} \) of \( n \)-variate polynomial equations with integer coefficients, each polynomial \( P_i \in \mathbb{Z}[x_1, \ldots, x_n] \) being given as a constant-free arithmetic circuit.

**Question** Does the system \( S \) have a solution over \( \mathbb{C}^n \)?

**Theorem 4** (Koiran [11]). Assuming GRH is true, \( HN \in \text{PH} \).

Koiran’s result is stated here for polynomials given by arithmetic circuits, instead of the list of their coefficients. Adapting the result of the original paper in terms of arithmetic circuits is not difficult: it is enough to add one equation per gate expressing the operation made by the gate, thus simulating the whole circuit.

The second result is used in the proof of Schnorr’s result mentioned in the introduction.

**Lemma 4** (Schnorr [18]). Let \( (U_n) \) be the family of polynomials defined inductively as follows:

\[
\begin{align*}
U_1 &= a_0^{(1)} + b_0^{(1)} x \\
U_n &= \left( \sum_{i=1}^{n-1} a_i^{(n)} U_i \right) \left( \sum_{i=1}^{n-1} b_i^{(n)} U_i \right) \\
& \text{where } a_i^{(n)}, b_i^{(n)} \text{ are new variables.}
\end{align*}
\]

Thus \( U_n \) has variables \( x, a_i^{(j)} \text{ and } b_i^{(j)} \) (for \( 1 \leq j \leq n \) and \( 0 \leq i < j \)). For simplicity, we will write \( U_n(a,b,x) \), where the total number of variables in the tuples \( a, b \) is \( n(n+1) \).

For every univariate polynomial \( P(x) \) over \( \mathbb{C} \) computed by an arithmetic circuit of size \( s \), there are constants \( a, b \in \mathbb{C}^{s(s+1)} \) such that \( P(x) = U_s(a,b,x) \).

The polynomials \( U_s \) in this lemma are universal in the sense that they can simulate any circuit of size \( s \); the definition of such a polynomial indeed reproduces the structure of an arbitrary circuit by letting at each gate the choice of the inputs and of the operation, thanks to new variables.

The third result we’ll need is due to Hrubeš and Yehudayoff [7] and relies on Bézout’s Theorem. Showing Theorem 5 could also be done without using algebraic geometry, but this would complicate the overall proof.

**Lemma 5** (Hrubeš and Yehudayoff [7]). Let \( F : \mathbb{C}^n \rightarrow \mathbb{C}^m \) be a polynomial map of degree \( d > 0 \), that is, \( F = (F_1, \ldots, F_m) \) where each \( F_i \) is a polynomial of degree at most \( d \). Then \( |F(\mathbb{C}^n) \cap \{0,1\}^m| \leq (2d)^n \).

We are now ready to give our theorem.

**Theorem 5.** Assume GRH is true. For any constant \( k \), there is a family \( (P_n) \) of univariate polynomials with coefficients in \( \{0,1\} \) satisfying:

- \( \deg(P_n) = n^{O(1)} \) (polynomial degree);
- the coefficients of \( P_n \) are computable in \( \text{PH} \), that is, on input \((1^n, i)\) we can decide in \( \text{PH} \) if the coefficient of \( x^i \) is 1;
- \( (P_n) \) is not computed by arithmetic circuits over \( \mathbb{C} \) of size \( n^k \).
Proof. Fix \( s = n^k \). Consider the universal polynomial \( U_s(a, b, x) \) of Lemma 4 simulating circuits of size \( s \). If \( \alpha_i(s) \) denotes the coefficient of \( x^i \) in \( U_s \), then we have the relation
\[
\alpha_i(s) = \sum_{i_1 + i_2 = i, i_1, i_2 \leq s} a_{i_1} b_{i_2} \alpha_{i_1}(s) \alpha_{i_2}(s).
\]

By induction, the coefficient \( \alpha_i(s) \) is therefore a polynomial in \( a, b \) of degree \( \leq (i + 1)2^{2s} \).

Now, we would like to find a polynomial whose coefficients are different from the \( \alpha_i(s) \) for any value of \( a, b \). This will be done thanks to Lemma 5 but we have to use it in a clever way because our method requires to use interpolation on \( d + 1 \) points to identify two polynomials of degree \( d \): hence we need to “truncate” the polynomial \( U_s \) to degree \( d \).

Fix \( d = s^4 \). It follows from the beginning of the proof that the map computing the first \((d + 1)\) coefficients of \( U_s \)
\[
F : \mathbb{C}^{s(s+1)} \rightarrow \mathbb{C}^{d+1}
(a, b) \mapsto (\alpha_0(s), \ldots, \alpha_d(s))
\]
is a polynomial map of degree at most \((d+1)2^{2s}\). Since \(((d+1)2^{2s})^{s(s+1)} < 2^{d+1} \), by Lemma 5 there exist coefficients \((\beta_0, \ldots, \beta_d) \in \{0,1\}^{d+1} \) not in \( F(\mathbb{C}^{s(s+1)}) \). In other words, for any values of \( a, b \in \mathbb{C} \), the first \((d+1)\) coefficients of \( U_s \) differ from \((\beta_0, \ldots, \beta_d) \).

Let \( P_{\beta}(x) \) be the polynomial \( \sum_{i=0}^d \beta_i x^i \) and let us call \( U_s|_{d} \) the truncation of \( U_s \) up to degree \( d \), that is, the sum of all the monomials of degree \( \leq d \) in \( x \). For any instantiation of \( a, b \in \mathbb{C} \), we have \( U_s|_{d}(a, b, x) \neq P_{\beta}(x) \). Since both polynomials are of degree smaller than or equal to \( d \), this means that there exists an integer \( m \in \{0, \ldots, d\} \) such that \( U_s|_{d}(a, b, m) \neq P_{\beta}(m) \).

Therefore the following system of polynomial equations with unknowns \( a,b \):
\[
S_{\beta} = \{U_s|_{d}(a, b, m) = P_{\beta}(m) : m \in \{0, \ldots, d\}\}
\]
has no solution over \( \mathbb{C} \).

Conversely, consider now this system for other coefficients than \( \beta \), that is, \( S_{\gamma} \) for \( \gamma_0, \ldots, \gamma_d \in \{0,1\} \). If \( S_{\gamma} \) does not have a solution over \( \mathbb{C} \), this means that for any instantiation of \( a, b \in \mathbb{C} \) we have \( U_s|_{d}(a, b, x) \neq P_{\gamma}(x) \), hence \( P_{\gamma} \) is not computable by a circuit of size \( s \) by Lemma 4.

The goal now is then to find values of \( \gamma \in \{0,1\}^{d+1} \) such that \( S_{\gamma} \) does not have a solution over \( \mathbb{C} \).

Remark first that on input \( \gamma_0, \ldots, \gamma_d \in \{0,1\} \) and \( m \in \{0, \ldots, d\} \), we can describe in polynomial time a circuit \( C_{\gamma,m}(a, b) \) computing the polynomial \( U_s|_{d}(a, b, m) - P_{\gamma}(m) \). Indeed, \( U_s \) is computable by an easily described circuit following its definition, hence its truncation to degree \( d \) also is, and a circuit for \( P_{\gamma} \) is also immediate if we are given \( \gamma \). Therefore, we can describe in polynomial time the system \( S_{\gamma} \) to be used in Theorem 4.

The algorithm in \( \text{PH} \) to compute the coefficients of a polynomial \( P_{\beta} \) without circuits of size \( s \) is then the following on input \((1^n, i)\):

- Find the lexicographically first \( \gamma_0, \ldots, \gamma_d \in \{0,1\} \) such that \( S_{\gamma} \notin \text{HN} \);
- accept iff \( \gamma_i = 1 \).

This algorithm is in \( \text{PH}^{\text{HN}} \). By Theorem 5, if we assume GRH then the problem HN is in \( \text{PH} \). We deduce that computing the coefficients of \( P_{\gamma} \) can be done in \( \text{PH} \).
Hard polynomials with coefficients in MA

Allowing $n$ variables instead of only one, we can even obtain lower bounds for polynomials with coefficients in MA.

**Corollary 1.** Assume GRH is true. For any constant $k$, there is a family $(P_n)$ of polynomials on $n$ variables, with coefficients in $\{0,1\}$, of degree $n^{O(1)}$, with coefficients computable in MA, and such that $(P_n) \notin \text{asize}_c(n^k)$.

**Proof.** If the Hamiltonian family $(HC_n)$ does not have circuits of polynomial size over $\mathbb{C}$, consider the following variant of a family with $n$ variables: $HC_n'(x_1,\ldots,x_n) = HC_{\lfloor \sqrt{n} \rfloor}(x_1,\ldots,x_{\lfloor \sqrt{n} \rfloor^2})$. This is a family whose coefficients are in $\mathbb{P}$ (hence in MA) and without circuits of size $n^k$.

On the other hand, if the Hamiltonian family $(HC_n)$ has circuits of polynomial size over $\mathbb{C}$, then $PH = MA$ by Lemma 3. Therefore the family of polynomials of Theorem 5 has its coefficients in MA.

**Hard polynomials that can be evaluated in AM**

A family of polynomials $(P_n(x_1,\ldots,x_n))$ is said to be evaluable in AM if the language

$$\{(x_1,\ldots,x_n,i,b) \mid \text{the } i\text{-th bit of } P_n(x_1,\ldots,x_n) \text{ is } b\}$$

is in AM, where $x_1,\ldots,x_n,i$ are integers given in binary and $b \in \{0,1\}$. In the next proposition, we show how to obtain polynomials which can be evaluated in AM. The method is based on Santhanam [17] and Koiran [12].

**Proposition 2.** Assume GRH is true. For any constant $k$, there is a family $(P_n)$ of polynomials on $n$ variables, with coefficients in $\{0,1\}$, of degree $n^{O(1)}$, evaluable in AM and such that $(P_n) \notin \text{asize}_c(n^k)$.

**Proof.** We adapt the method of Santhanam [17] to the case of circuits with complex constants.

If the permanent has polynomial-size circuits over $\mathbb{C}$, then $PH = MA$ by Lemma 3 and hence the family of polynomials of Theorem 5 is evaluable in $MA \subseteq \text{AM}$.

Otherwise, call $s(n)$ the minimal size of a circuit over $\mathbb{C}$ for $\text{per}_n$. The $n$-tuple of variables $(x_1,\ldots,x_n)$ is split in two parts $(y,z)$ in the unique way satisfying $0 < |y| \leq |z|$ and $|z|$ a power of two. Remark therefore that $|y|$ can take all the values from 1 to $|z|$ depending on $n$. We now define the polynomial $P_n(y,z)$:

$$
\begin{cases}
P_n(y,z) = \text{per}(y) & \text{if } |y| \text{ is a square and } s(\sqrt{|y|}) \leq n^{2k} \\
P_n(y,z) = 0 & \text{otherwise}.
\end{cases}
$$

Let us first show that $(P_n)$ does not have circuits of size $n^k$. By hypothesis there exist infinitely many $n$ such that $s(n) > (3n^2)^{2k}$; let $n_0$ be one of them and take $m$ the least power of two such that $s(n_0) \leq (m+n_0^2)^{2k}$, which implies $m \geq 2n_0^2$. Let $n_1 = m + n_0^2$; by definition of $(P_n)$, we have $P_{n_1}(y,z) = \text{per}_{n_0}(y)$. By definition of $m$, $s(n_0) > (m/2+n_0^2)^{2k} > (n_1/2)^{2k} > n_1^k$. This means that $\text{per}_{n_0}$, and hence $P_{n_1}$, does not have circuits of size $n_1^k$.

We now show that $(P_n)$ can be evaluated in AM. We give an AMA protocol which is enough since $AMA = AM$ (see [2]).
The protocol described below heavily relies on the technique used in [12, Theorem 2] to prove that $HN \in AM$.

In the following, we need to test if per$_t$ (for some $t$) has an arithmetic circuit of size $s$ over the complex field. If this is true, Merlin can give the skeleton of the circuit but he cannot give the complex constants. Hence, he gives a circuit $C(y, u)$ where $y$ is the input (of size $t \times t$) and $u$ a tuple of formal variables. Consider the following system $S$: for all $\varepsilon \in \{0, \ldots, 2^s\}^{|y|}$, take the equation $C(\varepsilon, u) = \text{per}_t(\varepsilon)$. For some values $\alpha \in \mathbb{C}^{|u|}$, the degree of the polynomial computed by the circuit $C(y, \alpha)$ is at most $2^s$; hence, the system $S$ is satisfiable over $\mathbb{C}$ if the variables $u$ can be replaced by complex numbers $\alpha$ such that $C(y, \alpha)$ computes the permanent over the complex field.

The system $S$ has the following parameters: the number of variables is $|u|$ which is at most $s$, the degree of each equation is bounded by $2^s$, the number of equations is $2^{O(s^2)}$ and the bitsize of each coefficient is $2^{sO(1)}$. Hence, by [12, Theorem 1], there is an integer $m = s^{O(1)}$ and $x_0 = 2^{sO(1)}$ such that the following holds. Let $E$ be the set of primes $p$ smaller than $x_0$ such that $S$ has a solution modulo $p$.

- If $S$ is not satisfiable over $\mathbb{C}$, then $|E| \leq 2^{m-2}$;
- If $S$ is satisfiable over $\mathbb{C}$, then $|E| \geq m2^m$.

Testing if $|E|$ is large or small is done via the following probabilistic argument. For some matrices $A_j$ over $\mathbb{F}_2$, the predicate $\phi(A_1, \ldots, A_m)$ is defined as

$$\exists p_0, p_1, \ldots, p_m \in E : \psi(A_1, \ldots, A_m, p_0, \ldots, p_m)$$

where

$$\psi(A_1, \ldots, A_m, p_0, \ldots, p_m) \equiv \bigwedge_{j=1}^{m} (A_j p_0 = A_j p_j \land p_0 \neq p_j).$$

If $A_j$ are seen as hashing functions, the predicate $\phi$ above expresses that there are enough collisions between elements of $E$. Based on [19], it is proved in [12] that if $|E| \leq 2^{m-2}$, the probability that $\phi(A_1, \ldots, A_m)$ holds is at most $1/2$ when the matrices $A_j$ are chosen uniformly at random, whereas it is 1 when $|E| \geq m2^m$.

We are now ready to explain the $AMA$ protocol to evaluate the family $(P_n)$. On input $(x_1, \ldots, x_n, i, b)$, the $AMA$ protocol is the following:

- Arthur splits $(x_1, \ldots, x_n)$ in $(y, z)$ in the unique way. If $|y|$ is not a square, he accepts if $b = 0$ and rejects if $b \neq 0$. Otherwise, call $t = \sqrt{|y|}$; Arthur sends to Merlin random matrices $A_1, \ldots, A_m$ over $\mathbb{F}_2$.
- Merlin sends to Arthur the skeleton $C(y, u)$ of a circuit of size $\leq n^{2k}$ supposedly computing per$_t$ over $C$ (that is, the circuit with complex constants replaced with formal variables $u$). He also sends prime integers $p_0, \ldots, p_m$ together with constants $\alpha_{p_j} \in \mathbb{F}_{p_j}$ for $C$, for all $0 \leq j \leq m$. He also sends a prime number $p \geq n!M^n$ (where $M$ is the largest value in $(x_1, \ldots, x_n)$) and constants of $\alpha_p$ over $\mathbb{F}_p$ for $C$.
- Arthur checks that $p_0, \ldots, p_m$ produce a collision (that is, that $\psi(A_1, \ldots, A_m, p_0, \ldots, p_m)$ is true). Then he checks that all $p_j$ and $p$ are primes and that the circuits $C(y, \alpha_{p_j})$ and $C(y, \alpha_p)$ compute the permanent modulo $p_0, \ldots, p_m, p$ (using the coRP algorithm of [2]). If any of these tests fails, Arthur accepts iff $b = 0$. Otherwise, he computes $C(y, \alpha_p)$ and accepts iff its $i$-th bit is equal to $b$.  

9
If $(y, z)$ is such that $|y|$ is a square and $s(|y|) \leq n^{2k}$, then $P_n(y, z) = \text{per}(y)$. We show that Merlin can convince Arthur with probability 1. Merlin sends a correct skeleton $C$: since $|E| \geq m2^n$, there are prime integers $p_0, \ldots, p_m \in E$ such that $\psi(A_1, \ldots, A_m, p_0, \ldots, p_m)$ holds. Merlin sends such numbers $p_j$ and $p$ together with the correct constants for the circuit $C$ to compute the permanent modulo $p_j$ and $p$. In the third round, all the verifications are satisfied with probability 1 and Arthur gives the right answer.

On the other hand, if $|y|$ is not a square then whatever Merlin sends, Arthur accepts only if $b = 0$, which is the right answer. Assume now that $s(|y|) > n^{2k}$; then $|E| \leq 2^{m-2}$. Whatever Merlin sends as prime numbers belong to $E$ and produce a collision is at most $1/2$. Since the error when testing if $p_j \in E$ can be made as small as we wish (testing if $C(y, \alpha_{p_j})$ computes $\text{per}(y) \mod p_j$ is done in $\text{coRP}$), the probability that the whole protocol gives the wrong answer in this case is bounded by $2/3$. 

\[ \square \]

4 Conditional lower bounds for uniform $\text{VNP}$

In characteristic zero

In this whole section we assume GRH is true. Our main result in this section is that if for all $k$, $C_\text{NP}$ has no circuits of size $n^k$, then the same holds for $\text{unif-VNP}$ (in characteristic zero). For the clarity of exposition, we first prove the weaker result where the assumption is on the class $\text{NP}$ instead.

**Lemma 6.** If there exists $k$ such that $\text{unif-VNP} \subseteq \text{asize}_C(n^k)$, then there exists $\ell$ such that $\text{NP} \subseteq \text{size}(n^{\ell})$.

**Proof.** Let us assume that $\text{unif-VNP} \subseteq \text{asize}_C(n^k)$. Let $L \in \text{NP}$. There is a polynomial $q$ and a polynomial time computable relation $\phi : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$ such that for all $x \in \{0, 1\}^n$, $x \in L$ if and only if $\exists y \in \{0, 1\}^{q(n)} \phi(x, y) = 1$.

We define the polynomial $P_n$ by

$$P_n(x_1, \ldots, x_n) = \sum_{x \in \{0, 1\}^n} \left( \sum_{y \in \{0, 1\}^{q(n)}} \phi(x, y) \right) \prod_{i=1}^{n} X_i^{x_i} (1 - X_i)^{1-x_i}.$$ 

Note that for $x \in \{0, 1\}^n$, $P_n(x)$ is the number of elements $y$ in relation with $x$ via $\phi$. By Valiant’s criterion (Proposition \( \blacksquare \)), the family $(P_n)$ belongs to $\text{unif-VNP}$ in characteristic 0. By hypothesis, there exists a family of arithmetic circuits $(C_n)$ over $\mathbb{C}$ computing $(P_n)$, with $C_n$ of size $t = O(n^k)$.

Let $\alpha = (\alpha_1, \ldots, \alpha_k)$ be the complex constants used by the circuit. We have $P_n(x_1, \ldots, x_n) = C_n(x_1, \ldots, x_n, \alpha)$. Take one unknown $Y_i$ for each $\alpha_i$ and one additional unknown $Z$, and consider the following system $S$:

$$\begin{cases} \left( \prod_{x \in L \cap \{0, 1\}^n} C_n(x, Y) \right) \cdot Z = 1 \\ C_n(x, Y) = 0 \text{ for all } x \in \{0, 1\}^n \setminus L. \end{cases}$$

Note that introducing one equation for each $x \in L \cap \{0, 1\}^n$ (as we did for each $x \in \{0, 1\}^n \setminus L$) would not work since it would require to introduce an exponential number of new variables.
Let $\beta = \left( \prod_{x \in L \cup \{0,1\}^n} C_n(x, \alpha) \right)^{-1}$. Then $(\alpha, \beta)$ is a solution of $S$ over $\mathbb{C}$.

The system $S$ has $t + 1 = O(n^k)$ unknowns. The degree of $C_n(x, Y)$ is bounded by $2^t$; hence the degree of $S$ is at most $2^O(n^k)$. Moreover, the weight of the polynomials in $S$ is bounded by $2^O(n^k)$ using Lemma 1.

Since the system $S$ has the solution $(\alpha, \beta)$ over $\mathbb{C}$, by Lemma 2 it has a solution over $\mathbb{F}_p$ for some $p$ small enough. We recall that $\pi(p) \sim p/\log p$; hence the system $S$ has a solution over $\mathbb{F}_p$ for $p = 2^O(n^k)$.

Consider $p$ as above and $(\alpha', \beta')$ a solution of the system $S$ over $\mathbb{F}_p$. By definition of $S$, when the circuit $C_n$ is evaluated over $\mathbb{F}_p$, the following is satisfied:

$$\begin{align*}
\forall x \in L \cap \{0,1\}^n, & \quad C_n(x, \alpha') \neq 0, \\
\forall x \in \{0,1\}^n \setminus L, & \quad C_n(x, \alpha') = 0.
\end{align*}$$

Computations over $\mathbb{F}_p$ can be simulated by Boolean circuits, using $\log_2 p$ bits to represent an element of $\mathbb{F}_p$, and $O((\log^2 p)$ gates to simulate an arithmetic operation. This yields Boolean circuits of size $n^\ell$ for $\ell = O(k)$ to decide the language $L$. \hfill \Box

**Theorem 6.** Assume GRH is true. Suppose one of the following conditions holds:

1. $\text{NP} \not\subset \text{size}(n^k)$ for all $k$;
2. $\text{C}=\text{P} \not\subset \text{size}(n^k)$ for all $k$;
3. $\text{MA} \subset \text{size}(n^k)$ for some $k$;
4. $\text{NP} = \text{MA}$.

Then $\text{unif}$-$\text{VNP} \not\subset \text{asize}_\mathbb{C}(n^k)$ for all $k$.

**Proof.** The first point is proved in Lemma 3.

The second point subsumes the first since $\text{coNP} \subseteq \text{C}=\text{P}$. It can be proved in a very similar way. Indeed consider $L \in \text{C}=\text{P}$ and $f \in \text{GapP}$ such that $x \in L \iff f(x) = 0$, and its associated family of polynomials

$$P_n(X_1, \ldots, X_n) = \sum_{x \in \{0,1\}^n} f(x) \prod_{i=1}^n X_i^{x_i} (1 - X_i)^{1-x_i}$$

as in the proof of Lemma 3. Then for all $x \in \{0,1\}^n$, $P_n(x) = 0$ iff $x \in L$. The family $(P_n)$ belongs to $\text{unif}$-$\text{VNP}$ and thus, assuming $\text{unif}$-$\text{VNP} \subset \text{asize}_\mathbb{C}(n^k)$, has arithmetic circuits $(C_n)$ over $\mathbb{C}$ of size $t = O(n^k)$. Constants of $\mathbb{C}$ are replaced with elements of a small finite field by considering the system:

$$\begin{align*}
C_n(x, Y) = 0 & \text{ for all } x \in L \cap \{0,1\}^n \\
\left( \prod_{x \in \{0,1\}^n \setminus L} C_n(x, Y) \right) \cdot Z = 1.
\end{align*}$$

The end of the proof is similar.

For the third point, let us assume $\text{unif}$-$\text{VNP} \subset \text{asize}_\mathbb{C}(\text{poly})$. It implies $\text{VP} = \text{VNP}$ thanks to the $\text{VNP}$-completeness of the uniform family $(\text{HC}_n)$, then $\text{MA} = \text{PP}$ by Lemma 4. This implies $\text{MA} \not\subset \text{size}(n^k)$ for all $k$ since $\text{PP} \not\subset \text{size}(n^k)$ for all $k$ [22].
For the last point, assume $NP = MA$. If $NP$ is without $n^k$ circuits for all $k$, then the conclusion comes from the first point. Otherwise $MA$ has $n^k$-size circuits and the conclusion follows from the previous point.

For any constant $c$, the class $P^{NP[n^c]}$ is the set of languages decided by a polynomial time machine making $O(n^c)$ calls to an $NP$ oracle. It is proven in [5] that $NP \subseteq \text{size}(n^k)$ implies $P^{NP[n^c]} \subseteq \text{size}(n^{ck^2})$. Hence, it is enough to assume fixed-polynomial lower bounds on this larger class $P^{NP[n^c]}$ for some $c$ to get fixed-polynomial lower bounds on $\text{unif-VNP}_C$.

**An unconditional lower bound in characteristic zero**

In this part we do not allow arbitrary constants in circuits. We consider instead circuits with $-1$ as the only scalar that can label the leaves. For $s : \mathbb{N} \to \mathbb{N}$, let $\text{asize}_0(s)$ be the family of polynomials computed by families of unbounded degree constant-free circuits of size $O(s)$ (in characteristic zero). Note that the formal degree of these circuits are not polynomially bounded: hence, large constants produced by small arithmetic circuits can be used.

We first need a result of [1]. Let $\text{PosCoefSLP}$ be the following problem: on input $(C,i)$ where $C$ is a constant-free circuit with one variable $x$ and $i$ is an integer, decide whether the coefficient of $x^i$ in the polynomial computed by $C$ is positive.

**Lemma 7 ([1]).** $\text{PosCoefSLP}$ is in $CH$.

**Theorem 7.** $\text{unif-VNP} \not\subseteq \text{asize}_0(n^k)$ for all $k$.

**Proof.** If the permanent family does not have constant-free arithmetic circuits of polynomial size, then this family matches the statement.

Otherwise, $CH = MA$ by Lemma 3. For a given constant-free circuit $C$ computing a univariate polynomial $P = \sum_{i=0}^d a_ix^i$, its “sign condition” is defined as the series $(b_i)_{i \in \mathbb{N}}$ where $b_i \in \{0, 1\}$, $b_i = 1$ iff $a_i > 0$.

Note that for some constant $\alpha$, there are at most $2^{n^{\alpha k}}$ different sign conditions of constant-free circuits of size $n^k$ (at most one per circuit). Hence there exists a sign condition

$$(b_{i_1}, \ldots, b_{i_{\alpha k}}, 0, 0, \ldots)$$

such that any polynomial with such a sign condition is not computable by constant-free circuits of size $n^k$. We define $b_{i_1}, \ldots, b_{i_{\alpha k}}$ to be the lexicographically first such bits.

We can express these bits as the first in lexicographic order such that for every constant-free circuit $C$, there exists $i$ such that:

$$b_i = 0 \text{ iff the coefficient of } x^i \text{ in } C \text{ is positive.}$$

Therefore they can be computed in $PH^{\text{PosCoefSLP}}$, hence in $CH$ by Lemma 7 hence in $MA$ since $CH = MA$. By reducing the probability of error in the $MA$ protocol, this means that there exists a polynomial-time function $a : \{0, 1\}^* \to \{0, 1\}$ such that:

\[
\begin{align*}
\exists y \sum_r a(i, y, r) &\geq (1 - 2^{-|y|-1})N \quad \text{if } b_i = 1 \\
\forall y \sum_r a(i, y, r) &\leq 2^{-|y|-1}N \quad \text{if } b_i = 0,
\end{align*}
\]
where \( y \) and \( r \) are words of polynomial size, and where \( N = 2^{|r|} \). Now, the following polynomial family:

\[
P_n(x) = \sum_{i=0}^{n^a_k} \left( \sum_{y,r} a(i, y, r) \right) - N/2\right) x^i
\]

is in \( \text{unif-VNP} \) and has sign condition \((b_0, \ldots, b_n^a_k, 0, 0, \ldots)\).

**In positive characteristic**

This subsection deals with fixed-polynomial lower bounds in positive characteristic. The results are presented in characteristic 2 but they hold in any positive characteristic \( p \) (replacing \( \oplus P \) with \( \text{Mod}_p P \)).

**Lemma 8.** Consider the polynomial

\[
P(X_1, \ldots, X_n) = \sum_{y_1, \ldots, y_p \in \{0, 1\}} C(X_1, \ldots, X_n, y_1, \ldots, y_p)
\]

where \( C \) is an arithmetic circuit of size \( s \) and total degree at most \( d \) (with respect to all the variables \( X_1, \ldots, X_n, y_1, \ldots, y_p \)). Then \( P \) is a projection of \( HC_{(sd)O(1)} \).

**Proof.** This lemma follows from a careful inspection of the proof of \( \text{VNP}\)-completeness of the Hamiltonian given in Malod [15]. We give some more details below.

From the fact that \( \text{VNP} = \text{VNP}_e \) [4, Theorem 2.13], we can write \( P \) as a Boolean sum of formulas, i.e.

\[
P(X_1, \ldots, X_n) = \sum_{z_1, \ldots, z_q \in \{0, 1\}} F(X_1, \ldots, X_n, z_1, \ldots, z_q).
\]

Moreover, \( q = s^{O(1)} \) and an inspection of the proof of \( \text{VNP} = \text{VNP}_e \) given in [15] shows that the size of the formula \( F \) is \( (sd)^{O(1)} \). By [15] Lemme 8], a formula is a projection of the Hamiltonian circuit polynomial of linear size. This yields

\[
P(X_1, \ldots, X_n) = \sum_{z_1, \ldots, z_q \in \{0, 1\}} HC_{s'}(a_1, \ldots, a_{s'})
\]

where \( s' = (sd)^{O(1)} \) and \( a_i \in \{X_1, \ldots, X_n, z_1, \ldots, z_q, -1, 0, 1\} \). At last, in order to write this exponential sum as a projection of a not too large Hamiltonian circuit, a sum gadget of size \( O(q) \) and \( O(s') \) XOR gadgets of size \( O(1) \) are needed [15 Théorème 7]. Hence, the polynomial \( P \) is a projection of \( HC_{(sd)O(1)} \). \( \square \)

**Theorem 8.** The following are equivalent:

- \( \text{unif-VNP}_{\mathbb{F}_2} \subset \text{asize}_{\mathbb{F}_2}(n^k) \) for some \( k \);
- \( \text{VP}_{\mathbb{F}_2} = \text{VNP}_{\mathbb{F}_2} \) and \( \oplus P \subset \text{size}(n^k) \) for some \( k \).

**Proof.** Suppose that \( \text{unif-VNP}_{\mathbb{F}_2} \subset \text{asize}_{\mathbb{F}_2}(n^k) \). Then the Hamiltonian polynomials \( (HC_n) \) has \( O(n^k) \) size circuits and thus \( \text{VP} = \text{VNP} \) over \( \mathbb{F}_2 \). Let \( L \in \oplus P \) and the corresponding function \( f \in \#P \) so that

\[
x \in L \iff f(x) \text{ is odd.}
\]
Consider the sequence of polynomials $P_n \in \mathbb{F}_2[X_1, \ldots, X_n]$ associated to $L$:

$$P_n(X_1, \ldots, X_n) = \sum_{x \in \{0,1\}^n} f(x) \prod_{i=1}^n X_i^{x_i}(1 - X_i)^{1-x_i}.$$ 

This family belongs to unif-VNP over $\mathbb{F}_2$. Hence, $P_n$ has $O(n^k)$ size circuits. It can be simulated by a Boolean circuit of the same size within a constant factor, and yields $O(n^k)$ size circuits for $L$. Hence $\oplus \mathbb{P} \subseteq \text{size}(n^k)$.

For the converse, suppose that $\oplus \mathbb{P} \subseteq \text{size}(n^k)$ and $\mathbb{VP}_{\mathbb{F}_2} = \mathbb{VNP}_{\mathbb{F}_2}$, and let $(P_n) \in \text{unif-VNP}_{\mathbb{F}_2}$. We can write

$$P_n(X_1, \ldots, X_n) = \sum_{m_1, \ldots, m_n \in \{0, \ldots, d\}} \phi(m_1, \ldots, m_n) \prod_{i=1}^n X_i^{m_i},$$

where $d$ is a bound on the degree of each variable of $P_n$. Since the coefficients of $P_n$ belong to $\oplus \mathbb{P}$, they can be computed by Boolean circuits of size $O(\tilde{n}^k)$ with $\tilde{n} = n \log n$ (by our hypothesis on circuits size for $\oplus \mathbb{P}$ languages and the fact that the function $\phi$ takes $n \log d$ bits).

These Boolean circuits can in turn be simulated by (Boolean) sums of arithmetic circuits of size and formal degree $O(\tilde{n}^k)$ by the usual method (see e.g. the proof of Valiant’s criterion in [4]).

Hence we have written $P_n = \sum_{\tilde{m}} \psi(\tilde{m})X^{\tilde{m}}$, i.e. $P_n$ is a sum over $O(\tilde{n}^k)$ variables in $\mathbb{F}_2$ of an arithmetic circuit $\psi$ of size $O(\tilde{n}^k)$, and the degree of $\psi$ is $O(\tilde{n}^k)$. By Lemma [8] $P_n$ is a projection of HC$_{\mathbb{F}_2, O(k^2)}$. By hypothesis, the uniform family (HC$_n$) has $O(n^k)$ arithmetic circuits. Hence, $(P_n)$ has arithmetic circuits of size $n^{O(k^2)}$. □

Acknowledgements

We thank Guillaume Malod for useful discussions (in particular on Lemma [8] and Thomas Colcombet for some advice on the presentation.

References

[1] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. SIAM J. Comput., 38(5):1987–2006, 2009.

[2] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. Structural complexity. II, volume 22 of EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1990.

[3] Walter Baur and Volker Strassen. The complexity of partial derivatives. Theor. Comput. Sci., 22:317–330, 1983.

[4] Peter Bürgisser. Completeness and reduction in algebraic complexity theory, volume 7 of Algorithms and Computation in Mathematics. Springer-Verlag, Berlin, 2000.

[5] Lance Fortnow, Rahul Santhanam, and Ryan Williams. Fixed-polynomial size circuit bounds. In IEEE Conference on Computational Complexity, pages 19–26, 2009.
[6] Lane A. Hemaspaandra and Mitsunori Ogihara. *The complexity theory companion*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2002.

[7] Pavel Hrubes and Amir Yehudayoff. Arithmetic complexity in ring extensions. *Theory of Computing*, 7(1):119–129, 2011.

[8] Maurice J. Jansen and Rahul Santhanam. Stronger lower bounds and randomness-hardness trade-offs using associated algebraic complexity classes. In *STACS*, pages 519–530, 2012.

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

[10] Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1-3):40–56, 1982.

[11] Pascal Koiran. Hilbert’s Nullstellensatz is in the polynomial hierarchy. *J. Complexity*, 12(4):273–286, 1996.

[12] Pascal Koiran. Hilbert’s Nullstellensatz is in the polynomial hierarchy. Technical Report 96-27, DIMACS, July 1996.

[13] Richard J. Lipton. Polynomials with 0-1 coefficients that are hard to evaluate. In *FOCS*, pages 6–10, 1975.

[14] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *FOCS*, pages 2–10, 1990.

[15] Guillaume Malod. *Polynômes et coefficients*. PhD thesis, Université Claude Bernard Lyon 1, 2003. http://tel.archives-ouvertes.fr/tel-00087399.

[16] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010.

[17] Rahul Santhanam. Circuit lower bounds for merlin–arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.

[18] Claus-Peter Schnorr. Improved lower bounds on the number of multiplications/divisions which are necessary to evaluate polynomials. *Theor. Comput. Sci.*, 7:251–261, 1978.

[19] Larry J. Stockmeyer. On approximation algorithms for #p. *SIAM J. Comput.*, 14(4):849–861, 1985.

[20] Volker Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM J. Comput.*, 3(2):128–149, 1974.

[21] Leslie G. Valiant. Completeness classes in algebra. In *STOC*, pages 249–261, 1979.

[22] N. V. Vinodchandran. A note on the circuit complexity of PP. *Theor. Comput. Sci.*, 347(1-2):415–418, 2005.