NOTES ON BOOLEAN READ-K CIRCUITS

STASYS JUKNA*

Abstract. A monotone Boolean $(\vee, \wedge)$ circuit F computing a Boolean function f is a read-k circuit if the polynomial produced (purely syntactically) by the arithmetic $(+, \times)$ version of F 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 $(\vee, \wedge)$ circuit is a read-k circuit for some k.

We first show that already read-1 circuits are interesting in the context of dynamic programming: tropical $(\min, +)$ circuits solving 0/1 optimization problems have the same power as Boolean read-1 circuits, and that monotone read-1 $(\vee, \wedge)$ circuits computing homogeneous Boolean functions are not stronger than monotone arithmetic circuits. Then we show that already read-2 circuits can be exponentially smaller than read-1 circuits. Finally, we show that so-called (semantically) multilinear DeMorgan $(\vee, \wedge, \neg)$ circuits computing monotone Boolean functions are not stronger than monotone read-1 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 obtained monotone arithmetic $(+, \times)$ circuit will produce (as a formal expression) a unique polynomial P with the following two properties: (i) every monomial of P contains all variables of at least one prime implicant of f, and (ii) for every prime implicant p of f, there at least one monomial t (a "shadow" of p) in P with the same set of variables as p. We call the (Boolean) circuit F a *read-k* circuit if in at least one shadow of every prime implicant, each variable has degree $\leqslant k$. There are no restrictions on other monomials of the produced polynomial P. Every monotone $(\vee, \wedge)$ circuit of size is a read-k circuit for some k.

Our main interest in read-k circuits is that already 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. Notable examples of pure DP algorithms are the well-known Bellman–Ford–Moore shortest s-t path algorithm [1, 4, 17], the Roy–Floyd–Warshall all-pairs shortest paths algorithm [22, 3, 25], the Bellman–Held–Karp travelling salesman algorithm, the Dreyfus–Levin–Wagner Steiner tree algorithm [2, 7], and many others.

Tropical $(\min, +)$ circuits constitute a rigorous mathematical model for pure DP algorithms. Namely, such an algorithm is just a special (recursively constructed) tropical circuit. In particular, lower bounds on the size of tropical circuit are also lower bounds on the minimum possible number of operations used by pure DP algorithms.

We first show that tropical $(\min, +)$ circuits solving 0/1 minimization problems have the *same* power as monotone Boolean read-1 circuits (Theorem 1).

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 [19, 20, 21], and its subsequent symmetric versions (see, e.g., [10, Chapter 9]). Not surprisingly, this task is much easer for $k = 1$: then we actually are in a (much simpler) monotone *arithmetic* world. Namely, Jerrum and Snir [9] observed

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

ISSN 1433-8092

that the monotone arithmetic circuit complexity of lower envelopes of polynomials is not larger than that of polynomials themselves. This observation fairly easily implies that read-1 $(\vee, \wedge)$ circuits computing homogeneous Boolean functions (those with all minterms of the same length) are not stronger than monotone arithmetic circuits (Theorem 2). Thus, all known exponential lower bounds on the monotone arithmetic $(+, \times)$ circuit complexity of homogeneous multilinear polynomials give the same lower bounds for read-1 circuits.

Finally, we show that so-called multilinear (not necessarily monotone) DeMorgan $(\vee, \wedge, \neg)$ circuits computing *monotone* Boolean functions are not stronger than monotone read-1 circuits as well (Theorem 3). A DeMorgan $(\vee, \wedge, \neg)$ circuit is (semantically) *multilinear* if the Boolean functions g and h computed at the inputs to any AND gate depend on disjoint sets of variables. Note that this does not exclude that 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.

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. A *circuit* F over a semiring[1] $(R, \oplus, \odot)$ 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 $c \in R$ (a circuit is *constant-free* if it has no semiring elements $c \in R$ as inputs). 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.

Convention: To avoid considering "pathological" situations, under a semiring $(R, \oplus, \odot)$ we will understand only one of the following three semirings: 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 circuits over the following three semirings[2]:

- ○ $x \oplus y := x + y$ 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).

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 ith 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.

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

[1] 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.

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

$F = x \odot y \oplus z$ 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.

3. Monotone read-k circuits. A set of vectors $A \subseteq \mathbb{N}^n$ is an *antichain* if $a \in A$ and $b \leqslant a$ implies $a = b$. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if $b \leqslant a$ and $f(b) = 1$ imply $f(a) = 1$. A *lower one* of a monotone Boolean function f is a vector $a \in f^{-1}(1)$ such that $f(b) = 0$ for all vectors $b \leqslant a$, $b \neq a$. Let $A_f \subseteq f^{-1}(1)$ denote the set of all lower ones[3] of f. Note that the setA_f is an *antichain*, and it uniquely determines the entire function f. Namely,

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

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.

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).

We start with the following almost obvious 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 lower 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 the set B has the following two properties:*

(i) $B \subseteq A_f^{\uparrow}$;

[3] Note that each vector $a \in A_f$ is a characteristic 0-1 vector of the set of variables in some prime implicant of f. Thus, one can think of the set A_f as the set of prime implicants of f (represented as vectors).

(ii) *every lower 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, (i) yields $F(x) \leqslant f(x)$, while (ii) 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 the property (ii), 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. Item (i) of Claim 1 means that every monomial of the polynomial P produced by the arithmetic $(+, \times)$ version of the Boolean $(\vee, \wedge)$ circuit F must contain all variables of at least one prime implicant p of f, while (ii) means that for every prime implicant p of f there must be a monomial in P with the same set of variables as p. For example, the Boolean circuit $F(x,y,z) = (x \vee y)(x \vee z) \vee xy$ computes the Boolean function $f = x \vee yz$, whose set of lower ones is $A_f = \{(1,0,0), (0,1,1)\}$. 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, the shadows $b \in B$ of vectors $a \in A_f$ in property (ii) given by Claim 1 may have arbitrary large entries: we only have 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) $B \subseteq A_f^{\uparrow}$;

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

That is, we now require that every lower one $a \in A_f$ must have at *least one* its shadow in B with no entry larger than k; there are no restriction on the magnitude of the entries of other vectors of B. 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. There 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 $\langle b,x\rangle=b_1x_1+\cdots+b_nx_n$ 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 polynomial is also constant-free in that $c_b=0$ 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 [12, Lemma 3.2], when computing tropical constant-free polynomials, we can safely restrict us to constant-free circuits: the minimal circuit size will not increase. 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 (Eliminating constants, [12]). *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, and 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. ∎

The following lemma shows 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 $A \subseteq \{0,1\}^n$ be an antichain, and $f(x) = \bigvee_{a\in A} \bigwedge_{i\in\sup(A)} x_i$ be the Boolean function defined by A. Then*

$$\mathsf{Min}(A) = \mathsf{B}_1(f)\,.$$

Proof. To show the inequality $\mathsf{Min}(A) \leqslant \mathsf{B}_1(f)$, take a monotone read-once $(\vee, \wedge)$ 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-k circuit, we also have: $\forall a \in A\ \exists b \in B$ such that $\sup(b) = \sup(a)$ and $b_i \leqslant 1$ for all $i \in \sup(b)$. Since $a \in A$ are 0-1 vectors, this latter property means that $A \subseteq B$. Thus, the set B satisfies the inclusions $A \subseteq B \subseteq A^{\uparrow}$. The tropical $(\min, +)$ version F' of F (obtained by replacing OR gates by min gates, and AND gates by 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 [9, Theorem 2.4] observed that, by appropriately discarding some addition gates, every monotone arithmetic circuit computing a polynomial can be easily transformed into a monotone arithmetic circuit computing its lower envelope.

Claim 2 (Jerrum and Snir [9]). *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 lower ones of f is homogeneous. Note that then $\lfloor A_f\rfloor = A_f$.

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

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

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

Proof. Let $A := A_f \subseteq f^{-1}(1)$ be the set of lower ones of f, and 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 Claim 2 yields $\mathsf{Arith}(\lfloor A\rfloor) = \mathsf{Arith}(\lfloor B\rfloor) \leqslant \mathsf{Arith}(B) \leqslant s$. □

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 [23], Valiant [24], Jerrum and Snir [9], and Gashkov [5]. Together with Theorems 1 and 2, these bounds are also lower bounds on the size of monotone Boolean read-once $(\vee, \wedge)$ 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 is *cover-free* if $a+b \geqslant c$ with $a,b,c \in A$ implies $c \in \{a,b\}$. Schnorr [23] 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, this function is homogeneous. The set $A = A_f$ of lower 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 [8], Valiant [24], and Jerrum and Snir [9]: 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 lower 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 [6] 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 [5]. Using a fairly elementary "bottlenecks counting" argument, a lower bound $\mathsf{Arith}(A) \geqslant |A|/2lk^2$ was proved in [11] 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., [15]) that $|A| = (q^t - 1)/(q - 1)$. Kollár, Rónyai and Szabó [13, Theorem 3.3] proved that, for every t distinct elements $a_1, \ldots, a_t$ of $\mathbb{F}$, the system of equations $\mathrm{N}(a_1 + x) = 1$, $\mathrm{N}(a_2 + x) = 1, \ldots,$ $\mathrm{N}(a_t + x) = 1$ 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[4] $\mathsf{Arith}(A_{n,t}) \geqslant 2^{n-2\sqrt{n}\log n}$.

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 $\mathsf{Arith}(A) \geqslant 2^{n/2 - o(n)}$. □

3.4. Read-2 circuits can be exponentially stronger than read-1.

Let us 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 3. *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 lower 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

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

and Snir [9] 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

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

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

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

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. □

4. Multilinear DeMorgan circuits. We now turn to general, not necessarily monotone Boolean $(\vee, \wedge, \neg)$ circuits. For readers convenience, let us first recall some standard concepts regarding (not necessarily monotone) Boolean functions and circuits.

A *literal* is either a variable $x_i = x_i^1$ or its negation $\overline{x}_i = x_i^0$. A *term* is an AND of literals. A term t is a *zero term* if it contains a variable x_i together with its negation $\overline{x}_i$; otherwise, t is a *nonzero* term. or a term t, $\mathrm{var}(t)$ denotes the set of variables x_i such that x_i or $\overline{x}_i$ appears in t. For two Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$, the notation $f \leqslant g$ means that $f(a) \leqslant g(a)$ holds for all $a \in \{0,1\}^n$.

An *implicant* of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is a nonzero term $t \neq 0$ such that $t \leqslant f$ holds, that is, for every $a \in \{0,1\}^n$, $t(a) = 1$ implies $f(a) = 1$. In other words, a nonzero term t is an implicant of f if every evaluation of the literals of t to 1 already forces the function f to take value 1, regardless of the 0/1 values given to the remaining variables. An implicant t of f is a *prime implicant* of f if no proper subterm t' of t has this property, that is, if $t \leqslant t' \leqslant f$, then $t' = t$. For example, if $f(x,y,z) = xy \vee x\overline{y}z$, then xy, $x\overline{y}z$ and xz are implicants of f, but $x\overline{y}z$ is not a prime implicant.

The *i*th *neighbor* of a vector $a \in \{0,1\}^n$ if the vector $b \in \{0,1\}^n$ differing from a in only the ith position. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ *depends* on the ith variable x_i if $f(b) \neq f(a)$ holds for the ith neighbor b of some vector $a \in \{0,1\}^n$.

Claim 3. *A Boolean function $f(x_1, \ldots, x_n)$ depends on a variable x_i iff some prime implicant of f contains x_i or $\overline{x}_i$.*

Proof. The ⇐ direction follows directly from the definition of *prime* implicants. For the ⇒ direction, assume that $f(a) = 1$ for some vector a but $f(b) = 0$ for its ith neighbor b, and take a prime implicant p such that $p(a) = 1$. If p contained neither x_i nor $\overline{x}_i$, then we would have $p(b) = 1$ and, hence, $f(b) = 1$. □

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 to any AND gate have no input literals of the same variable x_i. Krieger [14] that syntactically multilinear $(\vee, \wedge, \neg)$ circuits generalize non-deterministic 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, any such circuit F computing a Boolean function f must have at least $\mathsf{Arith}(\lfloor A_f \rfloor)$ gates.

Ponnuswami and Venkateswaran [18] relaxed the "syntactic" multilinearity to "semantic" multilinearity as follows.

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

Note that this does not exclude that 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 show that multilinear DeMorgan $(\vee, \wedge, \neg)$ circuits are not stronger than monotone read-1 $(\vee, \wedge)$ circuits and, hence, also not stronger than monotone arithmetic $(+, \times)$ circuits. And as we will see, this happens because the former circuits are actually syntactically multilinear "with respect to prime implants," as directly follows from the following direct consequence of Claim 3.

Claim 4. *If g and h are the functions computed at the inputs of some AND gate of a multilinear DeMorgan $(\vee, \wedge, \neg)$ circuit, then $\mathrm{var}(p) \cap \mathrm{var}(q) = \emptyset$ holds for every prime implicant p of g and any prime implicant q of h.*

4.1. Monotone multilinear circuits. Recall that a *monotone* Boolean circuit is a $(\vee, \wedge)$ circuit, that is a DeMorgan $(\vee, \wedge, \neg)$ circuit, where negated input literals $\overline{x}_1, \ldots, \overline{x}_n$ are not used as inputs.

Lemma 4. *Every monotone multilinear Boolean circuit F is a read-once circuit.*

Proof. Easy induction on the size of the circuit F. Suppose that F is a multilinear circuit. Let G and H be the subcircuits of F whose output gates enter the output gate of F, and let g and h be the monotone Boolean functions computed by these subcircuits, and let $B = B_F$, $U = B_G$ and $V = B_H$ be the sets of exponent vectors produced by these three circuits. Suppose that the lemma holds for the circuits G and H. Since the entire circuit F is multilinear, both subcircuits G and H are multilinear. So, by the induction hypothesis, both G and H are read-once circuits. To show that then the entire circuit F is a read-once circuit, take an arbitrary lower one $a \in A_f$ of f. We have to show that $a \in B$. If $F = G \vee H$, then $B = U \cup V$ and $a \in A_g$ (or $a \in A_h$). By the induction hypothesis, we have that $a \in U \subseteq B$ (or $a \in V \subseteq B$), and we are done.

Now let $F = G \wedge H$. Then $B = U + V$ (a Minkowski sum), and $a = x \vee y$ (a componentwise OR of x y) for some vectors $x \in A_g$ and $y \in A_h$. Since the circuit F is *multilinear*, Claim 4 implies that the prime implicants of the (monotone) Boolean functions computed at the inputs to any AND gate are *disjoint*. Thus, $\sup(x) \cap \sup(y) = \emptyset$, that is, $a = x + y$. By the induction hypothesis, we have that $x \in U$ and $y \in V$. So, $a = x + y$ belongs to $B = U + V$, as desired. ∎

Remark 4.1. Note that the converse of Lemma 4 *does not* hold. For example, the circuit $F = xy \vee (x \vee z)(y \vee z)$ computing $f = xy \vee xz \vee yz$ is a read-once circuit: the polynomial $P = 2xy + xz + yz$ produced by the arithmetic version $xy + (x+z)(y+z)$ of F contains all three prime implicants of f (one with coefficient 2, and two with coefficients 1). But the circuit F is not multilinear because the functions $g = x \lor z$ and $h = y \lor z$ depend on the same variable z. □

For a monotone Boolean function f, let

$$\mathsf{B}^+_{\text{lin}}(f) := \text{min size of a monotone multilinear } (\lor, \land) \text{ circuit computing } f.$$

Corollary 1. *For every monotone Boolean function f, we have*

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

In particular, if the function f is homogeneous, then $\mathsf{B}^+_{\text{lin}}(f) \geqslant \mathsf{Arith}(A_f)$.

Proof. The first inequality $\mathsf{B}^+_{\text{lin}}(f) \geqslant \mathsf{B}_1(f)$ follows directly from Lemma 4, while the inequality $\mathsf{B}_1(f) \geqslant \mathsf{Arith}(\lfloor A_f \rfloor)$ is given by Theorem 2. □

Remark 4.2. Using different arguments, for homogeneous functions f, Lingas [16] proved a slightly worse lower bound $\mathsf{B}^+_{\text{lin}}(f) \geqslant \mathsf{Arith}(A_f)/\mathcal{O}(k^2)$, where k is the number of 1s in each vector of A_f.

4.2. Non-monotone multilinear circuits. Our goal now is to show that for any (not necessarily monotone) Boolean function, $\mathsf{Arith}(\lfloor A_f \rfloor)$ is also a lower bound on the size of *any* multilinear DeMorgan $(\lor, \land, \neg)$ circuit computing f (Theorem 3 below). In particular, we will show that if f is monotone, then minimal multilinear DeMorgan circuits for f must, in fact, be monotone.

Every DeMorgan circuit F not only computes a particular Boolean function but also *produces* (purely syntactically) a unique set $T(F)$ of terms in a natural way:

- ∘ if $F = z$ is an input literal, then $T(F) = \{z\}$;
- ∘ if $F = F_1 \lor F_2$, then $T(F) = T(F_1) \cup T(F_2)$;
- ∘ if $F = F_1 \land F_2$, then $T(F) = \{t_1 t_2 \colon t_i \in T(F_i), i = 1, 2\}$.

During the production of terms, we use the "shortening" axiom $x \land x = x$, but do not use the "annihilation" axiom $x \land \overline{x} = 0$. So, $T(F)$ can contain zero terms, that is, terms with a variable x_i and its negation $\overline{x}_i$. Easy induction on the circuit size shows that the Boolean function computed by a circuit F is the function computed as the OR of all produced terms. Thus, every nonzero term $t \in T(F)$ is an implicant of the Boolean function computed by F.

A *nonzero version* of a zero term t is a nonzero term t' obtained from t by removing exactly one literal from each pair $x_i, \overline{x}_i$ of contradicting literals appearing in t (it t is a nonzero term, then we let $t' = t$). For example, nonzero versions of $t = x_1\overline{x}_1 x_2 \overline{x}_3$ are $t' = x_1 x_2 \overline{x}_3$ and $t' = \overline{x}_1 x_2 \overline{x}_3$. Note that, in general, a circuit F may produce zero terms whose nonzero versions are *not* implicants of (the function computed by) F. For example, the circuit $F = (x \lor z)(y \lor \overline{z})$ produces a zero term $t = z\overline{z}$ neither of whose two nonzero extension $t' = z$ and $t' = \overline{z}$ is an implicant of the function computed by F.

The following lemma shows that multilinear circuits make an exception.

Lemma 5 (Zero terms lemma). *Let F be a DeMorgan circuit computing a Boolean function f. If F is multilinear, then nonzero versions of zero terms produced by F are implicants of f.*

Proof. Let F_1 and F_2 be the subcircuits of F rooted in the two gates entering the last gate of the circuit F. We argue by induction on the number s of gates in F. The basis case $s = 1$ is trivial, because then $F_1 = x_i^\alpha$ and $F_2 = x_j^\beta$ are input literals. If

$F = F_1 \wedge F_2 = x_i^\alpha x_j^\beta$ then, due to multilinearity of F, we have $i \neq j$, implying that $x_i^\alpha x_j^\beta$ is not a zero term. If $F = F_1 \vee F_2 = x_i^\alpha \vee x_j^\beta$, then we have no produced zero terms either.

Now suppose that the lemma holds for all DeMorgan circuits of size at most $s-1$, and let F be a 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, i.e. $t_1' \leqslant F_1$ and $t_2' \leqslant F_2$ hold for any nonzero version t_1' of any zero term $t_1 \in T(F_1)$ and for any nonzero version t_2' of any zero term $t_2 \in T(F_2)$. Take a zero term $t \in T(F)$ (if there is any), and let t' be any nonzero version of t. Our goal is to show that $t' \leqslant F$ holds.

The case when $F = F_1 \vee F_2$ is obvious: in this case, we have $t \in T(F_1)$ or $t \in T(F_2)$, and $t' \leqslant F$ follows by the induction hypothesis.

So let $F = F_1 \wedge F_2$. In this case, our zero term t is of the form $t = t_1 t_2$ for some (not necessarily zero) terms $t_1 \in T(F_1)$ and $t_2 \in T(F_2)$. Take any nonzero version t' of $t = t_1 t_2$. That is, t' is obtained by removing (replacing by constant 1) exactly one literal $z \in \{x_i, \overline{x}_i\}$ from each contradicting factor $x_i \overline{x}_i$ of t. Let Z be the set of all removed literals when forming t'. Call a literal $z \in Z$ *crossing* if z belongs to t_1, $\overline{z}$ belongs to t_2 and $z\overline{z}$ is a factor of neither of the terms t_1 and t_2. Let $Z' \subseteq Z$ consist of all crossing literals.

By the induction hypothesis, for the nonzero versions t_1' and t_2' of terms t_1 and t_2 obtained by removing all literals $z \in Z \setminus Z'$ from them, we have $t_1' \leqslant F_1$ and $t_2' \leqslant F_2$. Hence, there is a *prime* implicant p_1 of (the function computed by) F_1 such that $t_1' \leqslant p_1 \leqslant F_1$, and a *prime* implicant p_2 of F_2 such that $t_2' \leqslant p_2 \leqslant F_2$. Since the circuit F is multilinear, Claim 4 implies that $\mathrm{var}(p_1) \cap \mathrm{var}(p_2) = \emptyset$ and, hence, neither p_1 nor p_2 can contain any crossing literal $z \in Z'$. Our nonzero version t' of the zero term $t = t_1 t_2$ is of the form $t' = t_1'' t_2''$, where t_1'' and t_2'' are the terms resulting after crossing literals $z \in Z'$ are further removed from t_1' and t_2'. Since neither p_1 nor p_2 can contain any crossing literal, we still have $t_1'' \leqslant p_1$ and $t_1'' \leqslant p_1$. This yields $t' = t_1'' t_2'' \leqslant p_1 p_2 \leqslant F_1 \wedge F_2 = F$, as desired. □

4.3. From non-monotone to monotone circuits. The *positive factor* t_+ of a term t is obtained by replacing every its negated literal with constant 1. That positive factors of implicants of *monotone* Boolean functions are also their implicants is an almost obvious fact.

Claim 5. *Let f be a monotone Boolean function, and t be a nonzero term. If $t \leqslant f$ then also $t_+ \leqslant f$.*

Proof. Let $t \leqslant f$ hold (i.e., t is an implicant of f). Since the function f is monotone, $f(a) = 0$ implies $f(b) = 0$ for all $b \leqslant a$. That is, if f reject a vector a, then it reject any vector obtained from a by flipping some its 1s to 0s. Thus, if $t \leqslant f$, then $t(a) = 0$ must imply $t(b) = 0$ for all $b \leqslant a$, meaning that $t_+(a) = 0$ must then hold as well. □

We can view every DeMorgan circuit $F(x)$ computing a Boolean function $f(x)$ of n variables as a monotone 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.

The *upward closure* of a not necessarily monotone Boolean function $f(x)$ is the monotone Boolean function

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

A *lower one* of a (not necessarily monotone) Boolean function f is a vector a such that $f(a)=1$ but $f(b)=0$ for all $b\leqslant a$, $b\neq a$. Note that, if $A_f\subseteq f^{-1}(1)$ denotes the set of all lower ones of f, then $f^{\triangledown}(x)=1$ iff $x\geqslant a$ for some $a\in A_f$. For example, the set of lower 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.

Lemma 6. *Let F be a DeMorgan circuit computing a Boolean function f.*

(i) *The circuit F_+ computes $f^{\triangledown}$ if and only if the positive factor of every zero term produced by F is an implicant of $f^{\triangledown}$.*

(ii) *If the circuit F is multilinear, then F_+ computes $f^{\triangledown}$.*

Proof. (i) Terms produced by the monotone version F_+ of the circuit F are positive factors t_+ of terms $t\in T(F)$ produced by the circuit F: if $t=t(x,\overline{x})$ then $t(x,\vec{1})=t_+$. On the other hand, for every term t we have $t^{\triangledown}=0$ if t is a zero term, and $t^{\triangledown}=t_+$ if t is a non-zero term. Since $(g\vee h)^{\triangledown}=g^{\triangledown}\vee h^{\triangledown}$ holds for any Boolean functions $g,h:\{0,1\}^n\to\{0,1\}$, we obtain

$$f^{\triangledown}=\bigvee_{t\in T}t^{\triangledown}=\bigvee_{t\in T'}t_+\leqslant\bigvee_{t\in T}t_+=F_+\,, \tag{4.1}$$

where $T=T(F)$ is the set of all terms produced by the circuit F, $T'\subseteq T$ is the set of all non-zero terms of T. By (4.1), the equality $f^{\triangledown}=F_+$ holds iff $t_+\leqslant f^{\triangledown}$ holds for every term $t\in T\setminus T'$, that is, iff the positive factor t_+ of every *zero* term $t\in T(F)$ is an implicant of $f^{\triangledown}$.

(ii) Suppose that the circuit F is multilinear. It is enough to show that then $t_+\leqslant f^{\triangledown}$ holds in (4.1) for every zero term $t\in T(F)$ (if there is any). To show this, take an arbitrary zero term $t\in T(F)$. Since the circuit F is multilinear, the zero terms lemma (Lemma 5) implies that $t'\leqslant f$ holds for every nonzero version t' of t. In particular, $t'\leqslant f$ holds for the nonzero version t' obtained by removing the negated literal $\overline{x}_i$ from each contradicting factor $x_i\overline{x}_i$ of t (we replace each such literal by 1). In the positive factor t'_+ of the term t', we replace by 1s the remaining negated literals of t'; hence, $t'_+=t_+$. Since $t'\leqslant f$, we also have $t'\leqslant f^{\triangledown}$. Since the Boolean function $f^{\triangledown}$ is monotone, Claim 5 yields $t'_+\leqslant f^{\triangledown}$ and, hence, also the desired inequality $t_+\leqslant f^{\triangledown}$.□

Remark 4.3. The converse of Lemma 6(ii) does not hold: the monotone version F_+ of a circuit F may compute $f^{\triangledown}$ 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 $f^{\triangledown}=x\vee y$. The monotone version $F_+=y\cdot 1\vee x(1\vee 1\cdot y)=x\vee y$ of the circuit F computes $f^{\triangledown}$. 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

$$\begin{aligned}\mathsf{B}(f) &:= \text{min size of a DeMorgan } (\vee,\wedge,\neg) \text{ circuit computing } f;\\ \mathsf{B}_{\rm lin}(f) &:= \text{min size of a multilinear DeMorgan } (\vee,\wedge,\neg) \text{ circuit computing } f.\end{aligned}$$

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

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

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

Proof. Let F be a multilinear DeMorgan $(\vee, \wedge, \neg)$ circuit of size $s = \mathsf{B}_{\text{lin}}(f)$ computing f. Since the circuit F is multilinear, Lemma 6 implies that its monotone (also multilinear) version F_+ computes the (monotone) upward closure $g := f^{\triangledown}$ of f. The (monotone) circuit F_+ has size at most s and, by Lemma 4, is a read-once circuit. This shows the inequality $\mathsf{B}_{\text{lin}}(f) \geqslant \mathsf{B}_1(g)$. The inequality $\mathsf{B}_1(g) \geqslant \mathsf{Arith}(\lfloor A_g \rfloor)$ is given by Theorem 2. Since lower ones of a Boolean function f and of its upward closure $g = f^{\triangledown}$ are the same, we have $\lfloor A_g \rfloor = \lfloor A_f \rfloor$ and, hence, also $\mathsf{B}_{\text{lin}}(f) \geqslant \mathsf{B}_1(g) \geqslant \mathsf{Arith}(\lfloor A_g \rfloor) = \mathsf{Arith}(\lfloor A_f \rfloor)$. If the function f is monotone, then $g = f$. □

To demonstrate the weakness of (even non-monotone) multilinear $(\vee, \wedge, \neg)$ circuits, consider the *permutation matrix* function Per_n. This is a non-monotone Boolean function whose inputs are Boolean $n \times n$ matrices $x = (x_{i,j})$, and

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

That is, $\mathrm{Per}_n(x) = 1$ iff x is a permutation matrix.

Lemma 7. *For $f = \mathrm{Per}_n$, we have $\mathsf{B}_{\text{lin}}(f) = 2^{\Omega(n)}$ but $\mathsf{B}(f) = \mathcal{O}(n^3)$.*

Proof. Note that the upward closure $g := f^{\triangledown}$ of the function $f = \mathrm{Per}_n$ is the perfect matching function $g = \mathrm{Match}_n$ which accepts a 0-1 matrix $x = (x_{i,j})$ iff x contains at least one permutation matrix as a submatrix. We already know that $\mathsf{Arith}(A_g) = 2^{\Omega(n)}$ (Example 3.3). By Theorem 2, we have $\mathsf{B}_1(g) \geqslant \mathsf{Arith}(A_g)$, and Theorem 3 gives a lower bound $\mathsf{B}_{\text{lin}}(f) \geqslant \mathsf{B}_1(g) \geqslant \mathsf{Arith}(A_g) = 2^{\Omega(n)}$.

To show the upper bound $\mathsf{B}(f) = \mathcal{O}(n^3)$, consider the circuit $F = F_1 \wedge F_2$, where

$$F_1 = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{n} x_{i,j} \wedge \bigwedge_{k \neq j} \overline{x}_{i,k} \right) \text{ and } F_2 = \bigwedge_{j=1}^{n} \left(\bigvee_{i=1}^{n} x_{i,j} \wedge \bigwedge_{l \neq i} \overline{x}_{l,j} \right).$$

Note that $F_1(x) = 1$ iff every row of x has exactly one 1, and $F_2(x) = 1$ iff every column of x has exactly one 1, meaning that the circuit F computes $f = \mathrm{Per}_n$. The circuit F is actually a depth-3 formula with $\mathcal{O}(n^2)$ unbounded fanin gates. □

Acknowledgment. I am thankful to Andrzej Lingas for turning my attention to multilinear circuits.

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] R. W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 5:345, 1962.

[4] L. R. Ford. Network flow theory. Technical Report P-923, 1956.

[5] 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.

[6] 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.

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

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

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

[10] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012.

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

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

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

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

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

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

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

[18] 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.

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

[20] 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.

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

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

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

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

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