Lower Bounds for Monotone Counting Circuits[☆]

Stasys Jukna[1]

Institute of Computer Science, Goethe University, Frankfurt am Main, Germany

Abstract

A monotone arithmetic circuit *computes* a given multivariate polynomial f if its values on all nonnegative integer inputs are the same as those of f. The circuit *counts* f if this holds for 0-1 inputs; on other inputs, the circuit may output arbitrary values. The circuit *decides* f if it has the same 0-1 roots as f. We first show that some multilinear polynomials can be exponentially easier to count than to compute them, and that some polynomials can be exponentially easier to decide than to count them. Our main results are general lower bounds on the size of counting circuits.

Key words: Arithmetic circuits, boolean circuits, counting complexity, lower bounds

1. Introduction

In this paper we consider computational complexity of multivariate polynomials with nonnegative integer coefficients:

$$f(x_1,\ldots,x_n)=\sum_{e\in E}c_e\prod_{i=1}^{n}x_i^{e_i}\,, \tag{1}$$

where $E\subset\mathbb{N}^n$ is a finite set of vectors of nonnegative integers, coefficients c_e are positive integers, and $x_i^0=1$; here and throughout, $\mathbb{N}=\{0,1,2,\ldots\}$. Each coefficient c_e stands for the number of times the *monomial* $p=\prod_{i=1}^n x_i^{e_i}$ appears in f; the *support* of such a monomial is the set $X_p=\{x_i\colon e_i\neq 0\}$ of variables appearing in it with nonzero exponents, and the *degree* of the monomial p is the sum $e_1+\cdots+e_n$ of its exponents. The polynomial is *multilinear* if $E\subseteq\{0,1\}^n$, and is *homogeneous* of degree d if all its monomials have the same degree d.

A natural model for compact representation of such polynomials (with nonnegative coefficients) is that of monotone arithmetic $(+,\times)$ circuits. Such a circuit is a directed acyclic graph with three types of nodes: input, addition $(+)$, and multiplication $(\times)$. Input nodes have fanin zero, and correspond to variables $x_1,\ldots,x_n$. All other nodes have fanin two, and are called *gates.* Each gate computes either the sum or product of its inputs. The *size* of a circuit is the number of gates in it.

Every such circuit syntactically *produces* a unique polynomial h with nonnegative integer coefficients in a natural way: the polynomial produced at an input gate x_i consists of a single monomial x_i, and the polynomial produced at a sum (product) gate is the sum (product) of polynomials produced at its inputs; we use distributivity to write a product of polynomials as a sum of monomials. The polynomial h produced by the circuit itself is the polynomial produced at its output gate. Given a polynomial $f(x_1,\ldots,x_n)$, we say that the circuit:

- *computes* f (exactly) if $h(a)=f(a)$ holds for every $a\in\mathbb{N}^n$;

[☆] Research supported by the DFG grant SCHN 503/6-1.

Email addresses: jukna@thi.informatik.uni-frankfurt.de (Stasys Jukna)

URL: http://www.thi.informatik.uni-frankfurt.de/ jukna/ (Stasys Jukna)

[1] Affiliated with Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania.

- *counts* f if $h(a) = f(a)$ holds for every $a \in \{0,1\}^n$;
- *decides* f if for every $a \in \{0,1\}^n$, $h(a) = 0$ exactly when $f(a) = 0$.

In this paper we are mainly interested in $(+,\times)$ circuits *counting* a given polynomial f. Such a circuit needs only to correctly compute the restriction $f : \{0,1\}^n \to \mathbb{N}$ of f on 0-1 inputs. Note that, if the polynomial f is monic (has no coefficients > 1) then, on every 0-1 input $a \in \{0,1\}^n$, the value $f(a)$ taken by f on a is the number of monomials of f satisfied by (evaluated to 1 on) a. For example, in the case of the *permanent polynomial*

$$\mathrm{Per}_n(x) = \sum_{\sigma} \prod_{i=1}^{n} x_{i,\sigma(i)} \tag{2}$$

with the summation over all permutations σ of $[n] = \{1,\ldots,n\}$, its value $\mathrm{Per}_n(a)$ on every input $a \in \{0,1\}^{n\times n}$ is the number of perfect matchings in the bipartite $n \times n$ graph G_a specified by a; nodes i and j are adjacent in G_a if and only if $a_{ij} = 1$. Thus, a circuit counting Per outputs the number of perfect matchings in G_a, whereas a circuit deciding this polynomial merely tells us whether G_a contains a perfect matching. On the other hand, *computing* circuits must actually solve the same counting problem but in the case when all nonnegative integers (not just 0 and 1) are allowed as weights.

Remark 1. Let us stress that we only consider *monotone* arithmetic circuits. The reason is that counting $(+,-,\times)$ circuits are already omnipotent: they are as powerful as boolean $\{\vee,\wedge,\neg\}$ circuits, for which no super-linear lower bounds are known so far. This holds because then each of the three boolean operations can be simulated over $\{0,1\}$: $x \wedge y$ by $x \times y$, $\neg x$ by $1 - x$, and $x \vee y$ by $x + y - xy$.

If a $(+,\times)$ circuit computes, counts or only decides a given polynomial f, what can then be said about the structure of the *produced* by the circuit polynomial h? To answer these questions, we associate with every polynomial f the following three sets (this notation will be used throughout the paper):

- $\mathcal{M}(f)$ is the set of all monomials of f;
- $\mathcal{S}(f) = \{X_p : p \in \mathcal{M}(f)\}$ is the *support* of f;
- $\mathcal{L}(f) \subseteq \mathcal{S}(f)$ is the *lower support* of f consisting of all minimal sets of $\mathcal{S}(f)$; a set of a family of sets is *minimal* if it contains no other set of the family.

We have the following information about the structure of the produced by a circuit polynomial h (see Lemma 6). If the circuit:

- computes f then $h = f$, and hence, also $\mathcal{M}(h) = \mathcal{M}(f)$;
- counts f then $\mathcal{S}(h) = \mathcal{S}(f)$;
- decides f then $\mathcal{L}(h) = \mathcal{L}(f)$.

Thus, in the case of circuits exactly *computing* f we have a full knowledge about the produced by the circuit polynomial h: this polynomial must just coincide with f (the same monomials with the same coefficients). This ensures that no "invalid" monomials can be formed during the computation, and severely limits the power of such circuits. In particular, if the target polynomial f is homogeneous (all monomials have the same degree) then the circuit *itself* must be homogeneous: polynomials produced at its gates must be also homogeneous. If the target polynomial f is multilinear (no variable has degree larger than 1) then the circuit must be also multilinear: the polynomials produced at inputs of each product gate must depend on disjoint sets of variables. These limitations were essentially exploited in all known proofs of lower bounds for monotone arithmetic circuits, including [15, 17, 10, 21, 18, 6, 19, 7].

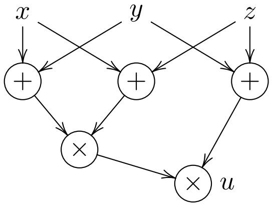


Figure 1: A circuit of size 5 *computes* the polynomial $F = (x + y)(y + z)(x + z)$, *counts* the polynomial $f = 2xyz + 2xy + 2xz + 2yz$, and decides the polynomial $g = xy + xz + yz$. Gate u is the output gate.

In the case of *counting* circuits, $\mathcal{M}(h) = \mathcal{M}(f)$ needs not to hold, due to the multiplicative idempotence axiom $x^2 = x$ valid on 0-1 inputs. That is, here exponents (and hence, degrees of monomials) do not matter (see Fig. 1). For example, a polynomial $f = 2x + yz$ is counted by any circuit producing a polynomial of the form $h = 2x^a + y^b z^c$ with $a, b, c \in \mathbb{N} \setminus \{0\}$. That is, nonzero exponents of the monomials in produced by counting circuits polynomials may be arbitrary: we only know which sets of variables these monomials must contain, but we do not know their actual degrees. In *deciding* circuits, even $\mathcal{S}(h) = \mathcal{S}(f)$ needs not to hold, due to the additional absorption axiom $x + xy = x$.

Due to the limitations we mentioned above, lower bounds for $(+, \times)$ circuits *computing* a given polynomial are relatively easy to obtain So, it is natural to ask whether known lower bounds for exactly computing $(+, \times)$ circuits can be extended to counting circuits?

That they sometimes *can* be extended was demonstrated by Sengupta and Venkateswaran in [16], where they show that the proof of an optimal lower bound $n2^{n-1}$ on the $(+, \times)$ circuit complexity of the permanent polynomial given by Jerrum and Snir [10], can be modified to yield the same lower bound for circuits only counting this polynomial. Still, at least three questions remained open:

1. Can counting circuits be substantially smaller than computing circuits?
2. Can deciding circuits be substantially smaller than counting circuits?
3. Can lower-bounds *arguments* for computing $(+, \times)$ circuits—not just bounds for specific polynomials, like the permanent polynomial—be extended to counting circuits?

In this paper, we answer these questions affirmatively. For this, we consider the following three basic complexity measures of a given polynomial f in the class of monotone arithmetic $(+, \times)$ circuits:

A(f) (*arithmetic* complexity) = minimum size of a circuit computing f;

C(f) (*counting* complexity) = minimum size of a circuit counting f;

D(f) (*decision* or *boolean* complexity) = minimum size of a circuit deciding f.

Note that we always have

$$\mathrm{D}(f) \leq \mathrm{C}(f) \leq \mathrm{A}(f)\,. \tag{3}$$

That is, it is never harder to decide a given polynomial than to count it, and it is never harder to count a polynomial than to compute it (exactly).

2. Main results

We will first show that both gaps in (3) can be exponentially large, and then give two general combinatorial lower bounds on the counting complexity C(f).

2.1. Complexity gaps

Our first result is that the gaps in (3) between arithmetic, counting and deciding complexities of some polynomial can be exponential. Actually, we will show even stronger gaps: it may be exponentially harder to count so-called lower and higher "envelopes" of a polynomial than to count the polynomial itself.

The *lower envelope* of a polynomial f is a homogeneous polynomial f_{le} consisting of the monomials of f of smallest degree. The *higher envelope* f_{he} is defined by taking monomials of largest degree.

As observed by Jerrum and Snir [10], every $(+,\times)$ circuit producing a polynomial f can be easily transformed into a circuit producing f_{le} or f_{he} by just removing (if necessary) one of the two edges entering some of the sum-gates. This observation yields

$$\mathrm{A}(f) \geq \max\{\mathrm{A}(f_{\mathrm{he}}), \mathrm{A}(f_{\mathrm{le}})\} \ . \tag{4}$$

Thus, lower and higher envelopes of a polynomial f are not harder to *compute* than to compute the polynomial itself. However, the following result shows that the situation with the *counting* complexity $\mathrm{C}(f)$ is completely different.

Theorem 1.

(i) *It can be exponentially harder to compute a polynomial than to count it.*

(ii) *It can be exponentially harder to count a polynomial than to decide it.*

(iii) *It can be exponentially harder to count lower or higher envelopes of a polynomial than to count the polynomial itself.*

Note that, together with (4), item (iii) already implies item (i). We stated these claims separately only to stress that both gaps in (3) can be exponential.

2.2. Lower bounds

If a $(+,\times)$ circuit *counts* a given polynomial $f(x_1,\ldots,x_n)$ then the produced by this circuit polynomial h must have the same family of supports, that is, $\mathcal{S}(h)=\mathcal{S}(f)$ must hold (Lemma 6 in Sect. 4.1).

Recall that the *support* of a monomial is the set of variables appearing in it with nonzero exponents. The *support* $\mathcal{S}(f)$ of a polynomial is the family of supports of all its monomials.

Let now g and h be two polynomials on the same set of variables as f. Let $\mathcal{A}=\mathcal{S}(f)$, $\mathcal{B}=\mathcal{S}(g)$ and $\mathcal{C}=\mathcal{S}(h)$ be the corresponding supports (families of subsets of these variables). If $f=g+h$ is the sum of these two polynomials then

$$\mathcal{A}=\mathcal{B}\cup\mathcal{C} := \{A\colon A\in\mathcal{B} \ \text{ or } \ A\in\mathcal{C}\}$$

is just the set-theoretic union of the corresponding two families. If $f=g\times h$ is the product then

$$\mathcal{A}=\mathcal{B}\vee\mathcal{C} := \{B\cup C\colon B\in\mathcal{B} \text{ and } C\in\mathcal{C}\}$$

is the *cross-union* of these families. Thus, when dealing with supports only, every gate in a $(+,\times)$ circuit is performing either a union or a cross-union operation on the supports of produced polynomials. It is clear that we always have $|\mathcal{B}\cup\mathcal{C}|\leq|\mathcal{B}|+|\mathcal{C}|$ and $|\mathcal{B}\vee\mathcal{C}|\leq|\mathcal{B}|\cdot|\mathcal{C}|$. In this context, it is perhaps worth to mentioning that the well-known Four Function Theorem of Ahlswede and Daykin [1] implies that $|\mathcal{B}|\cdot|\mathcal{C}|\leq|\mathcal{B}\vee\mathcal{C}|\cdot|\mathcal{B}\wedge\mathcal{C}|$, where $\mathcal{B}\wedge\mathcal{C}=\{B\cap C\colon B\in\mathcal{B} \text{ and } C\in\mathcal{C}\}$ is the cross-intersection.

For a nonnegative real number r, define the *r-th degree* $d_r(\mathcal{A})$ of a family $\mathcal{A}$ to be the maximal possible number of sets in $\mathcal{A}$ containing a fixed set with r or more elements. In other words, $d_r(\mathcal{A})$ is the maximal possible number of sets in $\mathcal{A}$ whose intersection has r or more elements. In particular, if $\mathcal{A}$ is a graph (viewed as a set of its edges) then $d_1(\mathcal{A})$ is the maximum degree of this graph; hence, the term "degree".

Theorem 2. *Let $f = g + h$ be a polynomial such that every monomial of g has fewer than $m/3$ variables and every monomial of h has at least $m \geq 2$ variables. Let $\mathcal{A} = \mathcal{S}(h)$ be the support of h. Then there is an integer r between $m/3$ and $2m/3$ for which*

$$\mathrm{C}(f) \geq \frac{|\mathcal{A}|}{d_r(\mathcal{A}) \cdot d_{m-r}(\mathcal{A})} .$$

In the case when the polynomial f is homogeneous (all monomials have the same degree), this fact was first proved by Hyafil [9], and a different and much simpler proof was found by Valiant [21]. Various versions of this fact (but also for homogeneous polynomials) were proved by other authors, including Jerrum and Snir [10] (implicitly), Raz and Yehudayoff [13], Hrubes and Yehudayoff [8]. Besides that now we have such a lower bound for *counting* circuits, the main difference of Theorem 2 from the previous versions is that now the polynomial needs not to be homogeneous.

One of the first general lower bounds on the (monotone) *arithmetic* circuit complexity of polynomials is due to Schnorr [15]. It states that

$$\mathrm{A}(f) \geq |\mathcal{M}(f)|$$

holds for every polynomial f which is *separated* in the following sense: the product of no two monomials of f can contain any third monomial of f as a factor; as before, $\mathcal{M}(f)$ stands for the set of all monomials of f.

By using different arguments, Gashkov and Sergeev [6, 7] extended this lower bound to a properly larger class of polynomials. Namely, they proved that

$$\mathrm{A}(f) \geq \frac{|\mathcal{M}(f)|}{\max\{k^3, l^2\}}$$

holds for every (k, l)-sparse polynomial. A polynomial f is (k, l)-*sparse* $(1 \leq k \leq l)$ if $\mathcal{M}(f)$ cannot contain all monomials of some product $g \times h$ of two polynomials such that g has more than k and h has more than l monomials. It is easy to see that every separated polynomial f is also $(1, 1)$-sparse. Indeed, if f is not $(1, 1)$-sparse then there are monomials $p \neq p'$ and $q \neq q'$ such that all four monomials of the polynomial $(p + p') \times (q + q')$ are monomials of f. But then the product $pq \times p'q'$ of two monomials of f contains a distinct monomial pq' of f as a factor, meaning that f is not separated.

Using a yet another and simpler argument, we extend these lower bounds to *counting* $(+, \times)$ circuits.

Call a family $\mathcal{A}$ of sets (k, l)-*free* if for every two antichains $\mathcal{B}$ and $\mathcal{C}$ such that $\mathcal{B} \vee \mathcal{C} \subseteq \mathcal{A}$, at least one of $|\mathcal{B}| \leq k$ or $|\mathcal{C}| \leq l$ must hold. Recall that a family of sets is an *antichain* if no two of its sets are comparable under set-inclusion. A family is l-*free* if it is (l, l)-free. A polynomial f is (k, l)-*free* if such is its support $\mathcal{A} = \mathcal{S}(f)$.

Remark 2. To demonstrate the definition, let us show that every 1-free antichain $\mathcal{A}$ must be also *union-free* in the sense that the union of no two distinct sets of $\mathcal{A}$ can be a member of $\mathcal{A}$. To see this, assume that $\mathcal{A}$ contains two sets $A \neq B$ whose union $A \cup B$ also belongs to $\mathcal{A}$. Then the cross-union $\mathcal{B} \vee \mathcal{B} = \{A, B, A \cup B\}$ of the antichain $\mathcal{B} = \{A, B\}$ with itself lies in $\mathcal{A}$, but $|\mathcal{B}| = 2 > 1$, implying that $\mathcal{A}$ is not 1-free.

Recall that the *lower support* $\mathcal{L}(f)$ consists of all minimal sets in the family $\mathcal{S}(f)$.

Theorem 3. *If a polynomial f is (k, l)-free for some $1 \leq k \leq l$ then*

$$\mathrm{C}(f) \geq \frac{|\mathcal{L}(f)|}{2kl^2} .$$

Remark 3. In all proofs of lower bounds on the arithmetic complexity $\mathrm{A}(f)$, the "monotonicity" of the measure $\mu(f) = |\mathcal{M}(f)|$ was essentially exploited: both $\mu(f) \leq \mu(f + g)$ and $\mu(f) \leq \mu(f \times g)$

then hold. When dealing with the counting complexity $\mathrm{C}(f)$, the corresponding measure is $\mu(f) = |\mathcal{S}(f)|$. But then $\mu(f) \leq \mu(f \times g)$ needs not to hold, due to the idempotent axiom $x^2 = x$ used by such circuits. If, for example, $f = x_1 + x_2 + \cdots + x_n$ and $g = x_1 x_2 \cdots x_n$ then $|\mathcal{S}(f)| = n$ but $|\mathcal{S}(f \times g)| = 1$. So, a care and new arguments are necessary when trying to capture the "progress" made by gates in *counting* circuits.

2.3. Some applications

We can associate with every set H of functions $h : [n] \to [n]$ the following multilinear homogeneous polynomial of degree n on n^2 variables:

$$f_H(x) = \sum_{h \in H} \prod_{i=1}^{n} x_{i,h(i)} \,.$$

Corollary 1. *Let $f = f_h + g$, where H is some set of permutations of $[n]$, and g is any polynomial, none of whose monomials contains $n/3$ or more variables. Then*

$$\mathrm{C}(f) \geq \binom{n}{n/3} \cdot \frac{|H|}{n!} \,.$$

Proof. The polynomial $h = f_H$ has $|H|$ monomials, each specified by a permutation $h \in H$ of $[n]$. If some r variables are fixed, this fixes r values of h. Hence, at most $(n-r)!$ of the permutations can take r pre-described values, implying that $d_r(h) \leq (n-r)!$. Theorem 2 implies that $\mathrm{C}(f)$ is at least $|H|$ divided by the maximum of $r!(n-r)!$ over all $n/3 < r \leq 2n/3$. □

By a t-(v,k,λ) *design* $(2 \leq t \leq k \leq v)$ we will mean a family $\mathcal{A}$ of k-element subsets (called *blocks*) of a fixed v-element set (of *points*) such that no t-element set is contained in more than λ blocks. [2] *Steiner triple systems* are 2-$(v,3,1)$-designs.

Every t-(v,k,λ) design $\mathcal{A}$ defines a multilinear homogeneous polynomial of degree k in v variables in a natural way:

$$f_{\mathcal{A}}(x) = \sum_{A \in \mathcal{A}} \prod_{e \in A} x_e \,.$$

Note that a trivial upper bound on the counting complexity of such a polynomial is $\mathrm{C}(f_{\mathcal{A}}) \leq k|\mathcal{A}|$.

Corollary 2. *Let $\mathcal{A}$ be a t-(v,k,λ) design. If $3t \leq k$ then $\mathrm{C}(f_{\mathcal{A}}) \geq |\mathcal{A}|/\lambda^2$.*

Proof. Since $\mathcal{A}$ is a t-(v,k,λ)-design, we have $d_r(\mathcal{A}) \leq \lambda$ for every $r \geq t$. Since for every $k/3 \leq r \leq 2k/3$, both r and $k-r$ are at least $k/3$, the desired lower bound follows directly from Theorem 2 (applied with $m = k$). □

Example 1. Let n be a prime power, and let the elements of our ground-set be all $v = n^2$ points (i,j) of the grid $\mathrm{GF}(n) \times \mathrm{GF}(n)$. The graph of a polynomial h over $\mathrm{GF}(n)$ is the set of n points $(i, h(i))$ with $i \in \mathrm{GF}(n)$. Let $\mathcal{A} = \mathcal{A}_{n,d}$ be the family of all n^d graphs of polynomials of degree at most $d-1$ over $\mathrm{GF}(n)$. Since no two distinct polynomials of degree at most $d-1$ can coincide on d points, $\mathcal{A}$ is a d-$(n^2,n,1)$ design with $|\mathcal{A}| = n^d$ blocks. Then we have

$$n^d \leq \mathrm{C}(f_{\mathcal{A}}) \leq n^{d+1} \,,$$

where the upper bound is trivial, and the lower bound follows from Corollary 2.

[2] In the standard definition of such a design, we have a stronger requirement that every t-element set must be contained in *exactly* λ blocks.

Thus, for polynomials defined by t-(v, k, λ) designs $\mathcal{A}$, where t is at most one third of the block size k, Theorem 2 can yield almost optimal lower bounds on their counting complexity. If t is larger, then the resulting bound can be pure because then $d_r(\mathcal{A})$ or $d_{k-r}(\mathcal{A})$ may be near to the total number $|\mathcal{A}|$ of blocks in the design.

For example, if $\mathcal{A}$ is a Steiner triple system, then the lower bound on $\mathrm{C}(f_{\mathcal{A}})$ given by Theorem 2 (when applied with $m = 3$) is of the form $|\mathcal{A}|/d_r(\mathcal{F})d_{3-r}(\mathcal{F})$ for some integer $1 \leq r \leq 2$. One of r or $3-r$ must be equal to 1. So, since we always have $d_1(\mathcal{A}) \geq |\mathcal{A}|/v$ (so many blocks can share one point), Theorem 2 can only yield a trivial lower bound $|\mathcal{A}|/d_1(\mathcal{A}) \leq v$ not exceeding the number v of points. For designs with more than v blocks, this bound is rather poor.

Still, Theorem 3 allows one to obtain strong lower bounds also for larger parameters t.

Lemma 4. *Every t-(v, k, λ) design with $2t \leq k$ is l-free for $l = \lambda 2^t$.*

Proof. Let $\mathcal{A}$ be a t-(v, k, λ) design with $t \leq k/2$. Take any two antichains $\mathcal{B}$ and $\mathcal{C}$ whose cross union $\mathcal{B} \vee \mathcal{C}$ is contained in $\mathcal{A}$. Our goal is to show that then either $|\mathcal{B}| \leq l$ or $|\mathcal{C}| \leq l$ must hold.

Since every set of the design has k elements, at least one of the antichains, say $\mathcal{B}$, must contain a set B' with at least $k/2 \geq t$ elements. So fix an arbitrary subset $B \subseteq B'$ with $|B| = t$ elements. For every set $C \in \mathcal{C}$, there can be at most $2^{|B|} = 2^t$ distinct sets $C' \in \mathcal{C}$ such that $B \cup C' = B \cup C$, implying that $|\mathcal{C}| \leq |\{B\} \vee \mathcal{C}| \cdot 2^t$. But all sets of the cross-sum $\{B\} \vee \mathcal{C}$ are sets of the design $\mathcal{A}$ and contain a t-element set B. So, $|\{B\} \vee \mathcal{C}| \leq \lambda$, and the desired upper bound $|\mathcal{C}| \leq \lambda 2^t = l$ follows. □

Together with Lemma 4, Theorem 3 yields the following lower bound which is weaker than that in Corollary 2 but holds for larger parameters t.

Corollary 3. *Let $\mathcal{A}$ be a t-(v, k, λ) design. If $2t \leq k$ then $\mathrm{C}(f_{\mathcal{A}}) \geq |\mathcal{A}|/\lambda^4 2^{4t+1}$.*

Lemma 4 implies that every Steiner triple system, that is, every 2-$(v, 3, 1)$-design is 4-free. By using a bit more detailed argument, one can show that these systems are in fact 1-free.

Lemma 5. *Every Steiner triple system is* 1*-free.*

Hence, the polynomial defined by every Steiner triple system $\mathcal{A}$ has counting complexity at least $|\mathcal{A}|/2$.

Proof. Let $\mathcal{A}$ be a Steiner triple system, and suppose contrariwise that $\mathcal{A}$ is *not* 1-free. Then there must be four sets A, B, C, D such that

(i) sets A and B, as well sets C and D are incomparable under set-inclusion

and all sets of the cross-union $\{A, B\} \vee \{C, D\}$ belong to $\mathcal{A}$. Since $\mathcal{A}$ is a Steiner triple system, this latter condition implies that

(ii) no two distinct sets of the cross-union $\{A, B\} \vee \{C, D\}$ can share more than one element in common.

Case 1: Some of the four sets, say, the set A has three elements. If at least one of the sets C or D, say, the set C has at least two elements then (ii) yields $B \cup C = A \cup C = A$, and hence, also $B \subseteq A$, a contradiction with (i). So, $C = \{c\}$ and $D = \{d\}$ for some $c \neq d \in A$. Since then B must have at least two elements, (ii) implies that $B \cup C = B \cup D$ is the same set of $\mathcal{A}$. But this set shares two elements $c \neq d$ with the member A of $\mathcal{A}$, implying that $B \cup C = A$, and hence also $B \subseteq A$, a contradiction with (i).

Case 2: None of the four sets A, B, C, D has three elements. Then at least one of them must have exactly two elements, say, $A = \{a, b\}$. Since the sets $A \cup C$ and $A \cup D$ of $\mathcal{A}$ share these two elements in common, they must be the same set $F = \{a, b, c\}$ for some $c \notin \{a, b\}$. The sets C and D are incomparable and both must contain the (missing in A) element c. Since none of these two sets can have more than two elements, this implies that each of them must have exactly two elements of F, say, $C = \{a, c\}$ and $D = \{b, c\}$. Since each of the sets $B \cup C$ and $B \cup D$ shares two elements with F, we obtain that $B \cup C = B \cup D = F = \{a, b, c\}$. Since $b \notin C$ and $c \notin D$, the set B must contain both elements a and b of A, a contradiction with A and B being incomparable. □

The *triangle polynomial* t_n has $\binom{n}{2}$ variables, each corresponding to an edge of the complete graph K_n on n nodes. There are $\binom{n}{3}$ monomials, each being the product of all three edges of some triangle in K_n. Thus, for every input 0-1 vector a, the value $t_n(a)$ is the number of triangles in the subgraph G_a of K_n specified by a.

Since every two edges of a triangle determine this triangle, the union of no two triangles can contain a third triangle. This means that the polynomial t_n is separated in Schnorr's [15] sense, and his general lower bound for separated polynomials (mentioned before Theorem 3) yields a lower bound $\mathrm{A}(t_n) \geq \binom{n}{3}$ on the arithmetic complexity of this polynomial.

On the other hand, a special case of the celebrated lower bound of Razborov [14] (see also [3]) on the monotone boolean circuit complexity of the clique function implies a sub-cubic lower bound $\mathrm{C}(t_n) \geq \Omega(n^3/\log^3 n)$ on the counting (and even on decision) complexity of t_n. When combined with Lemma 5, Theorem 3 yields a stronger bound for counting complexity.

Corollary 4. $\mathrm{C}(t_n) \geq \frac{1}{2}\binom{n}{3}$.

Proof. Let $\mathcal{A}$ be the set of all triples of edges forming a triangle in K_n. Thus, we have $v = \binom{n}{2}$ points and $|\mathcal{A}| = \binom{n}{3}$ sets. Since no two triangles can share two edges in common, this family is a Steiner triple system. □

We now turn to the proofs of our main results.

3. Proof of Theorem 1

By the *linearization* of a polynomial f we will mean a multilinear polynomial $\tilde{f}$ obtained from f by removing all exponents larger than 1 from all monomials of f. For example, the linearization of $f = 2xy^2 + 3x^4y^2 + 6y^2z$ is $\tilde{f} = 5xy + 6yz$. It is clear that $\tilde{f}(a) = f(a)$ holds for all $a \in \{0,1\}^n$.

3.1. Counting versus computing

To show that $\mathrm{C}(f_{\mathrm{le}})/\mathrm{C}(f)$, and hence also the gap $\mathrm{A}(f)/\mathrm{C}(f)$ can be exponential, call a subgraph G of $K_{n,n}$ an *almost perfect matching* if every node of G has degree one or two. Consider the following combinatorial counting problem: given a subgraph G of $K_{n,n}$, count the number of almost perfect matchings in G.

To come up with a corresponding to this problem polynomial, let U and V be the parts of $K_{n,n}$, and associate a variable x_{uv} with each its edge (u,v). Every assignment a of 0-1 values to these variables defines a subgraph G_a of $K_{n,n}$ in a natural way. Consider the polynomial

$$I_n(x) = \prod_{u\in U}\prod_{v\in V}\Big(\sum_{j\in V} x_{uj}\Big)\Big(\sum_{i\in U} x_{iv}\Big). \tag{5}$$

The polynomial has n^2 variables. Note that every monomial of this polynomial is obtained as follows: take for each node $u \in U$ exactly one edge x_{uj} incident with u, and then take for each node $v \in V$ exactly one edge x_{iv} incident with v. So, every variable has degree at most 2.

Let f be the linearization of I_n. That is, f is obtained from I_n by removing all nonzero exponents in all monomials. Every monomial of f has degree between n and $2n$, and corresponds to some almost perfect matching in $K_{n,n}$. Thus, on every 0-1 input a, the value $f(a)$ of f on this input is exactly the number of almost perfect matching in G_a. By using a trivial $(+,\times)$ circuit given by (5), our counting problem can be solved using at most $2n^3$ gates. Hence, $\mathrm{C}(f) \leq 2n^3$.

We will now show that "weighted" case of our counting problem requires exponential $(+,\times)$ circuits. For this, it is only enough to observe that the monomials of f of degree n correspond to perfect matchings. Thus, the lower envelope f_{le} of f is just the permanent polynomial, that is, $f_{\mathrm{le}} = \mathrm{Per}_n$. By Corollary 1, we have that $\mathrm{A}(f) \geq \mathrm{C}(f_{\mathrm{le}}) = 2^{\Omega(n)}$, as desired.

3.2. Deciding versus counting

Consider the polynomial $f = g + h$, where $h = \mathrm{Per}_n$ is the permanent polynomial (2) with $n > 3$, and $g = \sum_{i,j\in[n]} x_{ij}$ is the sum of all variables. Every monomial of h has n variables, and every monomial of g has $1 < n/3$ variable; so, we can apply Theorem 2. For every fixed set of r edges in $K_{n,n}$, only $(n-r)!$ of all $n!$ perfect matchings in $K_{n,n}$ can share all these edges. This gives an upper bound $d_r(\mathcal{A}) \leq (n-r)!$ for the support $\mathcal{A} = \mathcal{S}(h)$ of h. By Theorem 2, there is an integer r between $n/3$ and $2n/3$ such that $\mathrm{C}(f) \geq n!/r!(n-r)! = \binom{n}{r} = 2^{\Omega(n)}$.

On the other hand, on every 0-1 input a, we have that $f(a) = 0$ if and only if $g(a) = 0$, because $h(0,\ldots,0) = 0$. Hence, $\mathrm{D}(f) = \mathrm{D}(g) \leq n^2$. □

Remark 4. Another, less artificial polynomial exhibiting such a gap is given in A. Namely, using some known lower bounds on monotone boolean circuit complexity, we show that it is exponentially easier to decide whether an *s*-*t* path exists than to count the number of such paths.

3.3. Counting versus envelope counting

Recall that the *lower envelope* of a polynomial f is a homogeneous polynomial f_{le} consisting of the monomials of f of smallest degree. The *higher envelope* f_{he} is defined by taking monomials of largest degree. We have mentioned (see (4)) that none of these two envelops can be harder to *compute* than the polynomial itself, that is, both $\mathrm{A}(f_{\mathrm{he}})$ and $\mathrm{A}(f_{\mathrm{le}})$ are at most $\mathrm{A}(f)$. Our goal is to show that the situation with *counting* complexity is entirely different.

A polynomial f exhibiting an exponential gap $\mathrm{C}(f_{\mathrm{le}})/\mathrm{C}(f)$ was given in Sect. 3.1. To show that counting *higher* envelopes can be also much harder than counting the polynomials themselves, consider the following polynomial of $n^2 + n$ variables:

$$\mathrm{Per}^*(x,y) = \prod_{i=1}^{n}\sum_{j=1}^{n} x_{ij}y_j\,. \tag{6}$$

The relation to the permanent polynomial Per_n is that the coefficient of the monomial $y_1y_2\cdots y_n$ in $\mathrm{Per}^*(x,y)$ is exactly $\mathrm{Per}_n(x)$.

Let now $f(x,y)$ be the linearization of $\mathrm{Per}^*(x,y)$. That is, $f(x,y)$ is a multilinear polynomial obtained from $\mathrm{Per}^*(x,y)$ by removing all nonzero exponents from all monomials. Every monomial of f has degree (sum of exponents) between $n+1$ and $2n$, and the monomials

$$x_{1,j_1}x_{2,j_2}\cdots x_{n,j_n}y_1y_2\cdots y_n$$

of degree $2n$ with all $j_1,\ldots,j_n$ *distinct* are exactly the monomials of the polynomial

$$h(x,y) = \mathrm{Per}_n(x)\cdot y_1y_2\cdots y_n\,.$$

Thus, $h = f_{\mathrm{he}}$ is the higher envelope of f. Since $h(x,1,\ldots,1) = \mathrm{Per}_n(x)$, Corollary 1 yields $\mathrm{C}(f_{\mathrm{he}}) \geq \mathrm{C}(\mathrm{Per}_n) = 2^{\Omega(n)}$. On the other hand, since exponents play no role on 0-1 inputs, we have that $\mathrm{Per}^*(a) = f(a)$ holds for all 0-1 inputs a. Thus, the polynomial f itself can be counted by the circuit given by the definition (6) of Per^*. This gives the upper bound $\mathrm{C}(f) = O(n^2)$. □

4. Preliminaries

In the proofs of our lower bounds for counting circuits (Theorems 2 and 3), we will need some structural properties of monotone arithmetic circuits.

4.1. Structure of produced polynomials

As we mentioned in the introduction, every $(+,\times)$ circuit syntactically *produces* a unique polynomial h with nonnegative integer coefficients in a natural way: the polynomial produced at an input gate x_i consists of a single monomial x_i, and the polynomial produced at a sum (product) gate is the sum (product) of polynomials produced at its inputs; we use distributivity to write a product of polynomials as a sum of monomials. The polynomial produced by the circuit itself is the polynomial produced at its output gate.

If a $(+,\times)$ circuit computes, counts or decides a given polynomial, what can then be said about the structure of the *produced* by the circuit polynomial?

Recall that the linearization of f is a multilinear polynomial $\tilde{f}$ which is obtained from f by removing all nonzero exponents (the coefficients remain the same). Note that $\tilde{h} = \tilde{f}$ implies $\mathcal{S}(h) = \mathcal{S}(f)$.

Lemma 6. *Let $f(x_1,\ldots,x_n)$ be a polynomial, and $h(x_1,\ldots,x_n)$ be the polynomial produced by some $(+,\times)$ circuit.*

(i) *The circuit computes f if and only if $h = f$.*

(ii) *The circuit counts f if and only if $\tilde{h} = \tilde{f}$, and hence, also $\mathcal{S}(h) = \mathcal{S}(f)$.*

(iii) *The circuit decides f if and only if $\mathcal{L}(h) = \mathcal{L}(f)$.*

Proof. To show item (i), we will use the following very special version of the so-called *Combinatorial Nullstellensatz* of Alon [2]; we also include a very short proof of this special case.

Claim 1. Let $f(x_1,\ldots,x_n)$ be a polynomial in which each variable x_i has degree at most t_i, and let $S_i \subseteq \mathbb{N}$ be arbitrary subsets of sizes $|S_i| \geq t_i + 1$, $i = 1,\ldots,n$. Then f is uniquely determined by its values on $S_1 \times S_2 \times \cdots \times S_n$.

Proof. Induction on n. For $n = 1$, the claim is simply the assertion that a non-zero polynomial of degree t_1 in one variable can have at most t_1 distinct roots. For the induction step, expand the polynomial f by the variable x_n:

$$f(x_1,\ldots,x_n) = \sum_{i=0}^{t_n} f_i(x_1,\ldots,x_{n-1}) \cdot x_n^i\,.$$

For each point $a \in S_1 \times \cdots \times S_{n-1}$,

$$f(a, x_n) = \sum_{i=0}^{t_n} f_i(a) \cdot x_n^i$$

is a polynomial of degree at most t_n in one variable x_n, and hence, all its coefficients $f_i(a)$, $i = 0,1,\ldots,t_n$ can be recovered knowing the values $f(a,b)$ for all $b \in S_n$. Knowing the values $f_i(a)$ for all $a \in S_1 \times \cdots \times S_{n-1}$ we can, by the induction hypothesis, recover the polynomials f_i, and hence, the original polynomial f. □

To prove item (ii), suppose first that $\tilde{h} = \tilde{f}$. Then for every input $a \in \{0,1\}^n$, we have that $h(a) = \tilde{h}(a) = \tilde{f}(a) = f(a)$, meaning that the circuit must count the polynomial f. Suppose now that the circuit counts f. Then $h(a) = f(a)$, and hence, also $\tilde{h}(a) = \tilde{f}(a)$ must hold for all $a \in \{0,1\}^n$. When applied with all $S_i = \{0,1\}$, Claim 1 yields $\tilde{h} = \tilde{f}$, that is, $\tilde{h}$ and $\tilde{f}$ must coincide as polynomials.

To prove item (iii), suppose first that $\mathcal{L}(h) = \mathcal{L}(f)$ and take an input $a \in \{0,1\}^n$. Then $f(a) > 0$ holds if and only if some set of $\mathcal{L}(f)$ is contained in $\{x_i \colon a_i = 1\}$. Hence, if $\mathcal{L}(h) = \mathcal{L}(f)$ then the circuit decides f. Suppose now that the circuit decides f, i.e., that h and f have the same 0-1 roots. Our goal is to show that then $\mathcal{L}(h) = \mathcal{L}(f)$ must hold.

Suppose contrariwise that the polynomial h has some monomial p whose support $X_p = \{x_i \colon e_i \neq 0\}$ belongs to $\mathcal{L}(h)$ but does not belong to $\mathcal{L}(f)$. If $X_q \not\subseteq X_p$ holds for all monomials q of f then we can set all variables in X_p to 1 and the rest to 0. On the resulting assignment a, we will have $f(a) = 0$ but $h(a) \geq p(a) = 1$, a contradiction.

Thus, there must be a monomial q of f such that $X_q \subset X_p$; the inclusion must be proper, because $X_p \not\in \mathcal{L}(f)$. Set now all variables in X_q to 1 and the rest to 0. On the resulting assignment a, we will have $f(a) \geq q(a) = 1$. But since X_p belongs to $\mathcal{L}(h)$, every monomial p of h with $X_{p'} \neq X_p$ must have at least one variable outside X_p, and hence, also outside X_q. This implies that $h(a) = 0$, a contradiction again. □

4.2. Contents of gates and edges

Fix some $(+, \times)$ circuit, and let h be the produced by it polynomial. Recall that $\mathcal{M}(h)$ stands for the set of monomials of h. For a gate v in the circuit, let P_v denote the *set* of monomials of the (syntactically) produced at this gate v polynomial. Hence, if $v = x_i$ is an input gate then $P_v = \{x_i\}$. If $v = u + w$ is a sum gate then $P_v = P_u \cup P_w$ is just a set-theoretic union of the sets of monomials produced at its inputs. If $v = u \times w$ is a product gate then P_v is a *rectangle* (cross-product of two sets of monomials)

$$P_v = P_u \times P_w := \{pq \colon p \in P_u \text{ and } q \in P_w\}\,.$$

If v is the output gate then we have $P_v = \mathcal{M}(h)$. If however v is not the output gate then monomials of P_v need not to be monomials of the entire polynomial h. We only know that every monomial of P_v must be a factor of at least one monomial of h. That is, for every monomial $p \in P_v$, there is a unique monomial q (the *co-factor* of p within h) such that pq is a monomial of h. Define the *complement* of gate v to be the set Q_v of all possible monomials q which are co-factors of *all* monomials in P_v, that is,

$$Q_v = \{q \colon pq \in \mathcal{M}(h) \text{ for all } p \in P_v\}\,.$$

Following Jerrum and Snir [10], define the *content* of the gate v as the rectangle

$$R_v := P_v \times Q_v \subseteq \mathcal{M}(h)\,.$$

Even though the first set P_v is (very naturally) defined by the circuit alone, the second set Q_v is only implicitly defined by sets P_v and $\mathcal{M}(h)$. The following claim summarizes some properties of the sets P_v and Q_v.

Claim 2. If $v = u + w$ is a sum gate then $P_v = P_u \cup P_w$ and $Q_v = Q_u \cap Q_w$. If $v = u \times w$ is a product gate then $P_v = P_u \times P_w$ and $Q_v \times P_w \subseteq Q_u$.

Proof. Only the last inclusion needs a proof. Suppose contrariwise that there are monomials $p \in P_w$ and $q \in Q_v$ such that the monomial pq does not belong to Q_u. Then there must be a monomial $p' \in P_u$ such that $p'pq \notin \mathcal{M}(h)$. But this is impossible because $p'p$ is a monomial of the set $P_v = P_u \times P_w$ produced at gate v, and q is a monomial of the complement Q_v of this gate. □

We define the *content* R_e of an edge $e = (u, v)$ to be the rectangle $R_e := P_u \times Q_v$ if v is a sum $(+)$ gate, and to be the rectangle $R_e := P_v \times Q_v$ if v is a product $(\times)$ gate. Note that the contents of all edges also lie within $\mathcal{M}(h)$. This is clear when v is a product gate. If v is a sum gate then Claim 2 implies that $Q_v \subseteq Q_u$, and hence, also $R_e \subseteq R_u \subseteq \mathcal{M}(h)$ holds.

4.3. Traces

A *trace* in a $(+, \times)$ circuit is its subcircuit obtained by removing exactly one of the two edges entering each sum gate. That is, in order to obtain a trace, we start at the output gate v of the circuit and work backwards by the following rules:

1. If v is a product $(\times)$ gate then *both* its inputs are included.

2. If v is a sum $(+)$ gate then *exactly one* of its inputs is included.

Lemma 7. *For every monomial of h there is a trace, the contents of all whose gates and edges contain this monomial.*

For the contents of *gates*, this fact was proved already by Jerrum and Snir [10, Theorem 3.2].

Proof. Take an arbitrary monomial p of h. Then p belongs to the content $R_v = \mathcal{M}(h)$ of the output gate v. So we start at the output gate and construct a desired trace by traversing the circuit backwards. Suppose we have already reached some gate v, and let u and w be the gates in our circuit entering this gate. Suppose that our monomial p belongs to the content $R_v = P_v \times Q_v$ of v.

Case 1: $v = u \times w$ is a product gate. In this case, the contents of both edges entering v must contain the polynomial p because these contents coincide with R_v. So, it is enough to show that the monomial p must belong to the content $R_u = P_u \times Q_u$ of the head u of the edge (u, v). We know that p belongs to the content $R_v = P_v \times Q_v = P_u \times P_w \times Q_v$ of the tail v. By Claim 2, we also know that $P_w \times Q_v \subseteq Q_u$. So, p belongs to $R_u = P_u \times Q_u$, as desired. Thus, in the case of a product gate, we can include both entering edges into the trace.

Case 2: $v = u + w$ is a sum gate. Since our monomial p belongs to $P_v \times Q_v$, we have that $p = p'q$ for some $p' \in P_v$ and $q \in Q_v$. Since v is a sum gate, Claim 2 implies that $p' \in P_u \cup P_w$ and $q \in Q_u \cap Q_w$. Assume w.l.o.g. that $p' \in P_u$. Then the entire monomial $p = p'q$ belongs to the content $R_e = P_u \times Q_v$ of the edge $e = (u, v)$, as well as to the content $R_u = P_u \times Q_u$ of the head u of this edge. So, in this case, we can include edge (u, v) into the trace. □

4.4. Balanced rectangles

The support $\mathcal{A} = \mathcal{S}(R)$ of each rectangle $R = P \times Q$ is the cross-union $\mathcal{A} = \mathcal{B} \vee \mathcal{C}$ of the families $\mathcal{B} = \mathcal{S}(P)$ and $\mathcal{C} = \mathcal{S}(Q)$ of supports of the monomials in P and Q. Call a cross-union $\mathcal{A} = \mathcal{B} \vee \mathcal{C}$ *m-balanced* if $m/3 \leq |B| \leq 2m/3$ holds for all sets $B \in \mathcal{B}$.

Lemma 8. *Let f be a polynomial of counting complexity $t = \mathrm{C}(f)$. Then for every $m \geq 2$, there exit t cross-sums $\mathcal{A}_1, \ldots, \mathcal{A}_t$ such that:*

(i) *$\mathcal{A}_i \subseteq \mathcal{S}(f)$ holds for all i;*

(ii) *every $\mathcal{A}_i$ is m-balanced;*

(iii) *every set $S \in \mathcal{S}(f)$ of size $|S| \geq m$ belongs to at least one $\mathcal{A}_i$.*

Proof. Take a $(+, \times)$ circuit of size $t = \mathrm{C}(f)$ counting a given polynomial f, and let h be the produced by this circuit polynomial. Take a monomial p of h with at least m variables. By Lemma 7, we know that there must be a trace such that the monomial p belongs to the content $P_v \times Q_v$ of every gate v of this trace. That is, for every gate v of the trace, there must be a pair of monomials $p_v \in P_v$ and $q_v \in Q_v$ such that $p = p_v q_v$; if there are several such pairs, we just fix one of them.

Claim 3. There is a gate v in the trace at which the factor p_v of p has between $m/3$ and $2m/3$ variables.

Proof. Define the *weight* of a gate v of the trace as the number of variables in the monomial p_v. Hence, every input gate of the trace has weight 1, whereas the output gate has weight at least m. By starting at the output gate and traversing the trace backwards by always choosing the input of larger weight, we can find a gate v whose weight is larger than $2m/3$ but both gates u and w entering v have weights at most $2m/3$. By the subadditivity of the length-measure, at least one of the gates u and w, say u, has then weight larger than $(2m/3)/2 = m/3$ and at most $2m/3$. □

Since our circuit *counts* the given polynomial f, the produced by the circuit polynomial h must satisfy $\mathcal{S}(h) = \mathcal{S}(f)$ (by Lemma 6). Let now $v_1, \ldots, v_t$ be the gates of our circuit, and let $\mathcal{A}_i = \mathcal{B}_i \vee \mathcal{C}_i$ be the cross-union of supports $\mathcal{B}_i = \mathcal{S}(P_{v_i})$ and $\mathcal{C}_i = \mathcal{S}(Q_{v_i})$. All these cross-unions $\mathcal{A}_i$ lie in the support $\mathcal{S}(h)$ of h, and hence, also in the support $\mathcal{S}(f)$ of our polynomial f. By Claim 3, for every set $S \in \mathcal{S}(f)$ of size $|S| \geq m$, there must be a cross-sum $\mathcal{A}_i = \mathcal{B}_i \vee \mathcal{C}_i$ and sets $A \in \mathcal{B}_i$ and $C \in \mathcal{C}$ such that $S = B \cup C$ and $m/3 \leq |B| \leq 2m/3$. Thus, items (i) and (iii) are fulfilled. To fulfil also item (ii), it is enough to just remove from each family $\mathcal{B}_i$ every set which is smaller than $m/3$ or larger than $2m/3$. □

5. Proof of Theorem 2

We will need the following upper bound on the number of sets in a cross-union of two families. Recall that $d_r(\mathcal{A})$ is the maximum possible number of sets in a family $\mathcal{A}$ containing a fixed set with r or more elements.

Lemma 9. *Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be families of sets such that $\mathcal{B} \vee \mathcal{C} \subseteq \mathcal{A}$. If every set of $\mathcal{B} \vee \mathcal{C}$ has at least m elements, and $\mathcal{B}$ contains a set with r elements then*

$$|\mathcal{B} \vee \mathcal{C}| \leq d_r(\mathcal{A}) \cdot d_{m-r}(\mathcal{A}) .$$

Note that this upper bound is only nontrivial when $\mathcal{B} \wedge \mathcal{C} \neq \{\emptyset\}$, that is, when sets of $\mathcal{B}$ may intersect sets of $\mathcal{C}$: if $\mathcal{B} \wedge \mathcal{C} \neq \{\emptyset\}$ then $|\mathcal{C}| \leq d_r(\mathcal{A})$ and $|\mathcal{B}| \leq d_{m-r}(\mathcal{A})$ trivially hold.

Proof. Fix a set B in $\mathcal{B}$ of size $|B| = r$, and consider the family

$$\mathcal{A}_B := \{B\} \vee \mathcal{C} = \{B \cup C \colon C \in \mathcal{C}\} \subseteq \mathcal{A} .$$

Associate with every set $A \in \mathcal{A}_B$ the family

$$\mathcal{C}_A := \{C \in \mathcal{C} \colon B \cup C = A\} \subseteq \mathcal{C} .$$

The families $\mathcal{C}_A$ with $A \in \mathcal{A}_B$ give us a partition of $\mathcal{C}$ into $|\mathcal{A}_B|$ pairwise disjoint subfamilies. Since all sets in $\mathcal{A}_B$ contain the set B of size $|B| = r$, we have that

$$|\mathcal{A}_B| \leq d_r(\mathcal{A}) . \tag{7}$$

On the other hand, for each set $A \in \mathcal{A}_B$, all sets of the family $\mathcal{C}_A$, and hence, also all sets of the family $\mathcal{B} \vee \mathcal{C}_A$ contain the (fixed) set $A \setminus B$ of size $|A \setminus B| \geq m - r$, implying that for every $A \in \mathcal{A}_B$,

$$|\mathcal{B} \vee \mathcal{C}_A| \leq d_{m-r}(\mathcal{A}) . \tag{8}$$

Take now any two sets $B' \in \mathcal{B}$ and $C' \in \mathcal{C}$. By the definition of families $\mathcal{C}_A$, the set C' belongs to the family $\mathcal{C}_A$ defined by the set $A = B \cup C'$ of $\mathcal{A}_B$. Hence, the union $B' \cup C'$ belongs to the family $\mathcal{B} \vee \mathcal{C}_A$. This yields

$$|\mathcal{B} \vee \mathcal{C}| \leq \sum_{A \in \mathcal{A}_B} |\mathcal{B} \vee \mathcal{C}_A| .$$

Together with (7) and (8), the desired upper bound follows:

$$|\mathcal{B} \vee \mathcal{C}| \leq \sum_{A \in \mathcal{A}_B} |\mathcal{B} \vee \mathcal{C}_A| \leq \sum_{A \in \mathcal{A}_B} d_{m-r}(\mathcal{A}) = |\mathcal{A}_B| \cdot d_{m-r}(\mathcal{A}) \leq d_r(\mathcal{A}) \cdot d_{m-r}(\mathcal{A}) . \qquad \square$$

Proof of Theorem 2. Let $f = g + h$ be a polynomial such that every monomial of h has at least $m \geq 2$ variables, and every monomial of g has fewer than $m/3$ variables. Let $\mathcal{A} := \mathcal{S}(h)$ be the support of h. Our goal is to show that then $\mathrm{C}(f)$ must be at least $|\mathcal{A}|$ divided by the maximum K of $d_r(\mathcal{A}) \cdot d_{m-r}(\mathcal{A})$ over all integers r between $m/3$ and $2m/3$.

By Lemma 8, there are at most $t = \mathrm{C}(f)$ m-balanced cross-sums $\mathcal{A}_1, \ldots, \mathcal{A}_t \subseteq \mathcal{S}(f)$ such that every set of $\mathcal{S}(f)$ with at least m elements belongs to at least one of these cross-sums. Since the cross-sums $\mathcal{A}_i$ are m-balanced, they cannot have sets with fewer than $m/3$ elements, and hence, cannot have any support of a monomial of the "smaller" polynomial g. This implies that each $\mathcal{A}_i$ actually lies entirely in the support $\mathcal{A} = \mathcal{S}(h)$ of the "larger" polynomial h. By Lemma 9, each $|\mathcal{A}_i|$ is at most the product $d_{r_i}(\mathcal{A}) \cdot d_{m-r_i}(\mathcal{A})$ for some integer $m/3 \leq r_i \leq 2m/3$. By taking an $r = r_i$ for which this product is maximal, the desired lower bound $t \geq |\mathcal{A}|/d_r(\mathcal{A}) \cdot d_{m-r}(\mathcal{A})$ follows. $\square$

6. Proof of Theorem 3

We will give an amazing simple proof of a more general result, and then easily derive Theorem 3 as its special case. Associate with every finite set P of monomials a nonnegative real number $\mu(P)$. Call such a measure μ *legal* if the following three conditions are fulfilled.

1. *Normalization*: $\mu(\{x_i\}) \leq 1$ for every variable x_i.
2. *Additivity*: $\mu(P \cup Q) \leq \mu(P) + \mu(Q)$.
3. *Multiplicativity*: $\mu(P \times Q) \leq \mu(P) \cdot \mu(Q)$.

Call a polynomial h *(k,l)-free with respect to* μ if for any two sets P and Q of monomials, $P \times Q \subseteq \mathcal{M}(h)$ implies $\mu(P) \leq k$ or $\mu(Q) \leq l$. Recall that $\mathcal{M}(h)$ stands for the set of all monomials of h.

Theorem 10. *Let $1 \leq k \leq l$. If a polynomial f is (k,l)-free with respect to some legal measure μ then*

$$\mathrm{A}(h) \geq \frac{\mu(\mathcal{M}(h))}{2lk^2}\,.$$

Proof. Fix some $(+,\times)$ circuit computing h. Recall that the *content* R_e of an edge $e = (u,v)$ in the circuit is the rectangle $R_e = P_u \times Q_v$ if v is a sum $(+)$ gate, and is the rectangle $R_e = P_v \times Q_v$ if v is a product $(\times)$ gate; as before, P_v is the set of monomials produced at the gate v, and Q_v is the complement of v. Call an edge e of the circuit *light* if $\mu(R_e) \leq lk^2$.

Lemma 11. *Every monomial of h is contained in the content of at least one light edge.*

Proof. Call a gate u *small* if $\mu(P_u) \leq k$, and *large* otherwise. We can assume that the output gate is large because otherwise we would have $\mu(\mathcal{M}(h)) \leq |\mathcal{M}(h)| \leq k$ (by the additivity of μ), and there would be nothing to prove.

Fix a monomial p in $\mathcal{M}(h)$. By Lemma 7, there must be a trace in the circuit, the contents R_e of all whose edges e contain this vector. Start at the output gate of the trace (which is also the output gate of the entire circuit), and construct a path by going backwards and using the following rule, where u is the last already reached gate:

1. If u is a sum $(+)$ gate then go to the (unique) node entering u in the trace.
2. If u is a product $(\times)$ gate then go to any of its two inputs if they both are large or both are small, and go to the large input if the second input is small.

Since the output gate is large and (due to the normalization property of our measure) every input gate is small, we will eventually reach some input gate. Since the first node of this path is small, and the last one is large, there must be an edge $e = (u,v)$ such that $\mu(P_u) \leq k$ but $\mu(P_v) > k$. It remains to show that this edge must be light. Since $P_v \times Q_v \subseteq M$ and $\mu(P_v) > k$, the (k,l)-freeness of our polynomial f with respect to the measure μ implies that $\mu(Q_v) \leq l$ must hold.

If $v = u + w$ is a sum gate then $R_e = P_u \times Q_v$, and the multiplicativity of μ yields

$$\mu(R_e) = \mu(P_u \times Q_v) \leq \mu(P_u) \cdot \mu(Q_v) \leq kl\,.$$

If $v = u \times w$ is a product gate then $R_e = P_v \times Q_v = P_u \times P_w \times Q_v$. Since the gate u is small (it has $\mu(P_u) \leq k$), step 2 in the construction of the path implies that the second gate w entering v must be also small, that is, $\mu(P_w) \leq k$ must hold as well. The multiplicativity of μ yields

$$\mu(R_e) = \mu(P_u \times P_w \times Q_v) \leq \mu(P_u) \cdot \mu(P_w) \cdot \mu(Q_v) \leq k^2 l\,. \qquad \square$$

To finish the proof of Theorem 10, let E be the set of all light edges in the circuit. By Lemma 11, every monomial $p \in \mathcal{M}(h)$ belongs to the content R_e of at least one edge $e \in E$; hence, $\mathcal{M}(h) \subseteq \bigcup_{e\in E} R_e$. Since every edge $e \in E$ is light, its content R_e has bounded measure: $\mu(R_e) \leq lk^2$. The additivity of the measure yields

$$\mu(\mathcal{M}(h)) \leq \sum_{e\in E} \mu(R_e) \leq lk^2 \cdot |E| \,.$$

Since the gates have fanin at most two, the total number of gates must be at least $|E|/2$, as desired. □

Proof of Theorem 3. Take a polynomial f, and suppose that it is (k, l)-free for some $1 \leq k \leq l$. Let $\mathcal{A} = \mathcal{S}(f)$ be the family of supports of monomials of f. The (k, l)-freeness of f means that for every two antichains $\mathcal{B}$ and $\mathcal{C}$ such that $\mathcal{B} \vee \mathcal{C} \subseteq \mathcal{A}$, at least one of $|\mathcal{B}| \leq k$ or $|\mathcal{C}| \leq l$ must hold.

Fix a $(+, \times)$ circuit counting the polynomial f, and let h be the polynomial (syntactically) produced by the circuit itself. Our goal is to show that the circuit must have at least $|\mathcal{L}(f)|/lk^2$ gates, where $\mathcal{L}(f) \subseteq \mathcal{S}(f)$ is the lower support of f; this is the antichain formed by only taking *minimal* sets of $\mathcal{S}(f)$, those containing no other set of $\mathcal{S}(f)$.

Since the circuit counts f, Lemma 6 tells us that $\mathcal{S}(h) = \mathcal{S}(f)$ must hold, that is, the produced polynomial h must have the same support as the original polynomial f. In particular, then $\mathcal{L}(h) = \mathcal{L}(f)$ must hold as well.

We are going to apply Theorem 10 with the following measure $\mu(P) := |\mathcal{L}(P)|$ of sets P of monomials, where $\mathcal{L}(P)$ is the family of minimal supports of monomials in P. Since $\mathcal{L}(h) = \mathcal{L}(f)$ holds, we have that $\mu(\mathcal{M}(f)) = \mu(\mathcal{M}(h))$. So, it remains to show that this measure μ is legal, and that the polynomial h is (k, l)-free with respect to this measure.

To show that the polynomial h is (k, l)-free with respect to the measure $\mu(P) = |\mathcal{L}(P)|$, take any two sets P and Q of monomials such that $P \times Q \subseteq \mathcal{M}(h)$, and let $\mathcal{B} = \mathcal{L}(P)$ and $\mathcal{C} = \mathcal{L}(Q)$. Then also

$$\mathcal{B} \vee \mathcal{C} = \mathcal{L}(P) \vee \mathcal{L}(Q) \subseteq \mathcal{S}(P \times Q) \subseteq \mathcal{S}(h) = \mathcal{S}(f) \,.$$

Since $\mathcal{B} \vee \mathcal{C} \subseteq \mathcal{S}(f)$, the (k, l)-freeness of f implies that either $\mu(P) = |\mathcal{B}| \leq k$ or $\mu(Q) = |\mathcal{C}| \leq l$ must hold, as desired.

Let us now show that the measure $\mu(P) = |\mathcal{L}(P)|$ is legal. The measure is clearly normalized because $|\mathcal{L}(\{x_i\})| = 1$. To show the two remaining properties (additivity and multiplicativity), take any two finite sets P and Q of monomials.

Since every minimal set of a union of two families must be minimal in at least one of these families, we have that $\mathcal{L}(P \cup Q) \subseteq \mathcal{L}(P) \cup \mathcal{L}(Q)$, implying that

$$\mu(P \cup Q) = |\mathcal{L}(P \cup Q)| \leq |\mathcal{L}(P)| + |\mathcal{L}(Q)| = \mu(P) + \mu(Q) \,;$$

hence, the additivity. Since every minimal set of a cross-union of two families must be a union of some minimal sets of these families, we also have that $\mathcal{L}(P \times Q) \subseteq \mathcal{L}(P) \vee \mathcal{L}(Q)$, implying that

$$\mu(P \times Q) = |\mathcal{L}(P \times Q)| \leq |\mathcal{L}(P) \vee \mathcal{L}(Q)| \leq |\mathcal{L}(P)| \cdot |\mathcal{L}(Q)| = \mu(P) \cdot \mu(Q) \,;$$

hence, the multiplicativity. □

7. Concluding remarks

In this paper, we proved general lower bounds for a model of circuits whose power lies strictly between that of monotone *boolean* and monotone *arithmetic* $(+, \times)$ circuits—the model of *counting* $(+, \times)$ circuits.

A yet another interesting model, whose power lies between the monotone arithmetic and monotone boolean complexities, is that of *tropical* $(\min, +)$ circuits. Inputs here are variables $x_1, \ldots, x_n$ taking their values in $\mathbb{N} = \{0, 1, \ldots\}$, and gates are fanin-2 Min and Sum operations.

That is, the "sum" $(+)$ in a $(+,\times)$ circuit is now interpreted as $\min\{x,y\}$, and the "product" $(\times)$ as $x+y$. Each such circuit solves some minimization problem of the form

$$\hat{f}(x) = \min_{e\in E} \sum_{i=1}^{n} e_i x_i \,,$$

where $E \subset \mathbb{N}^n$ is some (fixed) finite set of vectors. Tropical circuits are important because any dynamic programming algorithm for minimization problems, using only Min and Sum operations, is just a (recursively constructed) tropical circuit.

As counting and deciding $(+,\times)$ circuits, tropical circuits can be also viewed as standard $(+,\times)$ circuits "tropically" computing a given polynomial

$$f(x) = \sum_{e\in E} c_e \prod_{i=1}^{n} x_i^{e_i}$$

in the following sense. The *tropicalization* of f is the minimization problem $\hat{f}$ above. For example, the tropicalization of $f = xy^2 + 3y^2z^3$ is $\hat{f} = \min\{x+2y, 2y+3z\}$.

If a $(+,\times)$ circuit produces some polynomial h then we say that the circuit *tropically computes* a given polynomial f, if $\hat{h}(a) = \hat{f}(a)$ holds for all $a \in \mathbb{N}^n$. The difference from counting $(+,\times)$ circuits is that, in tropically computing $(+,\times)$ circuits, the absorption axiom $x + xy = x$ is allowed ($\min\{x, x+y\} = x$), but the idempotent axiom $x^2 = x$ is not ($x + x \neq x$ unless $x = 0$). For a polynomial f, its *tropical complexity* $\mathrm{T}(f)$ is the minimum size of a $(+,\times)$ circuit tropically computing f. It is clear that $\mathrm{T}(f) \leq \mathrm{A}(f)$ holds for every polynomial f.

An interesting fact is that $\mathrm{T}(f) \geq \mathrm{A}(f_{\mathrm{le}})$ holds, as long as the polynomial f is multilinear [10, 11]. Recall that f_{le} is lower envelope of f, that is, a homogeneous polynomial consisting of all monomials of f of smallest degree. In particular, this implies that for multilinear polynomials f which are also *homogeneous*, their tropical complexity is at least their counting complexity.

Let us note, however, that the *homogeneity* is here crucial: no similar relation holds for nonhomogeneous polynomials. In fact, then both gaps $\mathrm{C}(f)/\mathrm{T}(f)$ and $\mathrm{T}(f)/\mathrm{C}(f)$ can be exponential in the number of variables. This means that tropical and counting complexity measures are incomparable!

To show the first gap, consider the extension $f = \mathrm{Per}_n + \sum_{i,j=1}^{n} x_{ij}$ of the permanent polynomial Per_n by adding the sum of all n^2 variables. Corollary 1 implies that the *counting* complexity of f remains exponential: $\mathrm{C}(f) = 2^{\Omega(n)}$. But the *tropical* complexity of f is $\mathrm{T}(f) \leq n^2$: since variables cannot take negative values, the minimum will be achieved on a single variable. Hence, the first gap $\mathrm{C}(f)/\mathrm{T}(f) = 2^{\Omega(n)}$.

To show the second gap, consider the isolated node polynomial I_n of n^2 variables introduced in Sect. 3.1. Let f be the linearization of I_n. That is, f is obtained from I_n by removing all nonzero exponents in all monomials. We observed that $\mathrm{C}(f) = O(n^3)$. On the other hand, every monomial of f has degree between n and $2n$, and the monomials of degree n correspond to perfect matchings. Thus, the lower envelope f_{le} of f is just the permanent polynomial, that is, $f_{\mathrm{le}} = \mathrm{Per}_n$. Since $\mathrm{A}(\mathrm{Per}_n) \geq \mathrm{C}(\mathrm{Per}_n) = 2^{\Omega(n)}$ (see Corollary 1) and $\mathrm{T}(f) \geq \mathrm{A}(f_{\mathrm{le}})$ holds for every multilinear polynomial, the desired lower bound $\mathrm{T}(f) = 2^{\Omega(n)}$ follows.

Acknowledgments

I am thankful to Georg Schnitger for interesting discussion on tropical circuits, to Tsuyoshi Ito for a hint in separating the decision and counting complexities, and especially to Igor Sergeev for numerous in-depth discussions on arithmetic circuit complexity. The research was supported by the DFG grant SCHN 503/6-1.

A. Number of *s*-*t* paths is hard to count

The *s*-*t path polynomial* P_n has one variable $x_{i,j}$ for each edge of a complete directed graph K_n on n nodes $\{1,\ldots,n\}$. Each monomial of P_n is the product of all variables corresponding to the edges of some simple directed path from node $s = 1$ to node $t = n$. *Hamiltonian paths* are s-t paths with $n - 1$ edges. The Bellman–Ford dynamic programming algorithm [4, 5] immediately yields that the *decision* complexity of this polynomial is small: $\mathrm{D}(P_n) = O(n^3)$.

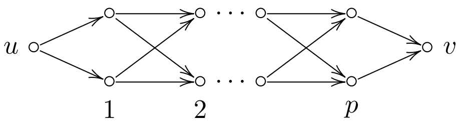
Every 0-1 input a to the polynomial P_n specifies some subgraph of K_n, and $P_n(a)$ is then exactly the number of simple s-t paths in this subgraph. Thus, every $(+, \times)$ circuit *counting* the polynomial P_n counts the number of s-t paths in the corresponding graphs. Our goal is to show that every such circuit must have exponential size. We will do this *indirectly* via reductions to some known results.

The Hamiltonian path function H_n is a monotone boolean function which, given a subgraph G of K_n (specified by 0-1 input), decides whether G has a Hamiltonian path. The Clique function Cl_n is also a monotone boolean function which, given a subgraph G of K_n, decides whether G has a complete subgraph on $\sqrt{n}$ or more nodes.

By tightening Razborov's method of approximations [14], Alon and Boppana [3] proved that Cl_n requires monotone boolean circuits of size $2^{n^{\Omega(1)}}$. Then Pudlák [12, Theorem 6] observed that Razborov's proof also works when arbitrary monotone *real-valued* functions $g : \mathbb{R}^2 \to \mathbb{R}$ are allowed as gates. This led to a lower bound $\mathrm{B}(Cl_n) = 2^{n^{\Omega(1)}}$ for the clique function, where $\mathrm{B}(f)$ stands for the minimum number of gates in a monotone real-valued circuit computing f.

By known reductions, this gives an exponential lower bound also for H_n. Namely, say that a boolean function $f(x_1, \ldots, x_n)$ is a *monotone projection* of a boolean function $g(y_1, \ldots, y_m)$ if there exists an assignment $\sigma : \{y_1, \ldots, y_m\} \to \{x_1, \ldots, x_n, 0, 1\}$ such that $f(x_1, \ldots, x_n) = g(\sigma(y_1), \ldots, \sigma(y_m))$. It is clear that then $\mathrm{B}(f) \leq \mathrm{B}(g)$: we only have to change inputs of a circuit for g to obtain a circuit computing f. Results of Valiant [20] imply that Cl_n is a monotone projection of H_m for $m = n^{O(1)}$; as noted by Alon and Boppana [3], already $m = O(n^2)$ is enough in this case. Thus, we have that $\mathrm{B}(H_m) \geq \mathrm{B}(Cl_n) \geq 2^{n^{\Omega(1)}}$. It remains therefore to show that $\mathrm{B}(H_n) \leq \mathrm{C}(P_m) + 1$ holds for $m = n^{O(1)}$.

This can be shown by a standard trick allowing to decide whether a graph has a Hamiltonian s-t path, if one can count the number of all s-t paths. Set $p = n \log n$, and assume for simplicity that p is an integer. Given an input graph G_n on n nodes, replace each edge (u, v) by a directed acyclic graph containing exactly 2^p paths from u to v:



The resulting graph G_m has only $m = O(pn^2) = O(n^3 \log n)$ nodes.

Claim 4. A graph G_n has a Hamiltonian path if and only if the number of s-t paths in G_m is at least $T = (2^p)^{n-1}$.

Proof. By the construction, every s-t path of length l in G_n gives exactly $(2^p)^l$ s-t paths in the resulting graph G_m. Thus, if G_n has a Hamiltonian s-t path (of length $l = n - 1$) then the graph G_m has at least T s-t paths. On the other hand, if G_n has no Hamiltonian s-t path then the longest s-t path has at most $n - 2$ edges. The number of s-t paths of length at most $n - 2$ is bounded from above by n^{n-1}. So, in this case, G_m can have at most $(2^p)^{n-2} \cdot n^{n-1} = T/n$ s-t paths. □

Suppose now we have a $(+, \times)$ circuit counting P_m. Extend this circuit by adding one additional threshold gate testing whether the number computed at the output gate of the circuit is at least T. By Claim 4, the resulting circuit is a (very special) monotone real-valued circuit computing the Hamiltonian path function H_n. □

References

[1] R. Ahlswede and D.E. Daykin. An inequality for the weights of two families of sets, their unions and intersections. Z. Wahrscheinlichkeitstheor. Verwandte Geb., 43:183–185, 1978.

[2] N. Alon. Combinatorial nullstellensatz. Combinatotics, Probability & Computing, 8:7–29, 1999.

[3] N. Alon and R. Boppana. The monotone circuit complexity of boolean functions. Combinatorica, 7(1):1–22, 1987.

[4] R. Bellman. On a routing problem. Quarterly of Appl. Math., 16:87–90, 1958.

[5] L.R. Ford. Network flow theory. Technical Report P-923, The Rand Corp., 1956.

[6] S.B. Gashkov. On one method of obtaining lower bounds on the monotone complexity of polynomials. Vestnik Moscov Univ., Ser. 1 Mat., Mekh., 5:7–13, 1987.

[7] S.B. Gashkov and I.S. Sergeev. A method for deriving lower bounds for the complexity of monotone arithmetic circuits computing real polynomials. Math. Sbornik, 203(10):33–70, 2012 (in Russian). English translation in: Sbornik: Mathematics, 203(10) (2012) 1411–1147.

[8] P. Hrubes and A. Yehudayoff. Homogeneous formulas and symmetric polynomials. Computational Complexity, 20(3):559–578, 2011.

[9] L. Hyafil. On the parallel evaluation of multivariate polynomials. SIAM J. Comput., 8(2):120–123, 1979.

[10] M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. J. ACM, 29(3):874–897, 1982.

[11] S. Jukna. Lower bounds for tropical circuits and dynamic programs. Theory of Comput. Syst., 57(1):160–194, 2014.

[12] P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. J. Symb. Log., 62(3):981–998, 1997.

[13] R. Raz and A. Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. J. Comput. Syst. Sci., 77(1):167–190, 2011.

[14] A. Razborov. Lower bounds on the monotone complexity of some boolean functions. Sov. Math. Doklady, 31:354–357, 1985.

[15] C.P. Schnorr. A lower bound on the number of additions in monotone computations. Theor. Comput. Sci., 2(3):305–315, 1976.

[16] R. Sengupta and H. Venkateswaran. A lower bound for monotone arithmetic circuits computing 0-1 permanent. Theor. Comput. Sci., 209(1–2):389–398, 1998.

[17] E. Shamir and M. Snir. On the depth complexity of formulas. Math. Syst. Theory, 13:301–322, 1980.

[18] M. Snir. Size-depth trade-offs for monotone arithmetic circuits. Theor. Comput. Sci., 82(1):85–93, 1991.

[19] P. Tiwari and M. Tompa. A direct version of Shamir and Snir's lower bounds on monotone circuit depth. Inf. Process. Lett., 49(5):243–248, 1994.

[20] L. G. Valiant. Completeness classes in algebra. In Proc. of 11h Annual ACM Symp. on Theory of Computing, pages 249–261, 1979.

[21] L.G. Valiant. Negation can be exponentially powerful. Theor. Comput. Sci., 12:303–314, 1980.