THEORY OF COMPUTING, Volume 16 (9), 2020, pp. 1–12
www.theoryofcomputing.org

On the Complexity of Computing a Random Boolean Function Over the Reals

Pavel Hrubeš*

Received January 26, 2019; Revised July 10, 2020; Published October 21, 2020

Abstract. We say that a first-order formula $A(x_1,\dots,x_n)$ over $\mathbb{R}$ defines a Boolean function $f:\{0,1\}^n \to \{0,1\}$, if for every $x_1,\dots,x_n \in \{0,1\}$, $A(x_1,\dots,x_n)$ is true iff $f(x_1,\dots,x_n) = 1$. 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: the nonconstructive lower bound on span programs of Babai, Gál, and Wigderson (Combinatorica 1999); Rothvoß's result (CoRR 2011) that there exist $0/1$ polytopes that require exponential-size linear extended formulations; a similar lower bound by Briët et al. (Math. Program. 2015) on semidefinite extended formulations; and a new result stating that a random Boolean function has exponential linear separation complexity. We note that (i) holds over any field of characteristic zero, and (ii) holds over any real closed or algebraically closed field.

ACM Classification: F.1.3, F.2.3

AMS Classification: 68Q17, 03C10

Key words and phrases: complexity theory, lower bounds, average case, formula complexity, random Boolean function, quantifier elimination

*Supported by GACR grant 19-05497S.

 DOI: 10.4086/toc.2020.v016.a009

1 Introduction

In computational complexity, we are typically interested in computing a Boolean function $f : \{0,1\}^n \to \{0,1\}$. The central computational model is a *Boolean circuit* which computes 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 intermediate 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* (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. Nor does he know 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 $A(x_1,\dots,x_n)$ over the reals. 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. Are there functions which are hard for this model? As we will see, this depends on whether we bound the quantifier complexity of A. 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 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 $+, \times, =$ 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 $B(t_1 = t'_1, \dots, t_m = t'_m)$, 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 $\exists x_1 \dots \exists x_n A$, where A is quantifier-free (a. k. a. Σ_0-formula). Similarly, a Σ_2-formula is of the form $\exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_m A$, and so on: a Σ_{k+2}-formula is of the form $\exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_m A$ 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 $A(x_1, \dots, x_n)$ be an $\mathbb{F}$-formula with no free variables other than $x_1, \dots, x_n$. We will say that A *defines* a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if the following holds:

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

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

Theorem 2.1. *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 2.2. *Let $\mathbb{F}$ be either a 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 2.1 implies that every n-variate Boolean function can be defined by a formula of linear size. We emphasize that Theorem 2.1 is possible, and Theorem 2.2 is 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 2.2 requires some assumption on the underlying field. Remarkably, it is false over the field of rationals.

Observation 2.3. Over $\mathbb{Q}$ (as an unordered field), every Boolean function in n variables can be defined by a Σ_4-formula of size $O(n)$.

Proof sketch. This relies on a beautiful result of Julia Robinson [16] who showed that integers can be defined inside $\mathbb{Q}$ by a single first-order formula. The same applies to non-negative integers. Working over $\mathbb{N}$, a Boolean function can be defined by a Σ_1-formula of linear size. This is because the truth-table of f

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

can be encoded by a single natural number from which values of f can be recovered by a Σ_1-formula—cf., the proof of Theorem 4.2. This gives a linear-size Σ_k-formula over $\mathbb{Q}$ for some fixed k. The more accurate bound $k = 4$ is achieved by inspecting the quantifier complexity of Robinson's formula. □

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 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 $A(x_1, \ldots, x_n)$ of size s can equivalently be written as $\exists y_1 \ldots \exists y_m (h_1 = 0 \wedge \cdots \wedge h_t = 0)$, where $m, t \le O(s)$, and $h_1, \ldots, h_t$ 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 $\exists y_1 \ldots \exists y_m (h = 0)$, 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 allows us to redefine Σ_1-complexity in a mathematically cleaner way.

3 An application: extension and separation complexity

As mentioned in the introduction, Theorem 2.2 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 $y = \langle y_1, \ldots, y_m \rangle$ new variables and a set $L(x,y)$ of linear inequalities or equalities over $\mathbb{R}$:

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

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

$$f(x) = 1 \text{ iff there exists } y \in \mathbb{R}^m \text{ such that } L(x,y) \text{ is satisfied}. \tag{3.1}$$

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 (3.1) 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 Yannakakis [23]. A similar quantity was also investigated by Pudlák and Oliveira in [15] in the context of proof complexity. Yannanakis's 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., [8, 19] and references in the latter).

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 extponential 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's proof to this setting. On the other hand, Theorem 2.2 readily implies the following result.

Theorem 3.1. *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 (3.1). 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 2.2, this means that $r \geq 2^{\Omega(n)}$ for some f. □

Theorem 3.1 also implies the result in [18]. However, Rothvoß's proof achieves better constants hidden in $\Omega(n)$ and is definitely more informative. The reasoning of Theorem 3.1 could also be applied to "semidefinite separation complexity" as considered in [5].

4 Proof of Theorem 2.1

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 4.1. *Given integers n and $1 \leq k \leq n$, 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 $\{0,1,\ldots,2^{2^n}-1\}$.*

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 2 and $\ell \geq 1$, 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 $m \geq 0$, 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

$$B_{N,\ell+1}(x) := \exists x_0,\ldots,x_{N-1}\left(x = g_{2^{N^\ell}}(x_0,\ldots,x_{N-1}) \wedge \forall z((z = x_0 \vee \cdots \vee z = x_{N-1}) \to B_{n,\ell}(z))\right)$$

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 us to express $x = g_{2^{N^\ell}}(x_0,\ldots,x_{N-1})$ as a formula of size $O(N)$ using only the constant $2^{N^\ell} = \tau_{\ell \log_2 N}$. 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 $i \leq \lceil n/k\rceil(k-1) \leq 2n$. 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 $O(k2^{n/k})$. □

The following is a stronger version of Theorem 2.1.

Theorem 4.2. *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 $B(x_1,\ldots,x_n,y)$ 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 $B(x_1,\ldots,x_n,a_f)$ defines the function f.*

Proof. For $x = \langle x_1, \ldots, x_n \rangle \in \{0,1\}^n$, 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 $\{0,1,\ldots,2^{2^n}-1\}$. Furthermore, $f(x) = 1$ if and only if

$$\exists y_1, y_2 \in \{0,1,\ldots,2^{2^n}-1\}\,,\, y_1 < 2^{b(x)},\ a_f = 2^{b(x)+1} y_2 + 2^{b(x)} + y_1 . \tag{4.1}$$

Using the previous lemma, the conditions $y_1, y_2 \in \{0,1,\ldots,2^{2^n}-1\}$ can be replaced by $A_{n,k}(y_1), A_{n,k}(y_2)$. Also, the ordering $y_1 < z$ on $\{0,1,\ldots,2^{2^n}-1\}$ 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 us to write $2^{b(x)}$ and $2^{b(x)+1} = 2 \cdot 2^{b(x)}$ as O(n)-size terms using the auxiliary constants $\tau_i = 2^{2^i}$, $i \le n-1$. As noted in the proof of the previous lemma, the constants can be defined by the formula T_{n-1}. Altogether, condition (4.1) can be written as a Σ_{2k-1}-formula of size $O(n + k2^{n/k})$, which in turn simplifies to $O(k2^{n/k})$. □

Let us remark that in the definition of a constant-free formula, one can insist that the formula contain no constants at all: this is because $0, 1$ and -1 can be defined by such a formula. Furthermore, in the proof of Theorem 4.2, 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" $1, 1+1, 1+1+1, \ldots$ are distinct.

5 Proof of Theorem 2.2

Our proof of Theorem 2.2 uses tools from algebraic geometry and real algebraic geometry, namely, counting the sign-patterns or zero-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 give an overview of 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

$$|\{\mathrm{sgn} f(a) : a \in \mathbb{R}^n\}| \leq (8\mathrm{e}dm/n)^n \,. \tag{5.1}$$

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

A similar bound holds for the number of *zero-patterns* over *every field* $\mathbb{F}$.

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 $\mathrm{sgn}^* f(a) := \langle \mathrm{sgn}^* f_1(a), \dots, \mathrm{sgn}^* f_m(a) \rangle \in \{0,1\}^m$, be the *zero-pattern of f at a*. A bound on the number of zero-patterns of f has been obtained by Heintz [10]. An elementary linear algebra proof of improved estimates was found more recently by Rónyai et al. in [17]. By [17], the number of zero-patterns can be bounded (assuming $d \geq 1$, $m \geq n$) by

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

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 double-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

$$G(\mathrm{sgn}(f_1) = \sigma_1, \dots, \mathrm{sgn}(f_m) = \sigma_m)\,, \tag{5.2}$$

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

A similar result holds over any algebraically closed field, as shown by Chistov and Grigoriev [6] (see also Corollary 6.4 in [12]). Every Σ_k-formula A of size s is equivalent to a quantifier-free formula of the form

$$G(f_1 = 0, \dots, f_m = 0)\,,$$

where m, the degrees of the f_i, and the formula size of G, can again be bounded by $2^{s^{O(k)}}$.

Let us remark that the cited papers contain more detailed information than presented here: they bound the number of the f_i in (5.2) and their degrees 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).

[2] Hence, also any ordered field

We now proceed to the proof of Theorem 2.2. 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

$$\beta = \langle \beta_1(y_1,\dots,y_n),\dots,\beta_m(y_1,\dots,y_n)\rangle \tag{5.3}$$

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

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

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 5.1. *Let $\mathbb{F}$ be an algebraically closed field or a real closed field. Let β as in (5.3) 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. First let us consider the real closed case. The bounds on quantifier elimination in (5.2) imply the following. Given β_i, there exists a sequence $f_i = \langle f_{i,1},\dots,f_{i,m_i}\rangle$ of polynomials in the variables $y_1,\dots,y_n$ 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 $f_{i,j}$, $i \in \{1,\dots,m\}$, $j \in \{1,\dots,m_i\}$. The length of the sequence is $M \leq m2^{s^{O(k)}}$ 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 (5.1), the latter can be bounded by $(8edM)^n$ which can be written[3] as $(2^{s^{O(k)}}m)^n$.

This completes the proof for the real closed case. In the case of algebraically closed fields the argument is the same, replacing sign-patterns by zero-patterns and using the algebraically closed version of quantifier elimination. □

Proof of Theorem 2.2. 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 $x_1,\dots,x_n$. Introduce fresh variables $y = \langle y_1,\dots,y_s\rangle$ and $z = \langle z_1,\dots,z_s\rangle$. A formula $S(x,y)$, with $x = \langle x_1,\dots,x_n\rangle$, 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 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

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

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 $s \geq 2^{\Omega(n/k)}$. □

6 Open problems

The proof of Theorem 2.2 uses machinery from algebraic geometry and real algebraic geometry as a black box. As a consequence, it teaches us little about the nature of the problem. For example, the aforementioned result of Rothvoss [18] is formally a consequence of Theorem 2.2. But his proof provides an additional insight into the geometry of polytopes which is completely lacking in our setting.

Problem 6.1. Find a more direct and self-contained proof of Theorem 2.2, at least for Σ_1-formulas.

This would be interesting especially over the reals. Already the case of linear separation complexity from Section 3, where the Σ_1-formula contains only linear inequalities or equalities, seems to require a new insight. On the other hand, the first ingredient of the current proof of Theorem 2.2 over algebraically closed fields, counting zero-patterns of a polynomial map, has been given a simple proof in [17]. Hence, the question may be easier in the algebraically closed setting.

Second, the upper and lower bounds given by Theorem 2.1 and 2.2 do not match exactly (they hardly can, for the constants are hidden in Theorem 2.2). Theorem 2.1 implies that every Boolean function can be defined by a Σ_3-formula of size $O(2^{n/2})$. But for Σ_1- or Σ_2-formulas, we are left with the trivial upper bound of $2^{n(1-o(1))}$. Whether we can improve the constant in the exponent is intriguing already in the case of Σ_1-formulas.

Problem 6.2. Over $\mathbb{R}$, can every $f : \{0,1\}^n \to \{0,1\}$ be defined by a Σ_1-formula of size $O(2^{(1-\varepsilon)n})$ for some $0 < \varepsilon < 1$?

Acknowledgement The author thanks Pavel Pudlák and James Lee for useful discussions, and the editors László Babai and Benjamin Rossman for considerably improving to the paper.

References

[1] NOGA ALON: Tools from higher algebra. In *Handbook of Combinatorics*, volume 2, pp. 1749–1783. MIT Press, 1996. Link at ACM DL. 2, 7

[2] LÁSZLÓ BABAI, ANNA GÁL, AND AVI WIGDERSON: Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999. [doi:10.1007/s004930050058] 2

[3] SAUGATA BASU, RICHARD POLLACK, AND MARIE-FRANÇOISE ROY: *Algorithms in Real Algebraic Geometry*. Springer, 2006. [doi:10.1007/3-540-33099-2] 8

[4] LENORE BLUM, FILIPE CUCKER, MICHAEL SHUB, AND STEVE SMALE: *Complexity and Real Computation*. Springer, 1998. [doi:10.1007/978-1-4612-0701-6] 2, 4

[5] JOP BRIËT, DANIEL DADUSH, AND SEBASTIAN POKUTTA: On the existence of 0/1 polytopes with high semidefinite extension complexity. *Math. Program.*, 153(1):179–199, 2015. Preliminary version in ESA'13. [doi:10.1007/s10107-014-0785-x, arXiv:1305.3268] 5

[6] ALEXANDER L. CHISTOV AND DIMA GRIGORIEV: Complexity of quantifier elimination in the theory of algebraically closed fields. In *Proc. 11th Internat. Symp. Math. Found. Comput. Sci. (MFCS'84)*, pp. 17–31. Springer, 1984. [doi:10.1007/BFb0030287] 8

[7] JAMES H. DAVENPORT AND JOOS HEINTZ: Real quantifier elimination is doubly exponential. *J. Symbolic Comput.*, 5(1–2):29–35, 1988. [doi:10.1016/S0747-7171(88)80004-X] 5, 8

[8] SAMUEL FIORINI, SERGE MASSAR, SEBASTIAN POKUTTA, HANS RAJ TIWARY, AND RONALD DE WOLF: Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2/17):95–106, 2015. Preliminary version in STOC'12. [doi:10.1145/2716307] 5

[9] DIMA GRIGORIEV: Complexity of deciding Tarski algebra. *J. Symbolic Comput.*, 5(1–2):65–108, 1988. [doi:10.1016/S0747-7171(88)80006-3] 8

[10] JOOS HEINTZ: Definability and fast quantifier elimination in algebraically closed fields. *Theoret. Comput. Sci.*, 24(3):239–277, 1983. [doi:10.1016/0304-3975(83)90002-6] 5, 8

[11] PAVEL HRUBEŠ AND AMIR YEHUDAYOFF: Arithmetic complexity in ring extensions. *Theory of Computing*, 7(8):119–129, 2011. [doi:10.4086/toc.2011.v007a008] 2

[12] DOUGLAS JOHN IERARDI: *The Complexity of Quantifier Elimination in the Theory of an Algebraically Closed Field*. Ph. D. thesis, Cornell University, 1989. Conference version in STOC'89. 8

[13] MAURICIO KARCHMER AND AVI WIGDERSON: On span programs. In *Proc. of 8th IEEE Structure in Complexity Theory Conf. (SCT'93)*, pp. 102–111. IEEE Comp. Soc. Press, 1993. [doi:10.1109/SCT.1993.336536] 2

[14] PASCAL KOIRAN: Circuits versus trees in algebraic complexity. In *Proc. 17th Symp. Theoretical Aspects of Comp. Sci. (STACS'00)*, pp. 35–52. Springer, 2000. [doi:10.1007/3-540-46541-3_3] 2

[15] PAVEL PUDLÁK AND MATEUS DE OLIVEIRA OLIVEIRA: Representations of monotone Boolean functions by linear programs. *ACM Trans. Comput. Theory*, 11(4):22:1–22:31, 2019. Preliminary version in CCC'17. [doi:10.1145/3337787, ECCC:TR17-106] 2, 5

[16] JULIA ROBINSON: Definability and decision problems in arithmetic. *J. Symbolic Logic*, 14(2):98–114, 1949. [doi:10.2307/2266510] 3

[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. [doi:10.1090/S0894-0347-01-00367-8] 2, 8, 10

[18] THOMAS ROTHVOSS: Some 0/1 polytopes need exponential size extended formulations. *Math. Program.*, 142(1):255–268, 2013. [doi:10.1007/s10107-012-0574-3, arXiv:1105.0036] 2, 5, 10

[19] THOMAS ROTHVOSS: The matching polytope has exponential extension complexity. *J. ACM*, 64(6):41:1–41:19, 2017. Preliminary version in STOC'14. [doi:10.1145/3127497, arXiv:1311.2369] 5

[20] AMIR SHPILKA AND AMIR YEHUDAYOFF: Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3–4):207–388, 2010. [doi:10.1561/0400000039] 2

[21] LESLIE G. VALIANT: Reducibility by Algebraic Projections. In *Logic and Algorithmic*, volume 30 of *Monographs of L'Enseignement Mathématique*, pp. 365–380. 1982. 5

[22] HUGH E. WARREN: Lower bounds for approximations by nonlinear manifolds. *Trans. Amer. Math. Soc.*, 133(1):167–178, 1968. [doi:10.2307/1994937] 2, 7

[23] MIHALIS YANNAKAKIS: Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.*, 43(3):441–466, 1991. Preliminary version in STOC'88. [doi:10.1016/0022-0000(91)90024-Y] 2, 5

AUTHOR

Pavel Hrubeš
Researcher
Institute of Mathematics of ASCR
Prague, Czechia
hrubes@math.cas.cz
http://www.math.cas.cz/homepage/main_page.php?id_membre=189

ABOUT THE AUTHOR

PAVEL HRUBEŠ graduated from Charles University in Prague in 2008 under the supervision of Pavel Pudlák.

After a period of joyful postdocs in the U. S. and Canada, he became a mature worker in the factory of academia. He contributes mainly to arithmetic circuit complexity and proof complexity. His hobbies include mountaineering and brooding.