Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 94 (2022)

NOTES ON BOOLEAN READ-K CIRCUITS

STASYS JUKNA*

Abstract. A monotone Boolean circuit computing a Boolean function f is a read-k circuit if the polynomial produced (purely syntactically) by the arithmetic $(+,\times)$ version of the circuit has the property that for every prime implicant of f, the polynomial contains a monomial with the same set of variables, each appearing with degree $\leqslant k$. Every monotone circuit is a read-k circuit for some k.

We show that monotone read-1 circuits have the same power as tropical $(\min,+)$ circuits solving 0/1 optimization, and the same power as monotone arithmetic $(+,\times)$ circuits computing multilinear homogeneous polynomials. We also show that monotone read-1 circuits are not weaker than multilinear non-monotone $(\vee,\wedge,\neg)$ circuits. Finally, we show that already read-2 monotone circuits can be exponentially smaller than read-1 monotone circuits.

Key words. arithmetic circuits, multilinear circuits, tropical circuits, lower bounds

AMS subject classifications. 68Q17, 94C11

1. Introduction.

Let F be a monotone Boolean $(\vee,\wedge)$ circuit F computing a Boolean function f. If we replace every OR gate by an addition gate, and every AND gate by a multiplication gate, then the polynomial P computed by the obtained monotone arithmetic $(+,\times)$ circuit has the following two properties: every monomial of P contains all variables of at least one prime implicant of f, and every prime implicant of f has at least one its "shadow" in P, that is, a monomial with the *same* set of variables. We call F a *read-k* circuit if every prime implicant of f has at least one "shadow" monomial in P with the degrees of all its variables not exceeding k.

We observe that already read-1 circuits capture the power of several interesting types of circuits considered so far. In particular, read-1 circuits are related to dynamic programming (DP) algorithms. Many classical DP algorithms for minimization problems are "pure" in that they only use $(\min,+)$ operations in their recursion equations. Such are, for example, the classical Bellman–Ford–Moore shortest s-t path algorithm [1, 5, 20], the Roy–Floyd–Warshall all-pairs shortest paths algorithm [28, 4, 32], the Bellman–Held–Karp travelling salesman algorithm [2, 8], the Dreyfus–Levin–Wagner Steiner tree algorithm [3, 17], and many others. Pure DP algorithms are (special, recursively constructed) *tropical* $(\min,+)$ circuits.

Monotone read-1 circuits are also related to monotone arithmetic $(+,\times)$ circuits computing multilinear polynomials, as well as to so-called multilinear (not necessarily monotone) DeMorgan $(\vee,\wedge,\neg)$ circuits, where the Boolean functions computed at the inputs to any AND gate must *depend* on disjoint sets of variables.

Namely, we prove the following.

(1) Read-1 circuits have the *same* power as *tropical* $(\min,+)$ circuits solving 0/1 minimization problems (Theorem 1), and the *same* power as monotone *arithmetic* $(+,\times)$ circuits computing multilinear homogeneous polynomials (Theorem 2).

(2) Read-1 circuits are *not weaker* than *multilinear* DeMorgan $(\vee,\wedge,\neg)$ circuits (Theorem 3).

(3) Already read-2 circuits can be exponentially smaller than read-1 and, hence, exponentially smaller than tropical $(\min,+)$, monotone arithmetic $(+,\times)$, and multilinear $(\vee,\wedge,\neg)$ circuits (Lemma 6).

*Faculty of Mathematics and Computer Science, Vilnius University, Lithuania (stjukna@gmail.com, https://web.vu.lt/mif/s.jukna/).

ISSN 1433-8092

2. Preliminaries. We start with recalling one simple but important concept: the set of exponent vectors "produced" (purely syntactically) by a circuit over any semiring. Recall that a (commutative) *semiring* $(R, \oplus, \odot)$ consists of a set R closed under two associative and commutative binary operations "addition" $x \oplus y$ and "multiplication" $x \odot y$, where multiplication distributes over addition: $x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)$. That is, in a semiring, we can "add" and "multiply" elements, but neither "subtraction" nor "division" are necessarily possible.

A *circuit* F over a semiring $(R, \oplus, \odot)$ is a directed acyclic graph; parallel edges joining the same pair of nodes are allowed. Each indegree-zero node (an *input* node) holds either one of the variables $x_1, \ldots, x_n$ or a semiring element $c \in R$. Every other node, a *gate*, has indegree two and performs one of the semiring operations $\oplus$ or $\odot$ on the values computed at the two gates entering this gate. The *size* of a circuit is the total number of gates in it.

Every circuit F over a semiring $(\oplus, \odot)$ *produces* (purely syntactically) a unique set of (exponent) vectors $B_F \subseteq \mathbb{N}^n$ in a natural way, where $\vec{0}$ is the all-0 vector, and $\vec{e}_i \in \{0,1\}^n$ has exactly one 1 in the i-th position:

- if $F = c \in R$, then $B_F = \{\vec{0}\}$;
- if $F = x_i$, then $B_F = \{\vec{e}_i\}$;
- if $F = G \oplus H$, then $B_F = B_G \cup B_H$;
- if $F = G \odot H$, then $B_F = B_G + B_H := \{x + y \colon x \in B_G, y \in B_H\}$.

That is, at an "addition" $(\oplus)$ gate, we take the union of the sets produced at the two gates entering that gate, and at a "multiplication" $(\odot)$ gate, we take the Minkowski sum of these sets.

An equivalent, and apparently more intuitive, way to see how the sets $B_F \subseteq \mathbb{N}^n$ of vectors are produced is to convert the circuit F to a monotone *arithmetic* $(+, \times)$ circuit by replacing every $\oplus$ gate with an addition $(+)$ gate, every $\odot$ gate with a multiplication $(\times)$ gate. If "additive" neutral element with $0 \odot x = 0$ is among the inputs then we replace this element by constant 0. Every other input semiring element $c \neq 0$ is replaced by constant 1.

The obtained arithmetic $(+, \times)$ circuit produces (purely syntactically) some polynomial $P(x) = \sum_{b \in B} c_b \prod_{i=1}^n x_i^{b_i}$, where $B \subseteq \mathbb{N}^n$ is the set of its exponent vectors, and the coefficients c_b are positive integers indicating the number of times the corresponding monomial $\prod_{i=1}^n x_i^{b_i}$ appears in the polynomial. We call $P(x)$ the polynomial *defined* by the original $(\oplus, \odot)$ circuit. Since the exponent vector of a product of two monomials is the sum of exponent vectors of these monomials, the set B of exponent vectors of the polynomial P is exactly the set B_F of vectors produced by the original circuit F.

It is clear that the same circuit F with only "addition" $(\oplus)$ and "multiplication" $(\odot)$ gates can compute *different* functions over different semirings. Say, the circuit $F = x \odot y \oplus z$ computes $xy + z$ over the arithmetic $(+, \times)$ semiring, but computes $\min\{x + y, z\}$ over the tropical $(\min, +)$ semiring. It is, however, important to note that:

1. The set of exponent vectors *produced* by a circuit over any semiring is always the same—it only depends on the circuit itself, not on the underlying semiring.
2. If a circuit F over a semiring produces a set $B \subseteq \mathbb{N}^n$, then the circuit computes a polynomial (over the same semiring) whose set of exponent vectors in B.

Item 1 is trivial. Item 2 holds because there is a natural homomorphism from the semiring of n-variate polynomials to the semiring $(2^{\mathbb{N}^n}, \cup, +)$ of finite sets of vectors that maps every polynomial $f(x) = \sum_{a \in A_f} c_a \prod_{i=1}^n x_i^{a_i}$ to the set $A_f \subseteq \mathbb{N}^n$ of its exponent

vectors. In particular, every single variable x_i is mapped to $A_{x_i} = \{\vec{e}_i\}$, and every input constant $c \in R$ is mapped to $A_c = \{\vec{0}\}$. That this is indeed a homomorphism follows from easily verifiable equalities $A_{f\oplus h} = A_f \cup A_h$ and $A_{f\odot h} = A_f + A_h$, the latter sum being the Minkowski sum of the sets A_f and A_h.

In this article, we will consider circuits over the following three semirings $(R, \oplus, \odot)$: the arithmetic semiring $(\mathbb{R}_+, +, \times)$ where $\mathbb{R}_+$ is the set of nonnegative real numbers, the tropical semiring $(\mathbb{R}_+, \min, +)$, and the Boolean semiring $(\{0,1\}, \vee, \wedge)$. That is, we will consider the following three types of circuits:

- $x \oplus y := x + y$ and $x \odot y := xy$ (monotone arithmetic circuits);
- $x \oplus y := x \vee y$ and $x \odot y := x \wedge y$ (monotone Boolean circuits);
- $x \oplus y := \min(x, y)$ and $x \odot y := x + y$ (tropical circuits).

An exception is section 4, where we also consider non-monotone DeMorgan $(\vee, \wedge, \neg)$ circuits.

3. Monotone read-k circuits. A monotone Boolean circuit is a circuit over the Boolean semiring $(\oplus, \odot)$ with $x \oplus y := x \vee y$ and $x \odot y := x \wedge y$; the domain is $\{0,1\}$. We will always assume that such circuits are constant-free: constant inputs 0 and 1 can be easily eliminated without increasing the circuit size (to avoid trivialities, we will only consider circuits computing non-constant functions).

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if $a \leqslant b$ and $f(a) = 1$ imply $f(b) = 1$. A *lowest one* of a (not necessarily monotone) Boolean function $f : \{0,1\}^n \to \{0,1\}$ is a vector $a \in f^{-1}(1)$ such that $f(b) = 0$ for all vectors $b \leqslant a$, $b \neq a$. We will denote the set of all lowest ones of f by A_f. Note that the set A_f is an *antichain* ($a \in A$ and $b \leqslant a$ implies $a = b$) and, the function f is monotone, it uniquely determines the entire function f:

$$f(x) = \bigvee_{a\in A_f} \bigwedge_{i\in \sup(a)} x_i \,.$$

Here and throughout, $\sup(x) := \{i \colon x_i \neq 0\}$ stands for the *support* of a vector $x \in \mathbb{N}^n$, that is, for the set of its nonzero positions. The *upward closure* of a set $A \subseteq \mathbb{N}^n$ of vectors is the set

$$A^{\uparrow} := \{b \in \mathbb{N}^n \colon b \geqslant a \text{ for some } a \in A\} \,.$$

Note that for every monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$, we have $f^{-1}(1) = A_f^{\uparrow} \cap \{0,1\}^n$. A *shadow* of a vector $a \in \mathbb{N}^n$ is any vector $b \in \mathbb{N}^n$ with $\sup(b) = \sup(a)$. That is, shadows of vectors are nonnegative integer vectors with the same set of nonzero positions.

We start with the following "folklore" property of sets of exponent vectors produced by monotone Boolean circuits. Let f be the monotone Boolean function, and let $A_f \subseteq f^{-1}(1)$ be the set of lowest ones of f. Let F be a monotone Boolean $(\vee, \wedge)$ circuit F, and let $B \subseteq \mathbb{N}^n$ be the set of exponent vectors produced by the monotone arithmetic $(+, \times)$ version of F.

Claim 1. *The circuit F computes f if and only if $B \subseteq A_f^{\uparrow}$ and every lowest one $a \in A_f$ of f has at least one its shadow in B.*

Proof. Our Boolean function f is of the form $f(x) = \bigvee_{a\in A_f} \bigwedge_{i\in\sup(a)} x_i$, while the Boolean function computed by the circuit F is of the form $F(x) = \bigvee_{b\in B} \bigwedge_{i\in\sup(b)} x_i$. The "if" direction follows directly from the following simple observation: for every input $x \in \{0,1\}^n$, we have $f(x) = 1$ iff $\sup(x) \supseteq \sup(a)$ for some $a \in A_f$, which happens iff $x \in A_f^{\uparrow}$. Hence, $B \subseteq A_f^{\uparrow}$ yields $F(x) \leqslant f(x)$, while the fact that every vector $a \in A_f$ has at least one its shadow in B yields $f(x) \leqslant F(x)$.

Now assume that the circuit F computes f. If $b \notin A^{\uparrow}$ held for some vector $b \in B$, then on the input $x \in \{0,1\}^n$ with $x_i = 1$ iff $i \in \sup(b)$, we would have $\sup(a) \setminus \sup(b) \neq \emptyset$ and, hence, $f(x) = 0$. But $F(x) = 1$, a contradiction. To show that every vector $a \in A_f$ must have at least one its shadow in B, suppose for the contradiction that there is a vector $a \in A_f$ such that $\sup(b) \neq \sup(a)$ holds for all vectors $b \in B$. Since $B \subseteq A_f^{\uparrow}$ and since A_f is an antichain, $\sup(b) \subset \sup(a)$ (proper inclusion) cannot hold. So, we have $\sup(b) \setminus \sup(a) \neq \emptyset$ for all vectors $b \in B$. Then $F(a) = 0$ while $f(a) = 1$, a contradiction. □

Remark 3.1. If $P(x) = \sum_{b \in B} c_b \prod_{i=1}^{n} x_i^{b_i}$ is the polynomial produced by the arithmetic $(+,\times)$ version of the circuit F, then Claim 1 implies that F computes f iff every monomial of P contains at least one prime implicant $p \leqslant f$ of f as a factor, and for every prime implicant $p \leqslant f$ there is a monomial in P with the same set of variables as p (a "shadow" of p in P). For example, the Boolean circuit $F = (x \vee y)(x \vee z) \vee xy$ computes the Boolean function $f = x \vee yz$, whose prime implicants are x and yz; hence, $A_f = \{(1,0,0),(0,1,1)\}$. The arithmetic version $F' = (x+y)(x+z)$ of F produces the polynomial $P = x^2 + xz + 2xy + yz$. Hence, the set of exponent vectors produced by the Boolean circuit F is $B = \{(2,0,0),(1,0,1),(1,1,0),(0,1,1)\} \subseteq A^{\uparrow}$. □

In general, shadows $b \in B$ of vectors $a \in A_f$ guaranteed by Claim 1 may have large entries (as large as 2^s, where s is the size of the Boolean circuit F computing f): we only know that $\sup(b) = \sup(a)$. In read-k circuits, we restrict the magnitude of entries in shadows b. A vector $b \in \mathbb{N}^n$ is *k-bounded* if no its entry is larger than k.

Definition 1 (Monotone read-k circuits). A monotone Boolean circuit F computing a Boolean function f a *read-k circuit* if the set $B \subseteq \mathbb{N}^n$ of vectors produced by F has the following two properties:

(i) $B \subseteq A_f^{\uparrow}$, that is, for every $b \in B$ there is an $a \in A_f$ such that $b \geqslant a$;

(ii) every lowest one $a \in A_f$ of f has at least one k-bounded its shadow in B.

That is, the (arithmetic) polynomial P produced by the monotone arithmetic $(+,\times)$ version of the $(\vee,\wedge)$ circuit F must now contain, for every prime implicant of f, at least one monomial with the *same* set of variables, with each variable appearing in that monomial with a power not exceeding k. There are no restrictions on the degrees of other monomials of P.

In particular, the circuit F is a *read-once* circuit (that is, a read-1 circuit) iff the inclusions $A \subseteq B \subseteq A^{\uparrow}$ hold.

For a monotone Boolean function f, let

$$\mathsf{B}_k(f) := \text{min size of a monotone read-}k\ (\vee,\wedge)\ \text{circuit computing } f.$$

In the next section, we will show that, in the context of dynamic programming, already the monotone read-once circuits are interesting. Namely, $\mathsf{B}_1(f)$ *coincides* with the minimum size of a tropical $(\min,+)$ circuit solving the minimization problem $f(x) = \min_{a \in A_f} \sum a_i x_i$.

3.1. From tropical (min,+) to Boolean read-once circuits. We now consider *tropical* $(\min,+)$ circuits. These are the circuits over the semiring $(R,\oplus,\odot)$ with $R = \mathbb{R}_+$, $x \oplus y := \min(x,y)$ and $x \odot y := x + y$. If $B \subseteq \mathbb{N}^n$ is the set of "exponent" vectors produced by such a circuit F, then the circuit computes the tropical polynomial $f(x) = \min_{b \in B} \langle x,b \rangle + c_b$ with some "coefficients" $c_b \in \mathbb{R}_+$, where $\langle b,x \rangle = b_1 x_1 + \cdots + b_n x_n$ is the scalar product of vectors b and x. That is, the circuit solves a minimization problem.

Note that if the circuit F is constant-free (has no constants as input gates), then the computed tropical $(\min,+)$ polynomial f is also constant-free (then $c_b = 0$ for all $b \in B$). Tropical polynomials describing *combinatorial* optimization problems are usually constant-free. For example, in the famous *MST problem* (minimum weight spanning tree problem on a given n-vertex graph G), the goal is to compute the constant-free $(\min,+)$ polynomial $f(x) = \min_{a\in A}\langle a,x\rangle$, where A is the set of characteristic 0-1 vectors of spanning trees of G (viewed as sets of their edges). In the not less prominent *assignment problem*, A is the set of characteristic 0-1 vectors of perfect matchings, etc.

We say that two tropical $(\min,+)$ polynomials f and g are *equivalent*, and write $f \equiv g$, if $f(x) = g(x)$ holds for all nonnegative input weightings $x \in \mathbb{R}^n_+$. Thus, a tropical $(\min,+)$ circuit F *computes* a given tropical $(\min,+)$ polynomial f iff the tropical $(\min,+)$ polynomial g *produced* by F is equivalent to f.

As shown in [13, Lemma 3.2], when computing tropical constant-free polynomials, we can safely restrict us to constant-free circuits: the circuit size will not increase. Namely, let $f(x) = \min_{a\in A}\langle a,x\rangle$ and $g(x) = \min_{b\in B}\ \langle x,b\rangle + c_b$ be tropical $(\min,+)$ polynomials with $A, B \subseteq \mathbb{N}^n$ and $c_b \geqslant 0$, and let $g^o(x) = \min_{b\in B}\ \langle x,b\rangle$ be the constant-free version of g.

LEMMA 1 ([13]). *If $g \equiv f$, then also $g^o \equiv f$.*

Proof. Since the constants c_b are nonnegative, we clearly have $g^o(x) \leqslant g(x) = f(x)$ for all input weightings $x \in \mathbb{R}^n_+$. So, it remains to show that $f(x) \leqslant g^o(x)$ holds for all $x \in \mathbb{R}^n_+$, as well. To show this, we will exploit the fact that $f(\lambda x) = \lambda \cdot f(x)$ and $g^o(\lambda x) = \lambda \cdot g^o(x)$ hold for every scalar $\lambda \in \mathbb{R}$. Assume for the sake of contradiction that $f(x_0) > g^o(x_0)$ holds for some input weighting $x_0 \in \mathbb{R}^n_+$. Then the difference $d = f(x_0) - g^o(x_0)$ is positive. We can assume that the constant $c := \max_{b\in B} c_b$ is also positive, for otherwise, there would be nothing to prove. Take the scalar $\lambda := 2c/d > 0$. Since $g^o(x_0) = f(x_0) - d$, we obtain $g(\lambda x_0) \leqslant g^o(\lambda x_0) + c = \lambda \cdot g^o(x_0) + c = \lambda[f(x_0) - d] + c = f(\lambda x_0) - c$, which is strictly smaller than $f(\lambda x_0)$, a contradiction with $f(x) = g(x)$ for all $x \in \mathbb{R}^n_+$. □

Sets B of "exponent" vectors produced by constant-free tropical $(\min,+)$ circuits have the following properties (which are even stronger than those for monotone Boolean circuits, as given by Claim 1).

LEMMA 2. *Let $f_A(x) = \min_{a\in A}\langle a,x\rangle$ and $f_B(x) = \min_{b\in B}\langle b,x\rangle$ be $(\min,+)$ polynomials, where $A \subseteq \{0,1\}^n$ is an antichain and $B \subseteq \mathbb{N}^n$. The following assertions are equivalent:*

(i) *$f_A(x) = f_B(x)$ holds for all $x \in \{0,1,n+1\}^n$;*
(ii) *$A \subseteq B \subseteq A^{\uparrow}$.*

Proof. The (ii) ⇒ (i) direction is simple, and even holds for all input weightings $x \in \mathbb{R}^n_+$. Indeed, since the input weights $x \in \mathbb{R}^n_+$ are nonnegative, $A \subseteq B$ implies $f_A(x) \geqslant f_B(x)$, while $B \subseteq A^{\uparrow}$ implies $f_A(x) \leqslant f_B(x)$.

To show the (i) ⇒ (ii) direction, suppose that $f_A(x) = f_B(x)$ holds for all input weightings $x \in \{0,1,n+1\}^n$. To show the inclusion $B \subseteq A^{\uparrow}$, take an arbitrary vector $b \in B$, and consider the weighting $x \in \{0,1\}^n$ such that $x_i := 0$ for $i \in \sup(b)$, and $x_i := 1$ for $i \notin \sup(b)$. Take a vector $a \in A$ on which the minimum $f_A(x) = \langle a,x\rangle$ is achieved. Then $\langle a,x\rangle = f_A(x) = f_B(x) \leqslant \langle b,x\rangle = 0$. Thus, $\sup(a) \subseteq \sup(b)$. Since $b \in \mathbb{N}^n$ and a is a 0-1 vector, this yields $b \geqslant a$, as desired.

To show the inclusion $A \subseteq B$, take an arbitrary vector $a \in A$, and consider the weighting $x \in \{1,n+1\}^n$ with $x_i := 1$ for all $i \in \sup(a)$ and $x_i := n+1$ for

all $i \notin \sup(a)$. Take a vector $b \in B$ for which $\langle b, x\rangle = f_B(x)$ holds. Let us first show that then $\sup(b) = \sup(a)$ must hold as well. On the weighting x, we have $\langle b, x\rangle = f_B(x) = f_A(x) \leqslant \langle a, x\rangle = \langle a, a\rangle \leqslant n$. If $b_i \geqslant 1$ held for some $i \notin \sup(a)$, then we would have $\langle b, x\rangle \geqslant b_i x_i = b_i(n+1) > n$, a contradiction. Thus, the inclusion $\sup(b) \subseteq \sup(a)$ holds. Since $B \subseteq A^{\uparrow}$, there is a vector $a' \in A$ such that $a' \leqslant b$. Hence, $\sup(a') \subseteq \sup(b) \subseteq \sup(a)$. Since both a and a' are 0-1 vectors, this yields $a' \leqslant a$. Since the set A is an antichain, we have $a' = a$ a and the equality $\sup(b) = \sup(a)$ follows. On this particular weighting x, we have $\langle b, x\rangle = \langle b, a\rangle$. Hence $\langle a, b\rangle = \langle b, x\rangle = f_B(x) = f_A(x) \leqslant \langle a, a\rangle$ which, together with $\sup(b) = \sup(a)$ and $b \in \mathbb{N}^n$ yields $b = a$. Thus, our vector $a \in A$ belongs to the set B, as desired. □

Together with Claim 1, Lemma 2 allows us to show that the power of tropical $(\min, +)$ circuits solving 0/1 optimization problem is the *same* as that of monotone read-once Boolean $(\vee, \wedge)$ circuits. For a finite set $A \subseteq \mathbb{N}^n$ of vectors, let

$$\begin{aligned}\mathsf{Min}(A) := \text{ min size of a } (\min, +) \text{ circuit solving the minimization}\\ \text{problem } g(x) = \min_{a\in A}\langle a, x\rangle \text{ on } A.\end{aligned}$$

THEOREM 1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function, and $A = A_f \subseteq f^{-1}(1)$ be the set of its lowest ones. Then $\mathsf{Min}(A) = \mathsf{B}_1(f)$.*

Proof. Our Boolean function f is of the form $f(x) = \bigvee_{a\in A}\bigwedge_{i\in\sup(A)} x_i$. To show the inequality $\mathsf{Min}(A) \leqslant \mathsf{B}_1(f)$, take a monotone read-1 $(\vee, \wedge)$ circuit F of size $s = \mathsf{B}_1(f)$ computing the Boolean function f, and let $B \subseteq \mathbb{N}^n$ be the set of exponent vectors produced by F. By Claim 1, we have $B \subseteq A^{\uparrow}$. Since the circuit F is a read-1 circuit, we also have the inclusion $A \subseteq B$. The tropical $(\min, +)$ version F' of F (obtained by replacing OR gates with min gates, and AND gates with addition gates) produces the same set B of "exponent" vectors. Since the inclusions $A \subseteq B \subseteq A^{\uparrow}$ hold, Lemma 2 implies that the circuit F' solves the minimization problem $g(x) = \min_{a\in A}\langle a, x\rangle$ on A. Hence, $\mathsf{Min}(A) \leqslant s = \mathsf{B}_1(f)$ holds.

To show the inequality $\mathsf{B}_1(f) \leqslant \mathsf{Min}(A)$, take a tropical $(\min, +)$ circuit F of size $s = \mathsf{Min}(A)$ solving the minimization problem $g(x) = \min_{a\in A}\langle a, x\rangle$ on the set A, and let $B \subseteq \mathbb{N}^n$ be the set of exponent vectors produced by the circuit F. By Lemma 1, we can assume that the circuit F is constant-free. So, Lemma 2 gives us the inclusions $A \subseteq B \subseteq A^{\uparrow}$. The Boolean version F' of F (obtained by replacing min gates by OR gates, and addition gates by AND gates) produces the same set B of "exponent" vectors. Together with Claim 1 and the definition of read-once circuits, inclusions $A \subseteq B \subseteq A^{\uparrow}$ imply that F' is a read-once circuit and computes the Boolean function f □

3.2. From read-once Boolean to arithmetic circuits. Say that two (arithmetic) polynomials with positive coefficients are *similar* if they have the same monomials (with apparently different coefficients). For a set $A \subseteq \mathbb{N}^n$ of vectors, let

$$\begin{aligned}\mathsf{Arith}(A) := \text{ min size of a monotone arithmetic circuit computing}\\ \text{a polynomial similar to } f(x) = \textstyle\sum_{a\in A}\prod_{i=1}^{n} x_i^{a_i}.\end{aligned}$$

Let the *degree* of a vector $b \in \mathbb{N}^n$ be the sum $b_1 + \cdots + b_n$ of its entries. For a set $B \subseteq \mathbb{N}^n$ of vectors, its *lower envelope* $\lfloor A\rfloor \subseteq A$ consists of vectors of smallest degree. The *lower envelope* of a polynomial $P(x) = \sum_{b\in B} c_b \prod_{i=1}^{n} x_i^{b_i}$ is the polynomial $\sum_{b\in\lfloor B\rfloor} c_b \prod_{i=1}^{n} x_i^{b_i}$. Jerrum and Snir [10, Theorem 2.4] observed that, by appropriately discarding some addition gates, every monotone arithmetic circuit computing a

polynomial can be transformed to a monotone arithmetic circuit computing its lower envelope.

LEMMA 3 (Envelope lemma [10]). *For every $A \subseteq \mathbb{N}^n$, $\mathsf{Arith}(\lfloor A \rfloor) \leqslant \mathsf{Arith}(A)$.*

Proof. This follows from simple properties of envelopes. The degree of a sum of two vectors is the sum of their degrees. Thus, $\lfloor A + B \rfloor = \lfloor A \rfloor + \lfloor B \rfloor$ for the Minkowski sum of sets of vectors. Second, for the union we have $\lfloor A \cup B \rfloor = \lfloor A \rfloor$ if the minimum degree of a vector in A is smaller than the minimum degree of a vector in B, $\lfloor A \cup B \rfloor = \lfloor B \rfloor$ if the minimum degree of a vector in B is smaller than the minimum degree of a vector in A, and $\lfloor A \cup B \rfloor = \lfloor A \rfloor \cup \lfloor B \rfloor$ otherwise.

Thus, given an arithmetic $(+,\times)$ circuit producing a polynomial P, we can obtain a $(+,\times)$ circuit producing the lower envelope of P by appropriately discarding some of the edges entering addition $(+)$ gates; discarding an edge (w,v) entering an addition gate $v = u + w$ means: delete that edge, delete the $+$ operation labeling the gate v, and contract the other edge (u,v). ∎

A set $A \subseteq \{0,1\}^n$ is *homogeneous* if all vectors of A have the same number of 1s. A monotone Boolean function f is *homogeneous* if the set $A_f \subseteq f^{-1}(1)$ of lowest ones of f is homogeneous (all prime implicants of f have the same number of variables). Note that then $\lfloor A_f \rfloor = A_f$.

THEOREM 2. *For every monotone Boolean function f, we have*

$$\mathsf{Arith}(\lfloor A_f \rfloor) \leqslant \mathsf{B}_1(f) \leqslant \mathsf{Arith}(A_f)\,.$$

In particular, if f is homogeneous, then $\mathsf{B}_1(f) = \mathsf{Arith}(A_f)$.

Proof. Let $A := A_f \subseteq f^{-1}(1)$ be the set of lowest ones of f. To show the first inequality $\mathsf{Arith}(\lfloor A_f \rfloor) \leqslant \mathsf{B}_1(f)$, let F be a monotone read-once Boolean $(\vee,\wedge)$ circuit of size $s = \mathsf{B}_1(f)$ computing f. Let also $P(x) = \sum_{b\in B} c_b \prod_{i=1}^n x_i^{b_i}$ be the polynomial computed by the arithmetic $(+,\times)$ version of the circuit F. Since F is a read-once circuit, we know that the inclusions $A \subseteq B \subseteq A^{\uparrow}$ hold. This yields $\lfloor B \rfloor = \lfloor A \rfloor$. Thus, the polynomial $Q(x) = \sum_{a\in\lfloor A\rfloor} c_a \prod_{i=1}^n x_i^{a_i}$ is the lower envelope of the polynomial P. Since the polynomial P is computed the arithmetic circuit F' of size s, we have $\mathsf{Arith}(B) \leqslant s$, and Lemma 3 yields $\mathsf{Arith}(\lfloor A \rfloor) = \mathsf{Arith}(\lfloor B \rfloor) \leqslant \mathsf{Arith}(B) \leqslant s$.

To show the inequality $\mathsf{B}_1(f) \leqslant \mathsf{Arith}(A_f)$, let F be a monotone arithmetic $(+,\times)$ circuit of size $s = \mathsf{Arith}(A)$ computing a polynomial similar to $P(x) = \sum_{a\in A} \prod_{i=1}^n x_i^{a_i}$. Thus, A is the set of exponent vectors produced by F. Convert F to a monotone Boolean $(\vee,\wedge)$ circuit: replace all nonzero constant inputs of F by constant 1, every addition gate by an OR gate, and every multiplication gate by an AND gate. The resulting Boolean circuit F' produces the same set A of exponent vectors. Hence, F' computes our Boolean function $f(x) = \bigvee_{a\in A} \bigwedge_{i\in\sup(a)} x_i$, as desired. ∎

3.3. Some explicit lower bounds. Currently, strong (even exponential) lower bounds on $\mathsf{Arith}(A)$ are known for many explicit homogeneous sets $A \subseteq \{0,1\}^n$, starting from the classical bounds by Schnorr [29], Valiant [31], Jerrum and Snir [10], and Gashkov [6]. Together with Theorems 1 and 2, these bounds are also lower bounds on the size of monotone Boolean read-1 $(\vee,\wedge)$ circuits computing the corresponding Boolean functions $f(x) = \bigvee_{a\in A} \bigwedge_{i\in\sup(a)} x_i$, and on the size of tropical $(\min,+)$ circuits solving the corresponding minimization problems $f(x) = \min_{a\in A} \sum_{i=1}^n a_i x_i$. We only mention some of known lower bounds on $\mathsf{Arith}(A)$.

The Schnorr bound. A set $A \subseteq \mathbb{N}^n$ of vectors is *cover-free* if $a + b \geqslant c$ with $a, b, c \in A$ implies $c \in \{a, b\}$. Schnorr [29] has proved that

$$\mathsf{Arith}(A) \geqslant |A| - 1 \tag{3.1}$$

holds for every cover-free set $A \subseteq \mathbb{N}^n$.

Example 3.2 (Cliques). The monotone Boolean function $f = \mathrm{CLIQUE}_{n,k}$ accepts a subgraph of K_n iff it contains a k-clique. Since all k-cliques have the same number $\binom{k}{2}$ of edges, the function f is homogeneous. The set $A = A_f$ of lowest ones of this function consists of characteristic 0-1 vectors of all $|A| = \binom{n}{k}$ k-cliques (viewed as sets of their edges). Since all k-cliques have the same number $\binom{k}{2}$ of edges, the set A is homogeneous, and Theorems 1 and 2 yield $\mathsf{Min}(A) = \mathsf{B}_1(f) \geqslant \mathsf{Arith}(A)$.

On the other hand, Schnorr's bound (3.1) yields $\mathsf{Arith}(A) \geqslant \binom{n}{k} - 1$. To show this, it is enough to verify that the set A is cover-free. To show this, assume the opposite, i.e., that the union of some two k-cliques contains some third k-clique. Since each k-clique has the same number k of nodes, the latter clique must then have a node u not in the first clique and a node v not in the second clique. If $u = v$ then the node u is not covered, and if $u \neq v$ then the edge $\{u, v\}$ is not covered by the union of the first two cliques, a contradiction. Thus, A is cover-free. □

The Hyafil–Valiant–Jerrum–Snir bound. A set $A \subseteq \mathbb{N}^n$ is *homogeneous* of degree m if $a_1 + \cdots + a_n = m$ holds for all vectors $a \in A$. A sumset $X + Y = \{x + y \colon x \in X, y \in Y\}$ of two sets of vectors $X, Y \subseteq \mathbb{N}^n$ is *r-homogeneous* if the set X is homogeneous of degree r. Let $h_r(A)$ be the maximum of $|X + Y|$ over all sets $X, Y \subseteq \mathbb{N}^n$ such that X is r-homogeneous and $X + Y \subseteq A$ holds. By viewing polynomials as sets of their exponent vectors, the following lower bound was implicitly proved by Hyafil [9], Valiant [31], and Jerrum and Snir [10]: if $A \subseteq \mathbb{N}^n$ is homogeneous of degree $m \geqslant 3$, and if $h_r(A) \leqslant h$ holds for all $m/3 \leqslant r \leqslant 2m/3$, then

$$\mathsf{Arith}(A) \geqslant |A|/h\,. \tag{3.2}$$

Example 3.3 (Perfect matchings). The *perfect matching* function is a monotone Boolean function $f = \mathrm{Match}_n$ which accepts a subgraph of $K_{n,n}$ iff it contains a perfect matching. Since perfect matchings have the same number n of edges, this function is homogeneous. The set $A = A_f$ of lowest ones of this function consists of $|A| = n!$ characteristic 0-1 vectors of all perfect matchings (viewed as sets of their edges). Since the set A is homogeneous, Theorems 1 and 2 yield $\mathsf{Min}(A) = \mathsf{B}_1(f) \geqslant \mathsf{Arith}(A)$.

On the other hand, the bound (3.2) yields $\mathsf{Arith}(A) \geqslant \binom{n}{n/3}$. To show this, it is enough to show that $h_r(A) \leqslant n!\binom{n}{r}^{-1}$ holds for every $n/3 \leqslant r \leqslant 2n/3$. So, take any r-homogeneous sumset $X + Y$ such that $X + Y \subseteq A$. Every matching with r edges can be contained in at most $(n - r)!$ perfect matchings. Hence, for every $x \in X$, we have $|Y| = |x + Y| \leqslant (n - r)!$. Similarly, every vector $y \in Y$ corresponds to a matching with $n - r$ edges, and we have $|X| = |X + y| \leqslant r!$. Thus, $|X + Y| \leqslant (n - r)!r! = n!\binom{n}{r}^{-1}$. Since $\binom{n}{r} \geqslant \binom{n}{n/3}$ for every $n/3 \leqslant r \leqslant 2n/3$, this yields $h_r(A) \leqslant h := n!\binom{n}{n/3}^{-1}$, and (3.2) yields the claimed lower bound $\mathsf{Arith}(A) \geqslant |A|/h = \binom{n}{n/3}$. □

The Gashkov–Sergeev bound. A set $A \subseteq \mathbb{N}^n$ of vectors is (k, l)-*thin* if the following holds for any two subsets $X, Y \subseteq \mathbb{N}^n$ of vectors: if $X + Y \subseteq A$ then $|X| \leqslant k$ or $|Y| \leqslant l$. In other words, a set $A \subseteq \mathbb{N}^n$ is (k, l)-thin if for any $k + 1$ distinct vectors $a_1, \ldots, a_{k+1} \in \mathbb{N}^n$, the system of relations $a_1 + x \in A$, $a_2 + x \in A$, …, $a_{k+1} + x \in A$

has at most l distinct solutions $x \in \mathbb{N}^n$. The interpretation of thin sets $A \subseteq \mathbb{N}^n$ in terms of *graphs* is the following. Associate with A the (infinite) bipartite graph G_A where two vertices $x \in \mathbb{N}^n$ and $y \in \mathbb{N}^n$ are adjacent iff $x + y \in A$. Then A is (k,l)-thin iff G_A contains no copy of a complete bipartite $(k+1) \times (l+1)$ graph as a subgraph.

Gashkov and Sergeev [7] have proved that

$$\mathsf{Arith}(A) \geqslant |A|/\max\{k^3, l^2\} - 1 \tag{3.3}$$

holds for any (k,l)-thin set $A \subseteq \mathbb{N}^n$. For $k = l$, this bound was proved much earlier by Gashkov [6]. Using a fairly elementary "bottlenecks counting" argument, a lower bound $\mathsf{Arith}(A) \geqslant |A|/2lk^2$ was proved in [12] for extended monotone arithmetic circuits, where instead of single variables, any polynomials with at most k monomials can be used as inputs.

Example 3.4 (Norm sets). Let q be a prime-power, $t \geqslant 2$ an integer, and consider the field $\mathbb{F} = \mathrm{GF}(q^t)$ with q^t elements. The *norm function* is a mapping $\mathrm{N} : \mathrm{GF}(q^t) \to \mathrm{GF}(q)$ given by $\mathrm{N}(a) = a \cdot a^q \cdots a^{q^{t-1}} = a^{(q^t-1)/(q-1)}$. Consider the set $A = \{a \in \mathbb{F} \colon \mathrm{N}(a) = 1\}$ of all elements of unit norm. It is known (see, e.g., [18]) that $|A| = (q^t - 1)/(q-1)$. Kollár, Rónyai and Szabó [14, Theorem 3.3] proved that, for every t distinct elements $a_1, \ldots, a_t$ of $\mathbb{F}$, the system of equations $\mathrm{N}(a_1 + x) = 1$, $\mathrm{N}(a_2 + x) = 1$,..., $\mathrm{N}(a_t + x) = 1$ has at most $t!$ solutions $x \in \mathbb{F}$. Hence, the set A is $(t, t!)$-thin over the group $(\mathbb{F}, +)$. Now let $q := 2^r$ and $n := rt$. By viewing elements of $\mathrm{GF}(2^n)$ as vectors in $\{0,1\}^n$, we obtain an explicit *norm set* $A_{n,t} \subseteq \{0,1\}^n$ of $|A_{n,t}| = (2^{rt} - 1)/(2^r - 1) \geqslant 2^{r(t-1)} = 2^{n-n/t}$ vectors which is $(t, t!)$-thin over $(\mathbb{F}, +)$, and hence, also over the semigroup $(\mathbb{N}^n, +)$. Thus, (3.3) yields the lower bound $\mathsf{Arith}(A_{n,t}) \geqslant 2^{n-n/t}/(t!)^2 \geqslant 2^{n-n/t-2t\log t}$. For $t = \sqrt{n}$, we obtain[1] $\mathsf{Arith}(A_{n,t}) \geqslant 2^{n-2\sqrt{n}\log n}$.

The norm sets are not necessarily homogeneous, but sets of 0-1 vectors can be easily made homogeneous by just doubling the length of vectors. Namely, let $n = 2m$ and define the *homogeneous extension* of a set $B \subseteq \{0,1\}^m$ to be the set $A = \{(b, \underline{b}) \colon b \in B\} \subseteq \{0,1\}^n$, where $\underline{b}$ denotes the componentwise negation of a 0-1 vector b. For example, if $b = (0,0,1,0,1)$, then $\underline{b} = (1,1,0,1,0)$. Note that the set A is already homogeneous because every its vector has exactly m ones. It is easy to verify that if B is (k,l)-thin, then also A is (k,l)-thin. Thus, if m is a square of an integer, and $A \subseteq \{0,1\}^n$ with $n = 2m$ is the homogeneous extension of the norm-set $A_{m,t} \subseteq \{0,1\}^m$ for $t = \sqrt{m}$, then $\mathsf{Arith}(A) \geqslant 2^{n/2-o(n)}$. □

4. Multilinear Boolean circuits. Due to the lack of strong lower bounds for general (non-monotone) arithmetic $(+, \times, -)$ circuits, and because they seem to be the most intuitive circuits for multilinear functions, a successful approach has been to consider a restriction called "multilinearity" first defined by Nisan and Wigderson [21].

Recall that a polynomial in the ring $\mathbb{R}[x_1, \ldots, x_n]$ is multilinear if in each of its monomials, the power of every input variable is at most one. An arithmetic $(+, \times, -)$ circuit is *multilinear* if the polynomial *function* computed (not necessarily the formal polynomial produced—in this case, the circuit would be *syntactically multilinear*) at each gate of the circuit is multilinear. Raz [24] proved that any multilinear arithmetic *formula* (a circuit with fanout-1 gates) for the permanent or the determinant of an $n \times n$ matrix has size super-polynomial in n. Furthermore, Raz [23] proved a super-polynomial separation between the size of multilinear arithmetic *circuits* and *formulas*.

[1] To our best knowledge, this is the *highest* known lower bound on the monotone arithmetic circuit complexity of an explicit multilinear polynomial.

Proving super-polynomial lower bounds for the size of multilinear arithmetic *circuits* is an open problem; see Shpilka and Yehudayoff [30] for a survey of further progress concerning multilinear arithmetic circuits.

Due to the lack of strong (even non-linear) lower bounds for general DeMorgan $(\vee, \wedge, \neg)$ circuits, the multilinearity restriction was also imposed on such circuits. Recall that a DeMorgan $(\vee, \wedge, \neg)$ circuit $F(x)$ on a vector $x = (x_1, \ldots, x_n)$ of variables has fanin-2 AND and OR gates, and inputs are the variables $x_1, \ldots, x_n$ and their negations $\overline{x}_1, \ldots, \overline{x}_n$. As before, the *size* of a circuit is the total number of gates in it. A *monotone* Boolean circuit is a DeMorgan circuit without negated input literals as inputs.

A DeMorgan $(\vee, \wedge, \neg)$ circuit F is *syntactically multilinear* if the two subcircuits rooted at inputs of any AND gate have no input literals of the same variable in common. For example, the circuit $F = (x \vee x\overline{y})y$ is *not* syntactically multilinear.

Krieger [15] has shown that syntactically multilinear $(\vee, \wedge, \neg)$ circuits generalize non-deterministic (syntactically) read-once branching programs. He also showed that minimal syntactically multilinear $(\vee, \wedge, \neg)$ circuits computing *monotone* functions are *monotone*. It is clear that exponent vectors produced by arithmetic $(+, \times)$ versions of monotone semantically multilinear $(\vee, \wedge)$ circuits F are 0-1 vectors. So, the "lower envelopes trick" of Jerrum and Snir [10] (Lemma 3) implies that any such circuit F computing a Boolean function f must have at least $\mathsf{Arith}(\lfloor A_f \rfloor)$ gates.

Actually, a lower bound of the form $2^{n/3}$ for syntactically multilinear DeMorgan $(\vee, \wedge, \neg)$ circuits computing an explicit n-variate Boolean function was already proved by Kuznetsov [16]. He proved such a bound for so-called DeMorgan $(\vee, \wedge, \neg)$ circuits *without zero paths*, and every syntactically multilinear circuit is a circuit without zero paths.

These results made it clear that *syntactic* multilinearity is a too severe restriction. Ponnuswami and Venkateswaran [22] relaxed this restriction to "semantic" multilinearity. The *i-th neighbor* of a vector $a \in \{0,1\}^n$ if the vector $a' \in \{0,1\}^n$ differing from a in only the i-th position. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ *depends* on the i-th variable x_i if $f(a') \neq f(a)$ holds for the i-th neighbor a' of some vector $a \in \{0,1\}^n$.

Definition 2 (Multilinear circuits). A DeMorgan $(\vee, \wedge, \neg)$ circuit F is *multilinear* if the Boolean functions computed at the inputs to any AND gate depend on disjoint sets of variables.

Note that multilinear DeMorgan circuits are not necessarily syntactically multilinear: in the former circuits, some paths in the circuit from the same input literal *can* reach both these gates. For example, $g = x \vee x\overline{y}$ and $h = y$ depend on disjoint sets of variables, because g does not depend on y.

Our goal in the rest of the paper is to reduce multilinear $(\vee, \wedge, \neg)$ circuits to read-1 $(\vee, \wedge)$ circuits. Namely, we will show (Corollary 1) that if $F(x, \overline{x})$ is a multilinear $(\vee, \wedge, \neg)$ circuit computing a Boolean function $f(x)$, then the monotone $(\vee, \wedge)$ subcircuit $F(x, \vec{1})$ of F obtained by replacing each negated input literal by constant 1 is a read-1 circuit and computes the monotone Boolean function f^∇ such that $f^\nabla(x) = 1$ iff $f(z) = 1$ holds for some $z \leqslant x$; note that $f^\nabla = f$ holds for *monotone* functions f.

4.1. Monotone multilinear circuits. Recall that a *lowest one* of a (not necessarily monotone) Boolean function f is a vector a such that $f(a) = 1$ but $f(b) = 0$ for all $b \leqslant a$, $b \neq a$. Let, as before, $A_f \subseteq f^{-1}(1)$ denote the set of all lowest ones of f.

The following simple claim shows how the sets A_g of lowest ones of functions g computed at the gates of a DeMorgan circuit do behave. For two sets $A, B \subseteq \{0,1\}^n$

of vectors, let $A \vee B := \{a \vee b \colon a \in A, b \in B\}$, where $a \vee b$ is the componentwise OR of vectors a and b.

Claim 2. *For any Boolean functions $g, h : \{0,1\}^n \to \{0,1\}$ we have the inclusions $A_{g \vee h} \subseteq A_g \cup A_h$ and $A_{g \wedge h} \subseteq A_g \vee A_h$.*

Proof. Let first $f = g \vee h$, and take an arbitrary lowest one $a \in A_f$ of f. Then $g(a) = 1$ or $h(a) = 1$, and both $g(b) = 0$ and $h(b) = 0$ hold for every vector $b \leqslant a$, $b \neq a$. Thus, either $a \in A_g$ or $a \in A_h$, as desired.

Now let $f = g \wedge h$, and take an arbitrary lowest one $a \in A_f$ of f. Then $g(a) = h(a) = 1$. Let $b \leqslant a$ be some lowest one of g with the smallest number of 1s, and $c \leqslant a$ be some lowest one of h with the smallest number of 1s. Then $b \vee c \leqslant a$. We actually have an equality $a = b \vee c$ because otherwise, a would not be a *lowest* one of f. Hence, $a \in A_g \vee A_h$, as desired. □

The dependency on variables is determined by the supports of lowest ones of the corresponding Boolean functions. As before, the *support* of a vector $x \in \mathbb{N}^n$ is the set $\sup(x) := \{i \colon x_i \neq 0\}$ of its nonzero positions.

Claim 3. *A Boolean function $f(x_1, \ldots, x_n)$ depends on a variable x_i if and only if $i \in \sup(a)$ for some $a \in A_f$.*

Proof. The "if" direction follows directly from the definition of lower ones. For the "only if" direction, take any vector $x \in f^{-1}(1)$, and assume that $f(x') = 0$ folds for the i-th neighbor of some vector $x \in f^{-1}(1)$. Take a lower one $a \in A_f$ with $a \leqslant x$. If $i \notin \sup(a)$ held, then $a \leqslant x'$ would also hold, meaning that $f(x') = 1$, a contradiction. □

Lemma 4. *Monotone multilinear Boolean circuits are read-1 circuits.*

Proof. Let F be a monotone multilinear Boolean circuit computing a monotone Boolean function f. Let $B \subseteq \mathbb{N}^n$ be the set of vectors produced by F. To show that F is a read-1 circuit, we have to show the inclusion $A_f \subseteq B$, i.e., that every lowest one $a \in A$ of f is produced by the circuit F.

Let F_1 and F_2 be the subcircuits of F whose output gates enter the output gate of F, and let f_1 and f_2 be the monotone Boolean functions computed by these subcircuits. Let also $B_1 \subseteq \mathbb{N}^n$ be the set of vectors produced by the subcircuit F_1, and $B_2 \subseteq \mathbb{N}^n$ be the set of vectors produced by the subcircuit F_2. We argue by induction on the number s of gates in F.

The basis case $s = 1$ is trivial, because then $F_1 = x_i$ and $F_2 = x_j$ for some $i, j \in [n]$ and, if $F = F_1 \wedge F_2$, then $i \neq j$ due to the multilinearity of the circuit F.

Now suppose that the lemma holds for all monotone Boolean circuits of size at most $s - 1$, and let F be a monotone Boolean circuit of size s. Since the circuit F is multilinear, both subcircuit F_1 and F_2 are also multilinear. Since each of F_1 and F_2 has at most $s - 1$ gates, the lemma holds for both these subcircuits. Thus, both inclusions $A_{f_1} \subseteq B_1$ and $A_{f_2} \subseteq B_2$ hold. The case when $F = F_1 \vee F_2$ is trivial: then $B = B_1 \cup B_2$ and, by Claim 2, $A_f \subseteq A_{f_1} \cup A_{f_2}$. So, the desired inclusion $A_f \subseteq B$ follows directly from the induction hypothesis.

Now let $F = F_1 \wedge F_2$. Then $B = B_1 + B_2$ (Minkowski sum) and, by Claim 2, $A_f \subseteq A_{f_1} \vee A_{f_2}$. To show the desired inclusion $A_f \subseteq B$, take an arbitrary vector $a \in A$. Then $a = x \vee y$ (a componentwise OR) for some vectors $x \in A_{f_1}$ and $y \in A_{f_2}$. Since the circuit F is multilinear, Claim 3 implies that $\sup(x) \cap \sup(y) = \emptyset$, and thus, vector $a = x + y$ is the sum of vectors x and y. By the induction hypothesis, we have that $x \in B_1$ and $y \in B_2$. So, vector $a = x + y$ belongs to the sumset $B = B_1 + B_2$, as desired. □

4.2. Non-monotone multilinear circuits. The *upward closure* of a Boolean function $f(x)$ is the monotone Boolean function

$$f^{\triangledown}(x) := \bigvee_{z \leqslant x} f(z)\,.$$

Note that $f^{\triangledown}(x) = 1$ iff $f(z) = 1$ holds for some vector $z \leqslant x$, which holds iff $x \geqslant a$ for some $a \in A_f$, that is, iff vector x contains at least one lowest one of f. For example, the set of lowest ones of the parity function $f(x) = x_1 \oplus \cdots \oplus x_n$ consists of n vectors, each with exactly one 1. So, $f^{\triangledown}(x) = x_1 \vee \cdots \vee x_n$. In particular, $f^{\triangledown} = f$ holds for every *monotone* Boolean function f. Note that $A_f = A_{f^{\triangledown}}$ holds for every Boolean function f, that is, both functions f and $f^{\triangledown}$ have the same lowest ones.

The following lemma shows how the upward closures $g^{\triangledown}$ of functions g computed at the gates of a *multilinear* DeMorgan circuit do behave.

LEMMA 5. *If Boolean functions $g, h : \{0,1\}^n \to \{0,1\}$ depend on disjoint sets of variables, then $(g \vee h)^{\triangledown} = g^{\triangledown} \vee h^{\triangledown}$ and $(g \wedge h)^{\triangledown} = g^{\triangledown} \wedge h^{\triangledown}$.*

In fact, $(g \vee h)^{\triangledown} = g^{\triangledown} \vee h^{\triangledown}$ holds for *any* g and h.

Proof. Let first $f = g \vee h$. To show the inequality $f^{\triangledown} \leqslant g^{\triangledown} \vee h^{\triangledown}$, take any vector $x \in \{0,1\}^n$ for which $f^{\triangledown}(x) = 1$ holds. Then $x \geqslant a$ for some lowest one $a \in A_f$. By Claim 2, we have $a \in A_g$ or $a \in A_h$. Hence, either $g^{\triangledown}(x) = 1$ or $h^{\triangledown}(x) = 1$, as desired. To show the opposite inequality $f^{\triangledown} \geqslant g^{\triangledown} \vee h^{\triangledown}$, take any vector $x \in \{0,1\}^n$ for which $g^{\triangledown}(x) = 1$ holds. Then $g(z) = 1$ and, hence, also $f(z) = 1$ holds for some $z \leqslant x$, meaning that $f^{\triangledown}(x) = 1$, as desired.

Now let $f = g \wedge h$. Then the inequality $f^{\triangledown} \leqslant g^{\triangledown} \wedge h^{\triangledown}$ is trivial: if $f^{\triangledown}(x) = 1$, then $f(z) = 1$ for some vector $z \leqslant x$ and, hence, also $g(z) = 1$ and $h(z) = 1$ hold. So, assume that the functions g and h depend on disjoint sets of variables. We have to show that then also $g^{\triangledown} \wedge h^{\triangledown} \leqslant f^{\triangledown}$ holds. Let I_g (resp, I_h) be the set of positions $i \in [n]$ such that the function g (resp., h) depends on the i-th variable. By Claim 3, we have $I_g = \{i \in [n] \colon i \in \sup(x) \text{ for some } x \in A_g\}$, and similarly for the set I_h. By our assumption, we have $I_g \cap I_h = \emptyset$. Take now any vector $x \in \{0,1\}^n$ for which $g^{\triangledown}(x) = 1$ and $h^{\triangledown}(x) = 1$ hold. Then there is a lowest one $b \in A_g$ of g, and a lowest one $c \in A_h$ of h such that $b \leqslant x$ and $g(b) = 1$, as well as $c \leqslant x$ and $h(c) = 1$ hold. Consider the vector $a = b \vee c$. Since $I_g \cap I_h = \emptyset$, we have $\sup(b) \cap \sup(b) = \emptyset$ and, hence, the vector a is the sum $a = b + c$ of vectors b and c. Since the function g does not depend on any variable x_i with $i \notin I_g$, we have $g(a) = g(b + c) = g(b + \vec{0}) = g(b) = 1$. Similarly, $h(a) = h(b + c) = h(c + \vec{0}) = h(c) = 1$. Since $a \leqslant x$, this yields $f^{\triangledown}(x) = 1$, as desired. □

We can view every DeMorgan $(\vee, \wedge, \neg)$ circuit $F(x)$ computing a Boolean function $f(x)$ of n variables as a *monotone* $(\vee, \wedge)$ circuit $H(x, y)$ on $2n$ variables with the property that $f(x) = H(x, \overline{x})$ holds for all $x \in \{0,1\}^n$, where $\overline{x} = (\overline{x}_1, \ldots, \overline{x}_n)$ is the complement of $x = (x_1, \ldots, x_n)$. The *monotone version* of the circuit $F(x)$ is the monotone circuit $F^+(x) = H(x, \vec{1})$ obtained by replacing every negated input literal $\overline{x}_i$ with constant 1.

COROLLARY 1. *Let F be a DeMorgan circuit computing a Boolean function f. If F is multilinear, then F^+ computes $f^{\triangledown}$.*

Proof. Suppose that the circuit $F(x) = H(x, \overline{x})$ is multilinear. Upward closures of input variables x_i are the variables $x_i^{\triangledown} = x_i$ themselves, while upward closures of negated input variables $\overline{x}_i$ are constant-1 functions $\overline{x}_i^{\triangledown} = 1$. Let g and h be the Boolean functions computed at the two inputs of an arbitrary gate of F. If this is an

OR gate, then Lemma 5 yields the equality $(g \vee h)^\nabla = g^\nabla \vee h^\nabla$. If this is an AND gate then, since the circuit F is multilinear, the functions g and h depend on disjoint sets of variables, and Lemma 5 also yields the equality $(g \wedge h)^\nabla = g^\nabla \wedge h^\nabla$. Thus, in the circuit $F^+ = H(x, \vec{1})$, the upwards closures g^∇ of the functions g computed at the gates of F are computed. Since this also holds for the output gate of F, at which the function f was computed, the upward closure f^∇ of f is computed at this gate of the circuit F^+, as desired. □

Remark 4.1. The converse of Corollary 1 does not hold: the monotone version F^+ of a circuit F may compute f^∇ even though the circuit F is not multilinear. Consider, for example, the circuit $F = y\overline{z} \vee x(\overline{y} \vee \overline{x}y)$ computing the Boolean function $f(x, y, z) = x\overline{y} \vee y\overline{z}$. The upward closure of f is $f^\nabla = x \vee y$. The monotone version $F^+ = y \cdot 1 \vee x(1 \vee 1 \cdot y) = x \vee y$ of the circuit F computes f^∇. But the circuit F is not multilinear, because the functions $g = x$ and $h = \overline{y} \vee \overline{x}y$ computed at the inputs of an AND gate depend on the same variable x: say, $h(0, 1, 0) = 1$ while $h(1, 1, 0) = 0$.

For a Boolean function f, let

$\mathsf{B}_{\mathrm{lin}}(f) :=$ min size of a multilinear DeMorgan $(\vee, \wedge, \neg)$ circuit computing f.

THEOREM 3. *For every Boolean function f, we have*

$$\mathsf{B}_{\mathrm{lin}}(f) \geqslant \mathsf{B}_1(f^\nabla) \geqslant \mathsf{Arith}(\lfloor A_f \rfloor)\,.$$

If f is monotone, then $\mathsf{B}_{\mathrm{lin}}(f) \geqslant \mathsf{B}_1(f) \geqslant \mathsf{Arith}(\lfloor A_f \rfloor)$.

Proof. Let F be a multilinear DeMorgan $(\vee, \wedge, \neg)$ circuit of size $s = \mathsf{B}_{\mathrm{lin}}(f)$ computing f. Since the circuit F is multilinear, its monotone version F^+ is also multilinear and, by Corollary 1, computes the upward closure $g := f^\nabla$ of f. The (monotone) circuit F^+ has size at most s and, by Lemma 4, is a read-1 circuit. This shows the inequality $\mathsf{B}_{\mathrm{lin}}(f) \geqslant \mathsf{B}_1(g)$. The inequality $\mathsf{B}_1(g) \geqslant \mathsf{Arith}(\lfloor A_g \rfloor)$ is given by Theorem 2. Since lowest ones of a Boolean function f and of its upward closure $g = f^\nabla$ are the same, we have $\lfloor A_g \rfloor = \lfloor A_f \rfloor$ and, hence, also $\mathsf{B}_{\mathrm{lin}}(f) \geqslant \mathsf{B}_1(g) \geqslant \mathsf{Arith}(\lfloor A_g \rfloor) = \mathsf{Arith}(\lfloor A_f \rfloor)$. If the function f is monotone, then $g = f$. □

Remark 4.2. If $\mathsf{B}^+_{\mathrm{lin}}(f)$ denotes the minimum size of a multilinear *monotone* $(\vee, \wedge)$ circuit computing a (monotone) Boolean function f then Theorem 3 trivially yields $\mathsf{B}^+_{\mathrm{lin}}(f) \geqslant \mathsf{B}_{\mathrm{lin}}(f) \geqslant \mathsf{B}_1(f) \geqslant \mathsf{Arith}(\lfloor A_f \rfloor)$ for any monotone Boolean function. Using different arguments, for homogeneous functions f (those with $\lfloor A_f \rfloor = A_f$), Lingas [19] proved a slightly weaker lower bound $\mathsf{B}^+_{\mathrm{lin}}(f) \geqslant \mathsf{Arith}(A_f)/\mathcal{O}(k^2)$, where k is the number of 1s in each vector of A_f.

5. Read-2 circuits can be exponentially stronger than read-1. Theorems 1 to 3 show that monotone read-1 $(\vee, \wedge)$ circuits are *not weaker* than tropical $(\min, +)$, monotone arithmetic $(+, \times)$, and multilinear $(\vee, \wedge, \neg)$ circuits. Let us now show that already monotone read-2 $(\vee, \wedge)$ circuits can be exponentially smaller than read-1 circuits. To show this, consider the following monotone Boolean function whose inputs are Boolean $n \times n$ matrices $x = (x_{i,j})$:

$$\mathrm{Isol}_n(x) = 1 := \text{iff every row and every column of } x \text{ has at least one 1.}$$

LEMMA 6. *For $f = \mathrm{Isol}_n$, we have $\mathsf{B}_1(f) = 2^{\Omega(n)}$ but $\mathsf{B}_2(f) \leqslant 2n^2$.*

Proof. The set $A := f^{-1}(1)$ consists of all matrices $a = (a_{i,j})$ with at least one 1 in each line (row or column). The smallest number of 1s in a matrix $a \in A$ is n, and the

matrices in A with this number of 1s are permutation matrices. So, $\lfloor A \rfloor = A_g$ is the set of the lowest ones of the perfect matching function $g = \mathrm{Match}_n$, and we already know that $\mathsf{Arith}(A_g) = 2^{\Omega(n)}$ (Example 3.3); actually, using a tighter argument, Jerrum and Snir [10] have proved that $\mathsf{Arith}(A_g) \geqslant n(2^{n-1}-1)$. Together with Theorem 2, this yields $\mathsf{B}_1(f) \geqslant \mathsf{Arith}(\lfloor A \rfloor) = \mathsf{Arith}(A_g) = 2^{\Omega(n)}$.

To show $\mathsf{B}_2(f) \leqslant 2n^2$, observe that f can be computed by a trivial monotone Boolean circuit

$$F(x) = \bigwedge_{i=1}^{n} \Big(\bigvee_{j=1}^{n} x_{i,j} \Big) \bigwedge_{j=1}^{n} \Big(\bigvee_{i=1}^{n} x_{i,j} \Big)$$

of size at most $2n^2$. So, it remains to verify that F is a read-2 circuit. Let $B \subseteq \{0,1\}^{n \times n}$ be the set of exponent vectors of the polynomial

$$P(x) = \prod_{i=1}^{n} \Big(\sum_{j=1}^{n} x_{i,j} \Big) \prod_{j=1}^{n} \Big(\sum_{i=1}^{n} x_{i,j} \Big)$$

produced by the arithmetic version of the circuit F. Note that each matrix $b \in B$ is the sum $b = x + y$ of a matrix x with exactly one 1 is each row and a matrix y with exactly one 1 in each column, while each matrix $a \in A_f$ is the entry-wise OR $a = x \vee y$ of two such matrices. Since $\sup(x+y) = \sup(x \vee y)$, and since none of the matrices $b \in B$ has any entry larger than 2, the circuit F is a read-2 circuit. ∎

Currently, strong (even exponential) lower bounds for monotone read-k circuits for *any* k are known. However, proving such bounds remains a rather nontrivial task: here, we essentially have only one tool—the celebrated Method of Approximations invented by Razborov [25, 26, 27], and its subsequent symmetric versions (see, e.g., [11, Chapter 9]). The Method of Approximations is much more *involved* than lower-bound arguments used to prove lower bounds for monotone *arithmetic* circuits, including those mentioned in subsection 3.3. Moreover, the former method can be only applied to monotone Boolean functions with very special combinatorial properties.

Problem 1. Prove a super-polynomial lower bound on $\mathsf{B}_2(f)$ *without* using the Method of Approximations.

Note that monotone Boolean read-2 circuits constitute the *first* (with respect to their power) model of computation—after tropical and monotone arithmetic circuits—which *can* use multiplicative idempotence $x \wedge x = x$. So, a solution of Problem 1 could apparently shed more light in understanding the power of this idempotence in computations. Of course, this task is incomparable with the ultimate task of understanding the role of cancellations $x \wedge \overline{x} = 0$, the role of negations in Boolean circuits (see, e.g., [11, Chapter 10]). But it is hard to imagine that we can understand the latter without first having understood the former.

REFERENCES

[1] R. Bellman. On a routing problem. *Quarterly of Appl. Math.*, 16:87–90, 1958.
[2] R. Bellman. Dynamic programming treatment of the Travelling Salesman problem. *J. ACM*, 9(1):61–63, 1962.
[3] S.E. Dreyfus and R.A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.
[4] R. W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 5:345, 1962.
[5] L. R. Ford. Network flow theory. Technical Report P-923, 1956.
[6] S. B. Gashkov. On one method of obtaining lower bounds on the monotone complexity of polynomials. *Vestnik MGU, Series 1 Mathematics, Mechanics*, 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. *Sbornik: Mathematics*, 203(10):1411–1147, 2012.

[8] M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *SIAM J. on Appl. Math.*, 10:196–210, 1962.

[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. *Boolean Function Complexity: Advances and Frontiers.* Springer-Verlag, 2012.

[12] S. Jukna. Tropical complexity, Sidon sets and dynamic programming. *SIAM J. Discrete Math.*, 30(4):2064–2085, 2016.

[13] S. Jukna and H. Seiwert. Approximation limitations of pure dynamic programming. *SIAM J. on Computing*, 49(1):170–207, 2020.

[14] J. Kollár, L. Rónyai, and T. Szabó. Norm-graphs and bipartite Turán numbers. *Combinatorica*, 16(3):399–406, 1996.

[15] M. P. Krieger. On the incompressibility of monotone DNFs. *Theory Comput. Syst.*, 41(2):211–231, 2007.

[16] S. E. Kuznetzov. Circuits composed of functional elements without zero paths in the basis $\{\&, \vee, -\}$. *Izv. Vyssh. Uchebn. Zaved. Mat.*, 228(5):56–63, 1981. In Russian.

[17] A.Y. Levin. Algorithm for the shortest connection of a group of graph vertices. *Sov. Math. Dokl.*, 12:1477–1481, 1971.

[18] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1986.

[19] A. Lingas. A note on lower bounds for monotone multilinear Boolean circuits. Technical report, ECCC Report Nr. 85, 2022.

[20] E. F. Moore. The shortest path through a maze. In *Proc. Internat. Sympos. Switching Theory*, volume II, pages 285–292, 1957.

[21] N. Nisan and A. Wigderson. Lower bounds on arithmetic circuits via partialderivatives. *Comput. Complexity*, 6(3):217–234, 1997.

[22] A. K. Ponnuswami and H. Venkateswaran. Monotone multilinear boolean circuits for bipartite perfect matching require exponential size. volume 3328 of *Lect. Notes in Comput. Sci.*, pages 460–468. Springer, 2004.

[23] R. Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(6):121–135, 2006.

[24] R. Raz. Multi-linear formulas for Permanent and Determinant are of super-polynomial size. *J. of the ACM*, 56(2):1–17, 2009.

[25] A. A. Razborov. Lower bounds for the monotone complexity of some boolean functions. *Soviet Math. Dokl.*, 31:354–357, 1985.

[26] A. A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Math. Notes of the Acad. of Sci. of the USSR*, 37(6):485–493, 1985.

[27] A. A. Razborov. On the method of approximations. In *Proc. of 21st Ann. ACM Symp. on Theory of Computing*, pages 167–176. ACM, 1989.

[28] B. Roy. Transitivité at connexité. *C. R. Acad. Sci. Paris*, 249:216–218, 1959. in French.

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

[30] A. Shpilka and A. Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

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

[32] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il