Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 36 (2019)

On the complexity of computing a random Boolean function over the reals

Pavel Hrubeš*

August 12, 2020

Abstract

We say that a first-order formula $A(x_1, \ldots, x_n)$ over $\mathbb{R}$ defines a Boolean function $f : \{0,1\}^n \to \{0,1\}$, if for every $x_1, \ldots, x_n \in \{0,1\}$, $A(x_1, \ldots, x_n)$ is true iff $f(x_1, \ldots, x_n) = 1$. We show that:

(i) every f can be defined by a formula of size $O(n)$,

(ii) if A is required to have at most $k \geq 1$ quantifier alternations, there exists an f which requires a formula of size $2^{\Omega(n/k)}$.

The latter result implies several previously known as well as some new lower bounds in computational complexity. We note that (i) holds over any field of characteristic zero, and (ii) holds for any real closed or algebraically closed field.

1 Introduction

In computational complexity, we are typically interested in computing a Boolean function $f : \{0,1\}^n \to \{0,1\}$. The major computational model is a *Boolean circuit* which obtains the function by means of the elementary operations $\wedge, \vee, \neg$. The major open problem is to prove super-polynomial (or even super-linear) lower bounds on the circuit size of an explicit function f. On the other hand, it is easy to prove, non-constructively, that hard Boolean functions exist: comparing the number of circuits of a given size with the total number of functions, there must exist Boolean functions which require circuits of exponential size.

The counting argument relies on the fact that the elementary operations used are functions over a small finite set. In the complexity literature, we also encounter algebraic models of computation which do not have this property. While we are still interested in computing a Boolean function, we are allowed to use intermediary operations over an infinite domain – typically the real numbers or some other infinite field. To give a simple example: suppose we want to obtain f by computing a real polynomial g by means of an *arithmetic circuit*

*Institute of Mathematics of ASCR, Prague, pahrubes@gmail.com. Supported by GACR grant 19-05497S.

ISSN 1433-8092

(see [20, 11] for details) such that $f(x) = g(x)$ holds over $x \in \{0,1\}^n$. Since an arithmetic circuit can use arbitrary real numbers as constants, we can no longer apply the counting argument in this case. A similar phenomenon occurs in the case of *span programs* [13, 2], and others.

A well-known strategy is to replace the counting argument with Warren's theorem [22], or some variant of it [17, 1] (see also Section 5). Warren's theorem tells us how many sign patterns can be achieved in the image of a polynomial map, which is quite enough to prove the existence of hard functions in the aforementioned models [11, 2, 17]. There is however at least one instance where this tool is apparently insufficient. Suppose we want to compute f by means of a parametrized linear program as follows: we have a system $L(x, y)$ of linear inequalities over $\mathbb{R}$ in the variables $x = \langle x_1, \dots, x_n \rangle$ and $y = \langle y_1, \dots, y_m \rangle$. We require that for every $x \in \{0,1\}^n$, $f(x) = 1$ iff the system $L(x, y)$ has a solution $y \in \mathbb{R}^m$. Is there a function f such that f requires an exponential number of inequalities to be defined this way? This measure, which we call *linear separation complexity*, has been considered at least in [23, 15] and arises in the context of the so-called extension complexity of polytopes (see Section 3 for details). The author does not know how to resolve this question directly using Warren's theorem. Neither he knows how to extend the closely related result of Rothvoß [18] to this situation.

We can view these algebraic models a bit more abstractly. Consider a Boolean function defined by a first-order formula over the reals $A(x_1, \dots, x_n)$. The function accepts on $x_1, \dots, x_n \in \{0, 1\}$ iff $A(x_1, \dots, x_n)$ is true. Here, the formula A may contain constant symbols representing arbitrary real numbers as well as quantifiers over $\mathbb{R}$. In all the above examples, we are in fact defining f in terms of an existentially quantified formula over the reals (or another underlying field). Are there functions which are hard for this model? As we will see, this depends on whether we bound the quantifier complexity of A or not. First, if no restriction is imposed, then every Boolean function can be defined by a linear size formula. Second, if A is required to have at most $k \geq 1$ quantifier alternations in the prenex form then there is a Boolean function requiring a formula of size $2^{\Omega(n/k)}$. The latter implies an exponential lower bound on the linear separation complexity as well as the other models discussed. Our first result is achieved by a direct construction, the second one is a corollary of known results on quantifier elimination over the reals. In this respect, our question is closely related to the problem whether $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$ in the real Turing machine model (see [4] and [14] for survey). We will see that both results hold in a greater generality, in other fields besides the reals.

2 Preliminaries

Let $\mathbb{F}$ be a field. An $\mathbb{F}$-formula, or simply a formula, is a first-order formula built from the function and predicate symbols "$+, \cdot, =$", constant symbols c_a for every element a of the field, as well as the usual logical symbols (variables, Boolean connectives, and quantifiers $\exists, \forall$). If $\mathbb{F}$ is an *ordered* field, we allow also the predicate symbols $<, \leq$ representing the ordering.[1] We define the size of a formula as the number of symbols in the formula (constants and variables having a unit cost). Every formula with no free variables is either true or false, under the intended interpretation of symbols as operations over $\mathbb{F}$.

Every quantifier-free formula over a field is of the form $B(t_1 = t'_1, \ldots, t_m = t'_m)$, where B is a propositional formula defining a Boolean function and t_i, t'_i are terms defining polynomials with coefficients from $\mathbb{F}$. Over an ordered field, we may also encounter the atomic formulas $t_i < t'_i$, $t_i \leq t'_i$. We will take the liberty to identify the constant c_a with a and, occasionally, identify terms with the polynomials they represent. A Σ_1-formula is a formula of the form $\exists x_1 \ldots \exists x_n A$, where A is quantifier-free (aka Σ_0-formula). Similarly, Σ_2-formula is of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m A$, and so on: Σ_{k+2}-formula is of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m A$ with A being Σ_k. Every formula can be converted to an equivalent Σ_k-formula of nearly the same size, for some k. One could also define Π_k-formulas, but we have no need for that.

Let $\mathbb{F}$ be a field or an ordered field. Let $A(x_1, \ldots, x_n)$ be an $\mathbb{F}$-formula with no other free variables other than $x_1, \ldots, x_n$. We will say that A *defines* a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if the following holds:

$$f(\sigma_1, \ldots, \sigma_n) = 1 \text{ iff } A(\sigma_1, \ldots, \sigma_n) \text{ is true, for every } \sigma_1, \ldots, \sigma_n \in \{0,1\}\,.$$

In Sections 4 and 5, we will prove the following main results:

Theorem 1. *Let $\mathbb{F}$ be a field of characteristic zero. Given an n-variate Boolean function f and $1 \leq k \leq n$, f can be defined by a Σ_{2k-1}-formula of size $O(k2^{n/k})$.*

Theorem 2. *Let $\mathbb{F}$ be either an ordered real closed field or an algebraically closed field. Then for every $k > 0$ and n, there exists a Boolean function f in n variables such that every Σ_k-formula defining f must have size at least $2^{\Omega(n/k)}$.*

Setting $k = n$, Theorem 1 implies that every n-variate Boolean function can be defined by a formula of linear size. We emphasize that Theorem 1 is possible, and Theorem 2 non-trivial, only due to the fact that we allow arbitrary constants from $\mathbb{F}$ to appear in the formula defining f. Let us also note that Theorem 2 requires some assumption on the underlying field: remarkably, it is false over the field of rationals. This follows from the fact, proved by Robinson [16], that integers can be defined inside $\mathbb{Q}$ and that, over $\mathbb{Z}$, the truth-table of a function can be encoded as a single integer (cf. the proof of Theorem 1).

The power of Σ_1-formulas

We note that already the class of Σ_1-formulas is quite robust. That is, many syntactic restrictions or relaxations of the definition lead essentially to the same class. Recall that a Σ_1-formula is of the form $\exists_{y \in \mathbb{F}^r} B(t_1 = t'_1, \ldots, t_m = t'_m)$, where B is a Boolean formula and t_i, t'_i are terms. The latter can be seen as

[1] The potential error resulting from forgetting the order in an ordered field would be small: $x \leq y$ can be defined as $\exists u(y = x + u^2)$.

the so-called *arithmetic formulas* defining polynomials over $\mathbb{F}$. Note that if we allow B to be a Boolean *circuit* instead, we do not get a stronger model: introducing new variables representing the gates of the circuit we can rewrite B as a Σ_1-formula of a linear size. The same applies if we allow the terms t_i, t_i' to be computed by arithmetic circuits. In fact, all polynomial-time computations in the sense of [4] can be expressed as small Σ_1-formulas. In turn, every Σ_1-formula $A(x_1, \dots, x_n)$ of size s can equivalently written as $\exists y_1 \dots \exists y_m (h_1 = 0 \wedge \dots \wedge h_t = 0)$, where $m, t \le O(s)$, and $h_1, \dots, h_t$ are polynomials of degree two. This is true both in an ordered and an unordered field. In the ordered case, this can furthermore be written as $\exists y_1 \dots \exists y_m (h = 0)$, where h is a single polynomial of degree four. That is, the complexity of a Σ_1-formula can be captured as the number of bound variables in an expression involving only low-degree polynomials. This would allow us to redefine Σ_1-complexity in a mathematically cleaner way.

3 An application: extension and separation complexity

As mentioned in the introduction, Theorem 2 has several obvious applications, and we focus on just one. Suppose we want to compute a Boolean function $f(x)$, $x \in \{0,1\}^n$, by the following parametrized linear program. We have $y = \langle y_1, \dots, y_m \rangle$ new variables and a set $L(x, y)$ of linear inequalities or equalities over $\mathbb{R}$:

$$\ell_1(x, y) \ge a_1, \dots, \ell_r(x, y) \ge a_r\,,\ u_1(x, y) = b_1, \dots, u_t(x, y) = b_t\,.$$

We say that $L(x, y)$ computes f, if for every $x \in \{0,1\}^n$,

$$f(x) = 1 \ \text{iff there exists } y \in \mathbb{R}^m \text{ such that } L(x, y) \text{ is satisfied}\,. \tag{1}$$

In other words, f accepts precisely on the Boolean inputs

$$\{x \in \{0,1\}^n : \exists y \in \mathbb{R}^m\ Ax + By \ge a,\ Cx + Dy = b\}\,,$$

where A, B, C, D, a, b are real matrices and vectors describing the linear system. We define the *linear separation complexity of* f as the smallest r so that f can be computed as in (1) by a linear system with r inequalities. Note that we disregard m, the number of extra variables, as well as t, the number of equalities, in the definition. This is because both these parameters can be bounded in terms of n and r.

The geometric interpretation is as follows. A polyhedron $P \subseteq \mathbb{R}^n$ will be called a *separating polyhedron for* f, if

$$f^{-1}(1) \subseteq P\,,\ f^{-1}(0) \cap P = \emptyset\,,$$

i.e., the polyhedron contains all accepting inputs of f and excludes all its rejecting inputs. Following [23, 18, 8], define the *extension complexity of* P as the smallest r such that P is a linear projection of a polyhedron $Q \subseteq \mathbb{R}^m$ where Q can be defined using r inequalities (and any number of equalities). In this language, the linear separation complexity of f equals the smallest r such that there exists a separating polyhedron for f of extension complexity r.

While the phrase "linear separation complexity" is introduced here, the same concept has appeared earlier. Already in [21], Valiant has observed that linear separation complexity is, up to a constant factor, a lower bound on the Boolean circuit complexity of f. This appears again in the seminal paper of Yannanakis [23]. A similar quantity was also investigated by Pudlák and Oliveira in [15] in the context of proof complexity. The Yannanakis' paper started a fruitful direction of research into the extension complexity of 0/1-polytopes. Rothvoß [18] has shown that there exists a polytope $P \subseteq \mathbb{R}^n$ with vertices in $\{0,1\}^n$ and extension complexity $2^{\Omega(n)}$. Since then, the same was proved for explicit polytopes (see, e.g., [19] and references within).

In our setting, the smallest separating polyhedron for f is simply the convex hull of accepting inputs of f, $P_0 = \mathrm{conv}(f^{-1}(1))$. Hence, the result [18] says that there exists an f such that P_0 has exponential extension complexity. This however does not imply a lower bound on the linear separation complexity, for there are infinitely many other separating polytopes. Furthermore, it is not apparent to the author how to adapt Rothvoß' proof to this setting. On the other hand, Theorem 2 readily implies:

Theorem 3. *For every n, there exists a Boolean function $f : \{0,1\}^n \to \{0,1\}$ with linear separation complexity $2^{\Omega(n)}$.*

Proof. Assume that f can be computed by a linear system $L(x,y)$ as in (1). It is easy to see that the number of extra variables y can be bounded by r and the number of equalities by n. Hence, f can be defined by a Σ_1-formula of size $O((r+n)^2)$. By Theorem 2, this means that $r \geq 2^{\Omega(n)}$ for some f. □

Theorem 3 also implies the result in [18]. However, Rothvoß' proof achieves better constants hidden in $\Omega(n)$ and is definitely more informative. The reasoning of Theorem 3 could also be applied to "semi-definite separation complexity" as considered in [5].

4 Proof of Theorem 1

We now show that quantifier alternations allow to efficiently define every Boolean function f. The idea is to encode the truth table of f as a natural number, a_f, so that the values of f can be efficiently recovered from a_f. The main ingredient is to show that over the field, we can argue about integers of doubly exponential size. This part is reminiscent of the construction in [10, 7].

Let $\mathbb{F}$ be a field of characteristic zero. We identify a natural number n with the finite sum $1 + \cdots + 1$ of length n. A formula will be called *constant-free*, if it contains only the constants 0, 1 and -1.

Lemma 4. *Given integers n and $1 \le k \le n$, there exists a constant-free Σ_{2k-1}-formula $A_{n,k}(x)$ of size $O(k2^{n/k})$ such that $A_{n,k}(x)$ defines the set of integers $\{0, 1, \ldots, 2^{2^n} - 1\}$.*

Proof. We will first construct the formula using auxiliary constants, $\tau_0, \tau_1, \ldots,$ where $\tau_i := 2^{2^i}$. These will be eliminated later. Given N a power of two and $\ell \ge 1$, we start by giving a $\Sigma_{2\ell-1}$-formula $B_{N,\ell}(x)$ of size $O(\ell N)$ defining the set $\{0, 1, \ldots, 2^{N^\ell} - 1\}$. Note that for an integer $m \ge 0$, the function

$$g_m(x_0, \ldots, x_{N-1}) = x_0 + mx_1 + \cdots + m^{N-1}x_{N-1}$$

is a bijection between $\{0, 1, \ldots, m-1\}^N$ and $\{0, 1, \ldots, m^N - 1\}$. We can set

$$B_{N,1}(x) := \exists x_0, \ldots, x_{N-1} \left(x = g_2(x_0, \ldots, x_{N-1}) \wedge \bigwedge_{i=0}^{N-1} (x_i = 0 \vee x_i = 1) \right) .$$

Given $B_{N,\ell}$ defining $\{0, 1, \ldots, 2^{N^\ell} - 1\}$, the formula

$$\begin{aligned} &B_{N,\ell+1}(x) := \\ &\qquad \exists x_0, \ldots, x_{N-1} \left(x = g_{2^{N^\ell}}(x_0, \ldots, x_{N-1}) \wedge \forall z ((\bigvee_{i=0}^{N-1} z = x_i) \to B_{n,\ell}(z)) \right) \end{aligned}$$

defines the set $\{0, 1, \ldots, 2^{N^{\ell+1}} - 1\}$. Moreover, we can write $x = g_m(x_0, \ldots, x_{N-1})$ as

$$x = x_0 + m(x_1 + m(x_2 + \ldots) \ldots) ,$$

which allows to express $x = g_{2^{N^\ell}}(x_0, \ldots, x_{N-1})$ as a formula of size $O(N)$ using only the constant $2^{N^\ell} = \tau_{\log_2 N\ell}$. Applying this recursively gives the required $B_{N,\ell}$ formula: its size is $O(\ell N)$ and, converted to prenex form, it is a $\Sigma_{2\ell-1}$-formula.

If k divides n, we can take $B_{2^{n/k},k}$ as the formula $A_{n,k}$. In general, let

$$A_{n,k}(x) := \exists u \left(x + u + 1 = 2^{2^n} \wedge B_{2^{\lceil n/k \rceil},k}(u) \wedge B_{2^{\lceil n/k \rceil},k}(x) \right) .$$

It remains to eliminate the constants τ_i. To this end, view them as free variables and let $T_{n'}$ be the conjunction of the equations

$$\tau_0 = 2, \tau_1 = \tau_0^2, \ldots, \tau_{n'} = \tau_{n'-1}^2 .$$

These equations have $2^{2^0}, \ldots, 2^{2^{n'}}$ as their only solution. Furthermore, $A_{n,k}$ has used the constants τ_i with $i = n$ or $i \le \lceil n/k \rceil (k-1) \le 2n$. Hence,

$$\exists \tau_0, \ldots, \tau_{2n} (T_{2n} \wedge A_{n,k}(x))$$

is a constant-free Σ_{2k-1}-formula defining $\{0, 1, \ldots, 2^{2^n} - 1\}$. The size of the formula is $O(n + k2^{n/k})$ which can be written as $O(k2^{n/k})$. $\square$

The following is a stronger version of Theorem 1.

Theorem 5. *Let $\mathbb{F}$ be a field of characteristic zero. For every n and $1 \leq k \leq n$, there exists a constant-free Σ_{2k-1}-formula $B(x_1, \dots, x_n, y)$ of size $O(k2^{n/k})$ such that the following holds. For every $f : \{0,1\}^n \to \{0,1\}$ there exists $a_f \in \mathbb{F}$ such that $B(x_1, \dots, x_n, a_f)$ defines the function f.*

Proof. For $x = \langle x_1, \dots, x_n \rangle \in \{0,1\}^n$, let $b(x) := \sum_{i=1}^{n} 2^{i-1} x_i$. Given $f : \{0,1\}^n \to \{0,1\}$, let

$$a_f := \sum_{x \in \{0,1\}^n} f(x) 2^{b(x)} .$$

In other words, a_f is the integer such that for every x, the $b(x)$-th bit of a_f is $f(x)$. Note that a_f lies in $\{0, 1, \dots, 2^{2^n} - 1\}$. Furthermore, $f(x) = 1$ if and only if

$$\exists y_1, y_2 \in \{0, 1, \dots, 2^{2^n} - 1\}\,,\ y_1 < 2^{b(x)},\ a_f = 2^{b(x)+1} y_2 + 2^{b(x)} + y_1 \,. \qquad (2)$$

Using the previous lemma, the conditions $y_1, y_2 \in \{0, 1, \dots, 2^{2^n} - 1\}$ can be replaced by $A_{n,k}(y_1), A_{n,k}(y_2)$. Also, the ordering $y_1 < z$ on $\{0, 1, \dots, 2^{2^n} - 1\}$ can be defined as $\exists u (z = y_1 + u + 1 \wedge A_{n,k}(u))$. Finally,

$$2^{b(x)} = 2^{\sum_{i=1}^{n} 2^{i-1} x_i} = \prod_{i=1}^{n} 2^{2^{i-1} x_i} = \prod_{i=1}^{n} (x_i (2^{2^{i-1}} - 1) + 1) \,.$$

This allows to write $2^{b(x)}$ and $2^{b(x)+1} = 2 \cdot 2^{b(x)}$ as O(n)-size terms using the auxiliary constants $\tau_i = 2^{2^i}$, $i \leq n - 1$. As noted in the proof of the previous lemma, the constants can be defined by the formula T_{n-1}. Altogether, condition (2) can be written as a Σ_{2k-1}-formula of size $O(n + k2^{n/k})$, which in turn simplifies to $O(k2^{n/k})$. $\square$

Let us remark that in the definition of constant-free formula, one can insist that the formula contains no constants at all: this is because $0, 1$ and -1 can be defined by such a formula. Furthermore, in the proof of Theorem 5, we did not use the fact that $\mathbb{F}$ is a field. It would be quite enough to assume that $\mathbb{F}$ is a ring or even a semiring with multiplicative unit 1 such that the "natural numbers" $1, 1+1, 1+1+1, \dots$ are distinct.

5 Proof of Theorem 2

Our proof of Theorem 2 uses tools from algebraic geometry, namely, counting the number of sign patterns of a polynomial map and quantifier elimination. The author would be happy to see a more direct and self-contained proof at least for the case of Σ_1-formulas. We first overview the results required.

Sign patterns of a polynomial map

For $b \in \mathbb{R}$, let

$$\mathrm{sgn}(b) = \begin{cases} 1\,, & b > 0 \\ 0\,, & b = 0 \\ -1\,, & b < 0 \end{cases}$$

Let $f = \langle f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\rangle$ be a sequence of real polynomials of degree at most d. For $a \in \mathbb{R}^n$, let $\mathrm{sgn} f(a) := \langle \mathrm{sgn} f_1(a), \ldots, \mathrm{sgn} f_m(a)\rangle \in \{-1, 0, 1\}^m$, be the *sign-pattern of f at a*. Warren [22] has obtained a bound on the number of sign-patterns of f lying in $\{-1, 1\}^m$; as noted by Alon [1], a similar bound applies to the total number of sign-patterns. Assuming $2m \geq n$ and $d \geq 1$, the number of sign patterns can be bounded as

$$|\{\mathrm{sgn} f(a) : a \in \mathbb{R}^n\}| \leq (8\mathrm{e}dm/n)^n\,. \tag{3}$$

The same estimate clearly holds over any real closed field[2].

Over unordered fields, a similar bound holds on the number of *zero patterns*. For $b \in \mathbb{F}$, let

$$\mathrm{sgn}^*(b) := \begin{cases} 1\,, & b \neq 0 \\ 0\,, & b = 0 \end{cases}$$

For $a \in \mathbb{F}^n$, let $\mathrm{sgn}^* f(a) := \langle \mathrm{sgn}^* f_1(a), \ldots, \mathrm{sgn}^* f_m(a)\rangle \in \{0, 1\}^m$, be the *zero-pattern of f at a*. A bound on the number of zero-patterns of f has been obtained by Heintz [10], and the estimates were recently improved and simplified by Rónyai et al. in [17]. The number of zero-patterns can be bounded by (assuming $d \geq 1$, $m \geq n$)

$$|\{\mathrm{sgn}^* f(a) : a \in \mathbb{F}^n\}| \leq (\mathrm{e}dm/n)^n\,.$$

Quantifier elimination

The celebrated Tarski-Seidenberg theorem asserts that every formula over a real closed field is equivalent to a quantifier-free formula. We are interested in the size of the resulting formula. It is known ([10, 7]) that in general, the size can increase doubly-exponentially if we allow a linear number of quantifier alternations. The situation is better if the number of quantifier alternations is small. The result of Grigoriev [9] (see also [3], Chapter 14, Theorem 14.16) implies the following: every Σ_k-formula A of size s is equivalent to a quantifier-free formula of size $2^{s^{O(k)}}$. More specifically, A can be written as

$$G(\mathrm{sgn}(f_1) = \sigma_1, \ldots, \mathrm{sgn}(f_m) = \sigma_m)\,, \tag{4}$$

where $f_1, \ldots, f_m$ are polynomials in the free variables of A, $\sigma_1, \ldots, \sigma_m \in \{-1, 0, 1\}$ and $G : \{0, 1\}^m \to \{0, 1\}$ is a Boolean function. Moreover, the degrees of f_i, formula size of G, and the parameter m, can all be bounded by $2^{s^{O(k)}}$.

[2] Hence, also any ordered field

The same result holds over any algebraically closed field, as shown by Chistov and Grigoriev [6] (see also Corollary 6.4 in [12]). The expression (4) is replaced by $G(f_1 = 0, \ldots, f_m = 0)$.

Let us remark that the cited bounds are more informative than presented here: they bound the number of f_i's in (4) and their degree separately, in terms of the number of atomic formulas in A, their degrees, and the number of quantifier alternations. Moreover, the constants in the big-O are different in the two cases (algebraically closed versus real closed field).

We now proceed to prove Theorem 2. At a high level, we use quantifier elimination to reduce to the quantifier-free case, and apply Warren's theorem to atoms of the quantifier-free formula.

For a formula A with no free variables, let $[A] \in \{0,1\}$ denote its truth-value. Let

$$\beta = \langle \beta_1(y_1, \ldots, y_n), \ldots, \beta_m(y_1, \ldots, y_n) \rangle \tag{5}$$

be a sequence of formulas with all their free variables among $y_1, \ldots, y_n$. For $a \in \mathbb{F}^n$,

$$[\beta(a)] := \langle [\beta_1(a)], \ldots, [\beta_m(a)] \rangle \in \{0,1\}^m$$

will be called the *truth-pattern of β at a*. We want to bound the number of truth-patterns of β in terms of its complexity,

Lemma 6. *Let $\mathbb{F}$ be an algebraically closed or an ordered real closed field. Let β as in (5) be a sequence of Σ_k-formulas, each of size at most s. Then the number of truth-patterns can be bounded as $|\{[\beta(a)] : a \in \mathbb{F}^n\}| \leq (2^{s^{O(k)}} m)^n$.*

Proof. We focus on the real closed case, the argument is the same for algebraically closed field. The bounds on quantifier elimination in (4) imply the following. Given β_i, there exists a sequence $f_i = \langle f_{i,1}, \ldots, f_{i,m_i} \rangle$ of polynomials in the variables $y_1, \ldots, y_n$ such that the truth value of $\beta_i(a)$, $a \in \mathbb{F}^n$, is determined by the sign-pattern of f_i at a. Moreover, m_i as well as the degrees of $f_{i,j}$ are bounded by $2^{s^{O(k)}}$. Let f be a sequence of all the polynomials $f_{i,j}$, $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, m_i\}$. The length of the sequence is $M \leq m 2^{s^{O(k)}}$ and each polynomial has degree $d \leq 2^{s^{O(k)}}$. Given $a \in \mathbb{F}^n$, the truth-pattern of β at a is determined be the sign-pattern of f at a, and hence the number of truth-patterns is at most the number of sign patterns of f. Using (3), the latter can be bounded by $(8edM)^n$ which can be written[3] as $(2^{s^{O(k)}} m)^n$. $\square$

Proof of Theorem 2. Assume that $s \geq n$ is such that every Boolean function in n variables can be defined by a Σ_k-formula of size at most s. Let $\mathcal{F}$ be the set of such formulas with free variables among $x_1, \ldots, x_n$. Introduce fresh variables $y = \langle y_1, \ldots, y_s \rangle$ and $z = \langle z_1, \ldots, z_s \rangle$. A formula $S(x, y)$, with $x = \langle x_1, \ldots, x_n \rangle$, will be called a *skeleton* if a) it contains only variables from x, y, z and no constant symbols, and b) its free variables are from x or y. We think of y as representing constants from $\mathbb{F}$ and z as names of bound variables. Let $\mathcal{S}$ be the

[3] As $s \geq 2$, the additional constants can be swallowed by the big-O.

set of Σ_k-skeletons of size at most s. Hence, for every $A(x) \in \mathcal{F}$ there exists $S(x, y) \in \mathcal{S}$ and $a \in \mathbb{F}^s$ such that $A(x) = S(x, a)$ (up to renaming of the bound variables z). Unlike $\mathcal{F}$, $\mathcal{S}$ is a finite set. A skeleton is a string of symbols from the alphabet $x, y, z, \forall, \exists, \wedge, \dots$ of size $O(s)$. Therefore,

$$|\mathcal{S}| \leq 2^{O(s \log s)} .$$

We will say that a skeleton $S(x, y)$ defines a Boolean function $f : \{0,1\}^n \to \{0,1\}$, if there exists $a \in \mathbb{F}^s$ such that $S(x, a)$ defines f. Hence, every f is defined by some skeleton in $\mathcal{S}$. We now want to bound the number of functions defined by a given skeleton $S(x, y) \in \mathcal{S}$. Let β be the sequence of the 2^n formulas $S(\sigma, y)$, $\sigma \in \{0,1\}^n$. Each formula in β has free variables in y. For a given $a \in \mathbb{F}^s$, the function defined by $S(x, a)$ is uniquely determined by the truth-pattern of β at a: indeed, $S(x, a)$ defines the function f such that $f(\sigma) = [S(\sigma, a)]$ for all σ. Hence, the number of functions defined by $S(x, y)$ is at most the number of truth-patterns of β. By the previous lemma, this can be bounded by $(2^{s^{O(k)}} 2^n)^s$ which is of the form $2^{s^{O(k)}}$ (we assumed $s \geq n$).

Altogether, skeletons in $\mathcal{S}$ can define at most $2^{O(s \log s)} 2^{s^{O(k)}}$ Boolean functions. Since the total number of functions is 2^{2^n}, we must have $s \geq 2^{\Omega(n/k)}$. $\square$

Acknowledgement The author thanks Pavel Pudlák and James Lee for useful discussions, and Benjamin Rossman for suggesting improvements to the paper.

References

[1] Noga Alon: Tools from higher algebra. In *Handbook of Combinatorics*. Elsevier and MIT Press, 1995.

[2] László Babai, Anna Gál, and Avi Wigderson: Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

[3] Saugata Basu, Richard Pollack, and Marie-Francoise Roy: *Algorithms in real algebraic geometry.* Springer-Verlag, 2006.

[4] Lenore Blum, Filipe Cucker, Michael Shub, and Steve Smale: *Complexity and real computation.* Springer-Verlag, 1998.

[5] Jop Briet, Daniel Dadush, and Sebastian Pokutta: On the existence of 0/1 polytopes with high semidefinite extension complexity. *J. Mathematical Programming*, 153(1):179–199, 2015.

[6] Alexander L. Chistov and Dima Grigoriev: Complexity of quantifier elimination in the theory of algebraically closed fields. In *Mathematical Foundations of Computer Science*, pp. 17–31, 1984.

[7] JAMES H. DAVENPORT AND JOOS HEINTZ: Real quantifier elimination is doubly exponential. *J. Symbolic Computation*, 5(29–35), 1988.

[8] SAMUEL FIORINI, SERGE MASSAR, SEBASTIAN POKUTTA, HANS RAJ TIWARY, AND RONALD DE WOLF: Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *CoRR*, abs/1111.0837, 2011.

[9] DIMA GRIGORIEV: Complexity of deciding Tarski algebra. *J. Symbolic Computation*, 5(1–2):1988, 1988.

[10] JOOS HEINTZ: Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 26:239–277, 1983.

[11] PAVEL HRUBEŠ AND AMIR YEHUDAYOFF: Arithmetic complexity in ring extensions. *Theory of Computing*, 7:119–129, 2011.

[12] DOUGLAS JOHN IERARDI: *The complexity of quantifier elimination in the theory of an algebraically closed field.* Ph. D. thesis, Cornell University, 1989.

[13] MAURICIO KARCHMER AND AVI WIGDERSON: On span programs. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference*, pp. 102–111, 1993.

[14] PASCAL KOIRAN: Circuits versus trees in algebraic complexity. In *STACS*, pp. 35–52, 2000.

[15] PAVEL PUDLÁK AND MATEUS DE OLIVEIRA OLIVEIRA: Representations of monotone Boolean functions by linear programs. In *Proceedings of the 32nd Computational Complexity Conference*, 2017.

[16] JULIA ROBINSON: Definability and decision problems in arithmetic. *J. Symb. Log.*, 14(2):98–114, 1949.

[17] LAJOS RÓNYAI, LÁSZLÓ BABAI, AND MURALI K. GANAPATHY: On the number of zero-patterns of a sequence of polynomials. *J. Amer. Math. Soc.*, 14(3):717–735, 2001.

[18] THOMAS ROTHVOSS: Some 0/1 polytopes need exponential size extended formulations. *CoRR*, abs/1105.0036, 2011.

[19] THOMAS ROTHVOSS: The matching polytope has exponential extension complexity. *J. of the ACM*, 64(6), 2017.

[20] AMIR SHPILKA AND AMIR YEHUDAYOFF: Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3):207–388, 2010.

[21] LESLIE G. VALIANT: Reducibility by algebraic projections. *Enseign. Math.*, 28:253–268, 1982.

[22] Hugh E. Warren: Lower bounds for approximations by nonlinear manifolds. *Trans. AMS*, 133:167–178, 1968.

[23] Mihalis Yannakakis: Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il