Separating OR, SUM, and XOR Circuits[☆]

Magnus Find[a,1], Mika Göös[b,c], Matti Järvisalo[c], Petteri Kaski[d], Mikko Koivisto[c], Janne H. Korhonen[c]

[a] *National Institute of Standards and Technology, USA*
[b] *Department of Computer Science, University of Toronto, Canada*
[c] *HIIT & Department of Computer Science, University of Helsinki, Finland*
[d] *HIIT & Department of Information and Computer Science, Aalto University, Finland*

Abstract

Given a boolean $n \times n$ matrix A we consider arithmetic circuits for computing the transformation $x \mapsto Ax$ over different semirings. Namely, we study three circuit models: monotone OR-circuits, monotone SUM-circuits (addition of non-negative integers), and non-monotone XOR-circuits (addition modulo 2). Our focus is on *separating* OR-circuits from the two other models in terms of circuit complexity:

(1) We show how to obtain matrices that admit OR-circuits of size $O(n)$, but require SUM-circuits of size $\Omega(n^{3/2}/\log^2 n)$.
(2) We consider the task of *rewriting* a given OR-circuit as a XOR-circuit and prove that any subquadratic-time algorithm for this task violates the strong exponential time hypothesis.

Keywords:
arithmetic circuits, boolean arithmetic, idempotent arithmetic, monotone separations, rewriting

---

[☆] A preliminary version of our second contribution has appeared in SAT 2012 [15].
[1] This work was partially conducted while M.F. was at University of Southern Denmark and visiting University of Toronto.

1. Introduction

Arithmetic circuit models. A basic question in arithmetic complexity is to determine the minimum size of an arithmetic circuit that evaluates a linear map $x \mapsto Ax$; see, e.g., the recent survey of Jukna and Sergeev [17]. In this work we approach this question from the perspective of relative complexity by varying the circuit model while keeping the matrix A fixed, with the goal of separating different circuit models. That is, our goal is to show the existence of matrices A that admit small circuits in one model but have only large circuits in a different model.

We will focus on boolean arithmetic and the following three circuit models. Our circuits consist of either

1. only $\vee$-gates (i.e., boolean sums; rectifier circuits),
2. only $+$-gates (i.e., integer addition; cancellation-free circuits), or
3. only $\oplus$-gates (i.e., integer addition mod 2).

These three types of circuits have been studied extensively in their own right (see Section 3), but fairly little is known about their relative powers.

Each model admits a natural description both from an algebraic and a combinatorial perspective.

Algebraic perspective. In the three models under consideration, each circuit with inputs $x_1, \ldots, x_n$ and outputs $y_1, \ldots, y_m$ computes a vector of *linear forms*

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1, \ldots, m.$$

That is, $y = Ax$, where $A = (a_{ij})$ is an m by n boolean matrix with $a_{ij} \in \{0, 1\}$ and the arithmetic is either

1. in the boolean semiring $(\{0, 1\}, \vee, \wedge)$,
2. in the semiring of non-negative integers $(\mathbb{N}, +, \cdot)$, or
3. in GF(2).

As an example, Figure 1 displays two circuits for computing $y = Ax$ for the same A using two different operators; the circuit on the right requires one more gate.

Combinatorial perspective. A circuit computing $y = Ax$ for a boolean matrix A can also be viewed combinatorially: every gate g is associated with a subset of the formal variables $\{x_1, \ldots, x_n\}$; this set is called the *support* of g and it is denoted $\mathrm{supp}(g)$. The input gates correspond to the singletons $\{x_j\}$, $j = 1, \ldots, n$, and every non-input gate computes either

1. the set union ($\vee$),
2. the disjoint set union ($+$), or
3. the symmetric difference ($\oplus$) of its children.

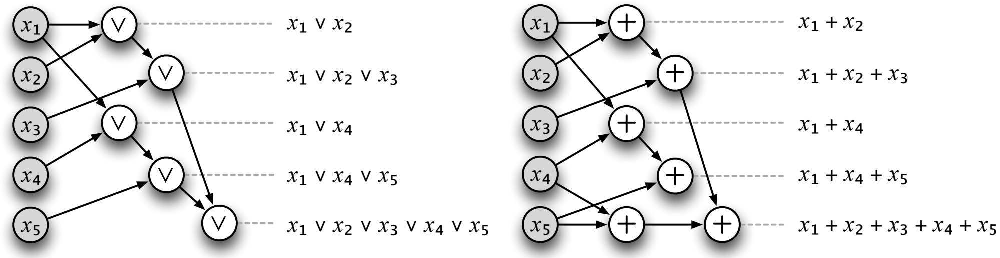

Figure 1: An $\vee$-circuit (left) and a $+$-circuit (right).

This way an output gate y_i will have $\mathrm{supp}(y_i) = \{x_j : a_{ij} = 1\}$.

Note the special structure of a $+$-circuit: there is at most one directed path from any input x_j to any output y_i. In fact, from this perspective, every $+$-circuit for A is easy to interpret both as an $\vee$-circuit for A, and as a $\oplus$-circuit for A (equivalently, there are onto homomorphisms from $(\mathbb{N}, +, \cdot)$ to $(\{0,1\}, \vee, \wedge)$ and $\mathrm{GF}(2)$). In this sense, both $\vee$- and $\oplus$-circuits are at least as efficient as $+$-circuits.

Relative complexity. More generally we fix a boolean matrix A and ask how the circuit complexity of computing $y = Ax$ depends on the underlying arithmetic. To make this quantitative, denote by $C_\vee(A)$, $C_+(A)$, and $C_\oplus(A)$ the minimum number of wires in an unbounded fan-in circuit for computing $y = Ax$ in the respective models. For simplicity, we restrict our attention to the case of square matrices so that $m = n$.

For $\mathsf{X}, \mathsf{Y} \in \{\vee, +, \oplus\}$, we are interested in the complexity ratios

$$\mathrm{gap}_{\mathsf{X}/\mathsf{Y}}(n) \;:=\; \max_{A \in \{0,1\}^{n \times n}} C_{\mathsf{X}}(A)/C_{\mathsf{Y}}(A).$$

For example, we have that $\mathrm{gap}_{\vee/+}(n) = \mathrm{gap}_{\oplus/+}(n) = 1$ and that $\mathrm{gap}_{+/\oplus}(n) \geq \mathrm{gap}_{\vee/\oplus}(n)$ for all n, by the above fact that each $+$-circuit can be interpreted as an $\vee$-circuit and as a $\oplus$-circuit.

We review the motivation for studying separation bounds in Section 3. We next state our results in Section 2; these are summarised in Figure 2 along with the prior separation results.

Notation. A *circuit* $\mathcal{C}$ is a directed acyclic graph where the vertices of in-degree (or *fan-in*) zero are called *input gates* and all other vertices are called *arithmetic gates*. One or more arithmetic gates are designated as *output gates*. The size $|\mathcal{C}|$ of the circuit is the number of edges (or *wires*) in the circuit. We abbreviate $[n] := \{1, \ldots, n\}$.

2. Results

+/∨-Separation. We begin by studying the monotone complexity of tensor product matrices of the form

$$A = B_1 \otimes B_2,$$

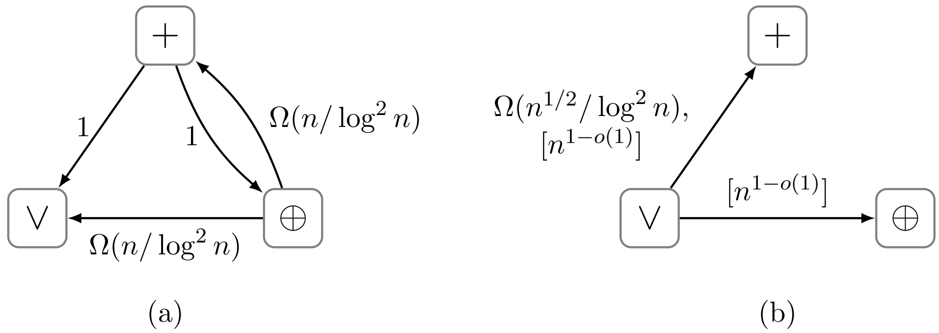

Figure 2: Separation bounds from (a) prior work, and (b) present work. An arrow from Y to X is labelled with $\mathrm{gap}_{\mathsf{X}/\mathsf{Y}}(n)$; bounds for (X,Y)-Rewrite are given inside square brackets.

where $\otimes$ denotes the usual Kronecker product of matrices. In Section 4, we prove a direct sum type theorem on their monotone complexity. As a corollary, we obtain matrices that are easy for $\vee$-circuits, $C_\vee(A) = O(n)$, but hard for $+$-circuits, $C_+(A) = \Omega(n^{3/2}/\log^2 n)$.

Theorem 1. $\mathrm{gap}_{+/\vee}(n) = \Omega(n^{1/2}/\log^2 n)$.

We are not aware of any prior lower bound techniques that work against $+$-circuits, but not against $\vee$-circuits. Hence, as far as we know, Theorem 1 is a first step in this direction. We also mention that, after a preprint of this work appeared, Jukna and Sergeev [17] have given an alternative proof and extensions of our Theorem 1.

While we are unable to enlarge the gap in Theorem 1, or prove any super-constant lower bounds on $\mathrm{gap}_{\oplus/\vee}$, we still conjecture that all the non-trivial complexity gaps between the three models are of order $n^{1-o(1)}$. Our second result provides some evidence towards these conjectures.

Circuit rewriting. In Section 5, we show that if certain $\vee$-circuits that are derived from CNF formulas could be efficiently *rewritten* as equivalent $+$- or $\oplus$-circuits, this would imply unexpected consequences for exponential-time algorithms. More precisely, we study the following problem.

The (X,Y)-Rewrite **problem.** On input an X-circuit $\mathcal{C}$, output a Y-circuit that computes the same matrix as $\mathcal{C}$.

Both $(\vee,+)$-Rewrite and $(\vee,\oplus)$-Rewrite admit simple algorithms that output a circuit of size $O(|\mathcal{C}|^2)$ in time $O(|\mathcal{C}|^2)$. However, we show that any significant improvement on these algorithms would give a randomised $2^{(1-\epsilon)n}\,\mathrm{poly}(n,m)$ time algorithm for deciding whether an n-variable m-clause CNF formula is satisfiable. This would directly violate *the strong exponential time hypothesis* [7, 13, 14], which states that no such randomised algorithm exists. (See [13] for a more careful definition of the hypothesis.)

Theorem 2. *Neither* $(\vee,+)$-Rewrite *nor* $(\vee,\oplus)$-Rewrite *can be solved in time* $O(|\mathcal{C}|^{2-\epsilon})$ *for any constant* $\epsilon > 0$*, unless the strong exponential time hypothesis fails.*

Theorem 2 provides evidence, e.g., for the conjecture $\mathrm{gap}_{\oplus/\vee}(n) = n^{1-o(1)}$ in the following sense. If there is a family of matrices A witnessing $C_\oplus(A)/C_\vee(A) = n^{1-o(1)}$, then clearly no $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm exists for $(\vee, \oplus)$-Rewrite: if we are given a minimum-size $\vee$-circuit for A as input, there is no time to write down a legal output.

Our proof of Theorem 2 shows, in particular, that an $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm for $(\vee, +)$-Rewrite would give an improved algorithm for counting the number of satisfying assignments to a given CNF formula (#CNF-SAT). Similarly, an $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm for $(\vee, \oplus)$-Rewrite would give an improved algorithm for deciding whether the number of satisfying assignments is odd ($\oplus$CNF-SAT).

3. Related work

Upper bounds. The trivial depth-1 circuit for a boolean matrix A uses $|A|$ wires, where we denote by $|A|$ the *weight* of A, i.e., the number of 1-entries in A. Even though $|A|$ might be of order $\Theta(n^2)$, Lupanov [22] (as presented by Jukna [16, Lemma 1.2]) constructs depth-2 circuits (applicable in all the three models) of size $O(n^2/\log n)$ for any A. This implies the universal upper bound

$$\mathrm{gap}_{\mathsf{X}/\mathsf{Y}}(n) = O(n/\log n). \tag{Lupanov}$$

Lower bounds. Standard counting arguments [16, §1.4] show that most $n \times n$ matrices have wire complexity $\Omega(n^2/\log n)$ in each of the three models. Combining this with Lupanov's upper bound we conclude that a random matrix does little to separate our models:

Fact 1. For a uniformly random A, the ratio $C_\mathsf{X}(A)/C_\mathsf{Y}(A)$ is a constant w.h.p.

Unsurprisingly, it can also be shown that finding a minimum-size circuit for a given matrix is NP-hard in all the models. For $\vee$- and $+$-circuits this follows from the NP-completeness of the Ensemble Computation problem as defined by Garey and Johnson [10, PO9]. For $\oplus$-circuits this was proved by Boyar et al. [5].

$\vee$-circuits. The study of $\vee$-circuits has been centered around finding *explicit* matrices that are hard for $\vee$-circuits. Here, dense *rectangle-free* matrices and their generalisations, *(s,t)-free* matrices, are a major source of lower bounds.

Definition. A matrix A is called *(s,t)-free* if it does not contain an $(s+1) \times (t+1)$ all-1 submatrix. Moreover, A is simply called *k-free* if it is (k,k)-free.

Nechiporuk [24] and independently Lamagna and Savage [21] we the first to apply a construction of dense 1-free matrices (e.g., incidence matrices of finite projective planes) to give a lower bound of $C_\vee(A) = \Omega(n^{3/2})$ for an explicit matrix A. Subsequently, Mehlhorn [23] and Pippenger [26] established the following theorem that gives a general template for this type of lower bound; we use it extensively later.

Theorem 3 (Mehlhorn–Pippenger)**.** *If A is (s,t)-free, then $C_\vee(A) \geq |A|/(st)$.*

Currently, the best lower bound for an explicit A is obtained by applying Theorem 3 to a matrix construction of Kollár et al. [19]; the lower bound is $C_\vee(A) \geq n^{2-o(1)}$ (see also Gashkov and Sergeev [11, §3.2]).

$\oplus$-circuits. It is a long-standing open problem to exhibit explicit matrices requiring super-linear size $\oplus$-circuits. No such lower bounds are known even for log-depth circuits, and the only successes so far are in the case of bounded depth [1, 9; 16, §13.5]. This, together with Fact 1, makes it particularly difficult to prove lower bounds on $\mathrm{gap}_{\oplus/\vee}$.

However, in the opposite direction, complexity gaps exist: Sergeev et al. [11, 12] obtained a bound $\mathrm{gap}_{\vee/\oplus}(n) = n^{1-o(1)}$. This was subsequently improved to $\mathrm{gap}_{\vee/\oplus}(n) = \Omega(n/\log^2 n)$ by Boyar and Find [4] with an alternative proof given by Jukna and Sergeev [17].

$+$-circuits. Additive circuits have been studied extensively in the context of the *addition chain* problem (see Knuth [18, §4.6.3] for a survey) and its generalisations [27].

Algebraic complexity. A particular motivation for studying the separation between $\vee$- and $+$-circuits is to understand the complexity of zeta transforms on partial orders [2]. Indeed, the characteristic matrix of every partial order $\leq$ has an $\vee$-circuit proportional to the number of covering pairs in $\leq$, but the existence of small $+$-circuits (and hence fast zeta transforms) is currently not understood satisfactorily.

Strong exponential time hypothesis. Theorem 2 is similar to other recent lower bound results for polynomial-time solvable problems based on the strong exponential time hypothesis [25]. See also Cygan et al. [8].

4. $+/\vee$-Separation

Overview. In this section we give a direct sum type theorem for the monotone complexity of tensor product matrices. Using this, we obtain a separation of the form

$$\begin{aligned} C_\vee(B \otimes A) &= O(N), \\ C_+(B \otimes A) &= \Omega(N^{3/2}/\log^2 N), \end{aligned} \tag{1}$$

where $\otimes$ denotes the usual Kronecker product of matrices and $N = n^2$ denotes the number of input and output variables. This will prove Theorem 1.

Tensor products. As a first example, let A be a fixed boolean $n \times n$ matrix and consider the matrix product

$$X \mapsto AX\,, \tag{2}$$

where we think of X as a matrix of $N = n \times n$ input variables. If we arrange these variables into a column vector x by stacking the columns of X on top of one another, then (2) becomes

$$x \mapsto (I \otimes A)x, \tag{3}$$

where I is the $n \times n$ identity matrix. That is, $I \otimes A$ is the block matrix having n copies of A on the diagonal.

The transformation (3) famously admits non-trivial $\oplus$-circuits due to the fact that fast matrix multiplication algorithms can be expressed as small bilinear circuits over GF(2). However, it is easy to see that in the case of our monotone models, no non-trivial speed-up is possible: any $\vee$-circuit for (3) must compute A independently n times, that is, we have

$$C_\vee(I \otimes A) = n \cdot C_\vee(A). \tag{4}$$

This follows from the observation that two subcircuits corresponding to two different columns of X cannot share gates due to monotonicity.

Our approach. We will generalise the above setting slightly and use tensor products of the form $B \otimes A$ to separate $\vee$- and $+$-circuits. Analogously to (2), one can check that the matrix $B \otimes A$ corresponds to computing the mapping

$$X \mapsto AXB^\top. \tag{5}$$

We aim to show that for suitable choices of A and B computing $B \otimes A$ is easy for $\vee$-circuits but hard for $+$-circuits. We will choose A to have large complexity (e.g., choose A at random), and think of B as dictating how many independent copies of A a circuit must compute.

More precisely, define $\mathrm{rk}_\vee(B)$ and $\mathrm{rk}_+(B)$ as the minimum r such that B can be written as $B = PQ^\top$ over the boolean semiring or over the semiring of non-negative integers, respectively, where P and Q are $n \times r$ matrices. Equivalently, $\mathrm{rk}_\vee(B)$ (resp., $\mathrm{rk}_+(B)$) is the minimum number of rectangles (resp., non-overlapping rectangles) that are required to cover all 1-entries of B.

These cover numbers appear often in the study of communication complexity [20]. In this context, the matrix $B = \bar{I}$—the boolean complement of the identity I—is the usual example demonstrating a large gap between the two concepts [20, Example 2.5]:

$$\begin{aligned} \mathrm{rk}_\vee(\bar{I}) &= \Theta(\log n), \\ \mathrm{rk}_+(\bar{I}) &= n. \end{aligned}$$

We will use this gap to show that, up to polylogarithmic factors,

$$\begin{aligned} C_\vee(\bar{I} \otimes A) &\approx \mathrm{rk}_\vee(\bar{I}) \cdot n^2, \\ C_+(\bar{I} \otimes A) &\approx \mathrm{rk}_+(\bar{I}) \cdot n^2. \end{aligned}$$

In terms of the number of input variables $N = n^2$, we will obtain (1).

Upper bound for $\vee$-circuits. Suppose $B = PQ^\top$ where P and Q are $n \times \mathrm{rk}_\vee(B)$ matrices. We can compute (5) as

$$(A(XQ))P^\top,$$

which requires 3 matrix multiplications, each involving $\mathrm{rk}_\vee(B)$ as one of the dimensions (the other dimensions being at most n).

If these 3 multiplications are naively implemented with an $\vee$-circuit of depth 3, each layer will contain at most $\mathrm{rk}_\vee(B)n^2$ wires so that $C_\vee(B \otimes A) \leq 3\,\mathrm{rk}_\vee(B)n^2$. However, one can still use Lupanov's techniques to save an additional logarithmic factor: if $\mathrm{rk}_\vee(B) = O(\log n)$, Corollary 1.35 in Jukna [16] can be applied to show that each of the three multiplications above can be computed using $O(n^2)$ wires. Thus, for $B = \bar{I}$ we get

Lemma 4. *$C_\vee(\bar{I} \otimes A) = O(n^2)$ for all A.* □

Lower bound for $+$-circuits. Intuitively, since low-rank decompositions are not available for $\bar{I}$ in the semiring of non-negative integers, a $+$-circuit for $\bar{I} \otimes A$ should be forced to compute $\mathrm{rk}_+(\bar{I}) = n$ independent copies of A. More generally, we ask the following.

Direct sum question. Do we have $C_+(B \otimes A) \geq \mathrm{rk}_+(B) \cdot C_+(A)$ for all A, B?

Alas, we can answer this affirmatively only in some special cases. For example, the trivial case $B = I$ was discussed above (4), and it is not hard to generalise the argument to show that the lower bound holds in case B admits a fooling set of size $\mathrm{rk}_+(B)$. (When B is viewed as an incidence matrix of a bipartite graph, a *fooling set* is a matching no two of whose edges induce a 4-cycle. See [20, §1.3].) However, since this will not be the case when $B = \bar{I}$, we will settle for the following version, which suffices for the separation result.

Theorem 5. *For all (s,t)-free A,*

$$C_+(B \otimes A) \geq \mathrm{rk}_+(B) \cdot \frac{|A|}{st}. \tag{6}$$

Note that if we set $B = I$ in Theorem 5 we recover essentially an analogue of Theorem 3 restricted to $+$-circuits.

For the purposes of the proof we switch to the combinatorial perspective: For A and B we introduce two sets of n formal variables X_A and X_B. Moreover, we let $A_1, \ldots, A_n \subseteq X_A$ and $B_1, \ldots, B_n \subseteq X_B$ denote the associated outputs. That is, each output A_i is defined by one row of A, and each output B_j is defined by one row of B. With this terminology, the input variables for $B \otimes A$ are the pairs in $X_A \times X_B$; we think of X_A as indexing the rows and X_B as indexing columns of the variable matrix $X_A \times X_B$. Finally, $B \otimes A$ corresponds to computing the n^2 outputs

$$A_i \times B_j, \quad \text{for } i, j \in [n].$$

In the following proof we use the (s,t)-freeness of A to "zoom in" on that layer of the circuit which reveals the large wire complexity (similarly to Mehlhorn [23]). We advise the reader to first consider the case $s = t = 1$, as this already contains the main idea of the proof.

Proof of Theorem 5. Let $\mathcal{C}$ be a $+$-circuit computing $B \otimes A$. As a first step, we simplify $\mathcal{C}$ by allowing input gates to have larger-than-singleton supports. Namely, let F consist of those gates of $\mathcal{C}$ whose supports are contained in a *t-wide row cylinder* of the form $Y \times X_B$ where $Y \subseteq X_A$ and $|Y| \leq t$. We simply declare that all computations done by gates in F come for free: we promote a gate in F to an input gate and delete all its incoming wires. We continue to denote the modified circuit by $\mathcal{C}$—clearly, these modifications only decrease its wire complexity.

Call a wire that is connected to an input gate an *input wire* and denote the set of input wires by W. The wire complexity lower bound (6) will follow already from counting the number $|W|$ of input wires.

For $i \in [n]$ denote by $\mathcal{C}_i$ the subcircuit of $\mathcal{C}$ computing the n outputs $A_i \times B_j$, $j \in [n]$, and denote by $W(i)$ the input wires of $\mathcal{C}_i$; we claim that

$$|W(i)| \geq \mathrm{rk}_+(B) \cdot \frac{|A_i|}{t}. \tag{7}$$

Before we prove (7), we note how it implies the theorem. Each input wire $w \in W$ is feeding into a non-input gate having their support not contained in a t-wide row cylinder. Due to (s,t)-freeness of A this means that w can appear only in at most s different $\mathcal{C}_i$. Thus, the sum $\sum_i |W(i)|$ counts w at most s times and, more generally, we have

$$|W| = \left|\bigcup_{i=1}^{n} W(i)\right| \geq \sum_{i=1}^{n} \frac{|W(i)|}{s},$$

which implies (6) given (7).

It now remains to prove (7). Fix $i \in [n]$. If A_i is empty the claim is trivial. Otherwise fix a variable $x \in A_i$ and consider the structure of $\mathcal{C}_i$ when restricted to the variables $\{x\} \times X_B$. Since this set of variables can be naturally identified with X_B by ignoring the first coordinate, we can view $\mathcal{C}_i$ as computing a copy of B on the variables $\{x\} \times X_B$.

Indeed, we define the *x-support* $\mathrm{supp}_x(w)$ of an input wire $w \in W(i)$ to be the set of $y \in X_B$ such that the variable (x, y) is contained in the support of w. (The support of w is simply the support of the adjacent input gate.) Moreover, we let

$$W_x(i) := \{w \in W(i) : \mathrm{supp}_x(w) \neq \varnothing\}.$$

Put otherwise, $W_x(i)$ consists of the input wires that are used by $\mathcal{C}_i$ in computing a copy of B on the variables $\{x\} \times X_B$. Associate to each $w \in W_x(i)$ a rectangle

$$R_x(w) := \text{co-supp}_x(w) \times \mathrm{supp}_x(w),$$

where $\text{co-supp}_x(w)$ is the set of $j \in [n]$ such that w appears in the subcircuit $\mathcal{C}_{ij}$ of $\mathcal{C}_i$ that computes the output $A_i \times B_j$. Now, the crucial observation is that the collection of rectangles $\{R_x(w) : w \in W_x(i)\}$ is a non-overlapping cover of B, because $\mathcal{C}_i$ computes a copy of B by taking disjoint unions of the supports $\{\mathrm{supp}_x(w) : w \in W_x(i)\}$. Therefore, we must have that

$$|W_x(i)| \geq \mathrm{rk}_+(B). \tag{8}$$

To finish the proof, we note that a single input wire $w \in W(i)$, being t-wide, can only be contained in the sets $W_x(i)$ for at most t different $x \in A_i$. Thus, the sum $\sum_x |W_x(i)|$ counts w at most t times and, more generally, we have

$$|W(i)| = \bigcup_{x \in A_i} W_x(i) \;\; \geq \sum_{x \in A_i} \frac{|W_x(i)|}{t},$$

which implies (7) given (8). $\square$

It is easy to check (and well-known in the context of random graphs [3, §11]) that a random matrix $A \in \{0,1\}^{n \times n}$ is $O(\log n)$-free w.h.p. Since a random matrix has weight $|A| = \Theta(n^2)$ w.h.p., we obtain from Theorem 5 the following corollary, which, together with Lemma 4, proves Theorem 1.

Corollary 6. *A random A satisfies $C_+(\bar{I} \otimes A) = \Omega(n^3/\log^2 n)$ w.h.p.* $\square$

5. Rewriting

Overview. In this section we study what would happen if $(\vee,+)$-Rewrite or $(\vee,\oplus)$-Rewrite could be solved in subquadratic time. Namely, we show that this eventuality would contradict the strong exponential time hypothesis. This will prove Theorem 2. As discussed in Section 2, we interpret this as evidence for our conjectures $\text{gap}_{+/\vee}(n) = n^{1-o(1)}$ and $\text{gap}_{\oplus/\vee}(n) = n^{1-o(1)}$.

Preliminaries. For purposes of computations, we assume that $|\mathcal{C}| \geq n$ for any n-input circuit $\mathcal{C}$ considered in this section. This is to make each $\mathcal{C}$ admit a binary representation of length $\tilde{O}(|\mathcal{C}|)$ where the $\tilde{O}$ notation hides factors polylogarithmic in n. For concreteness, $\mathcal{C}$ might be represented as two lists: (i) the list of gates in $\mathcal{C}$, with output gates indicated, and (ii) the list of wires in $\mathcal{C}$; both lists are given in topological order, with the input wires of each gate forming a consecutive sublist of the list of wires. Whatever the encoding, we assume it is efficient enough so that the following property holds.

Proposition 7. *On input an* X*-circuit $\mathcal{C}$ and a vector x, the output $\mathcal{C}(x)$ can be computed in time $\tilde{O}(|\mathcal{C}|)$ (in the usual RAM model of computation).* $\square$

The following proposition records a similar observation for circuit rewriting.

Proposition 8. *Both* $(\vee,+)$-Rewrite *and* $(\vee,\oplus)$-Rewrite *can be solved in time $\tilde{O}(|\mathcal{C}|^2)$.*

Proof. Suppose we are given an $\vee$-circuit $\mathcal{C}$ as input. The matrix A computed by $\mathcal{C}$ can be easily extracted from $\mathcal{C}$ in time $\tilde{O}(|\mathcal{C}|^2)$. We then simply output the trivial depth-1 $+$-circuit for A that has size at most $n^2 \leq |\mathcal{C}|^2$. $\square$

Rewriting and the strong exponential time hypothesis. The main technical ingredient in our proof is Lemma 9 below, which states that if subquadratic-time rewriting algorithms exist, then certain simple covering problems can be solved faster than in a trivial manner.

In the following we consider set systems defined by $L_1, \ldots, L_n$ and $R_1, \ldots, R_n$ that are (not necessarily distinct) subsets of $[m]$. We say that (i, j) is a *covering pair* if $L_j \cup R_i = [m]$.

Lemma 9. *Suppose we are given sets $L_1, \ldots, L_n, R_1, \ldots, R_n \subseteq [m]$ as input.*

*(a) If $(\vee, +)$-*Rewrite* can be solved in time $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ for some constant $\epsilon > 0$, then the number of covering pairs can be computed in time $\tilde{O}((nm)^{2-\epsilon})$.*

*(b) If $(\vee, \oplus)$-*Rewrite* can be solved in time $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ for some constant $\epsilon > 0$, then the parity of the number of covering pairs can be computed in time $\tilde{O}((nm)^{2-\epsilon})$.*

Proof. We begin by proving (a). Let $A = (a_{ij})$ be an $n \times n$ matrix defined by $a_{ij} = 1$ iff (i, j) is a covering pair. We show how to compute $|A|$ without constructing A explicitly.

Suppose for a moment that we had a small $+$-circuit $\mathcal{C}$ for A. The value $|A|$ can be recovered from the circuit $\mathcal{C}$ in time $\tilde{O}(|\mathcal{C}|)$ via the following trick: evaluate $\mathcal{C}$ (over the integers) on the all-1 vector $\mathbb{1}$ to obtain $y = \mathcal{C}(\mathbb{1}) \in \mathbb{N}^n$; but now

$$|A| = \mathbb{1}^\top A \mathbb{1} = \mathbb{1}^\top \mathcal{C}(\mathbb{1}) = y_1 + \cdots + y_n. \tag{9}$$

Unfortunately, we do not know how to construct a small $+$-circuit for A. Instead, our key observation below will be that *the complement matrix $\bar{A}$* admits an $\vee$-circuit $\mathcal{C}^\vee$ of size only $|\mathcal{C}^\vee| = O(nm)$. By assumption, we can then rewrite $\mathcal{C}^\vee$ as a $+$-circuit $\mathcal{C}^+$ in time $\tilde{O}(|\mathcal{C}^\vee|^{2-\epsilon}) = \tilde{O}((nm)^{2-\epsilon})$. In particular, the size of the new circuit must also be

$$|\mathcal{C}^+| = \tilde{O}\left((nm)^{2-\epsilon}\right).$$

Analogously to (9) we can then recover $|A|$ from $\mathcal{C}^+$ in time $\tilde{O}(|\mathcal{C}^+|)$:

$$|A| = n^2 - |\bar{A}| = n^2 - \mathbb{1}^\top \mathcal{C}^+(\mathbb{1}).$$

It now remains to describe how to construct $\mathcal{C}^\vee$ for $\bar{A}$ in time $\tilde{O}(nm)$. Define a depth-2 circuit $\mathcal{C}^\vee$ as follows: The 0-th layer of $\mathcal{C}^\vee$ hosts input gates l_j, $j \in [n]$; the 1-st layer contains intermediate gates g_k, $k \in [m]$; and the 2-nd layer contains output gates r_i, $i \in [n]$. Each input gate l_j is connected to gates g_k for $k \in [m] \smallsetminus L_j$; similarly, each output gate r_i is connected to gates g_k for $k \in [m] \smallsetminus R_i$. To see that $\mathcal{C}^\vee$ computes $\bar{A}$ note that there is a path from input l_i to output r_j iff there is a $k \in [m]$ such that $k \notin L_i \cup R_j$ iff (i, j) is not a covering pair. Note also that $|\mathcal{C}^\vee| \le 2nm$ and that the circuit can be constructed in time $\tilde{O}(nm)$.

Finally, we observe that (b) can be proven by the same argument as (a), except that the arithmetic is performed over GF(2). □

Next, we reduce #CNF-SAT and $\oplus$CNF-SAT to the covering problems in Lemma 9. Here we are essentially applying a technique of Williams [28, Theorem 5].

Theorem 10. *We have the following reductions:*

(a) If $(\vee, +)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some* $\epsilon > 0$*, then* #CNF-SAT *can be solved in time* $2^{(1-\epsilon/2)n} \operatorname{poly}(n,m)$*.*

(b) If $(\vee, \oplus)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some* $\epsilon > 0$*, then* $\oplus$CNF-SAT *can be solved in time* $2^{(1-\epsilon/2)n} \operatorname{poly}(n,m)$*.*

Proof. Let $\varphi = \{C_1, \ldots C_m\}$ be an instance of CNF-SAT over variables $x_1, \ldots, x_n$. Without loss of generality (by inserting one variable as necessary), we may assume that n is even. Call the variables $x_1, \ldots, x_{n/2}$ *left* variables and the variables $x_{n/2+1}, \ldots, x_n$ *right* variables.

For each truth assignment $s \in \{0,1\}^{n/2}$ to the left variables, let $L_s \subseteq \varphi$ be the set of clauses satisfied by s. Similarly, for assignment $t \in \{0,1\}^{n/2}$ to the right variables, let $R_t \subseteq \varphi$ be the set of clauses satisfied by t. Clearly, the compound assignment (s,t) to all the variables satisfies φ if and only if $L_s \cup R_t = \varphi$. That is, the number of satisfying assignments is precisely the number of covering pairs of the set system $\{L_s, R_t\}$, $s,t \in \{0,1\}^{n/2}$. Thus, both claims follow from Lemma 9. □

We can now finish the proof of Theorem 2. For $(\vee, +)$-Rewrite the result follows immediately from Theorem 10 since an algorithm for #CNF-SAT implies an algorithm for CNF-SAT. For $(\vee, \oplus)$-Rewrite, however, the result does not immediately follow from Theorem 10—it is non-trivial to convert an algorithm for $\oplus$CNF-SAT into an algorithm for CNF-SAT. Here we can invoke the main result (an "Isolation Lemma" for k-CNFs) of Calabro et al. [6]. They show that any $2^{(1-\epsilon)n} \operatorname{poly}(n,m)$ time algorithm for $\oplus$CNF-SAT (in fact, even for CNF-SAT under the promise that there is at most one satisfying assignment) can be turned into an $2^{(1-\epsilon')n} \operatorname{poly}(n,m)$ time randomised algorithm for CNF-SAT where $\epsilon' > 0$. (This is the only step in our proof that required the use of randomness.) This concludes the proof of Theorem 2.

Acknowledgements. We are grateful to Stasys Jukna and Igor Sergeev for comments and pointers to the literature. Furthermore, we thank Jukka Suomela for discussions, and the anonymous referee for comments.

This research is supported in part by Academy of Finland, grants 132380 and 252018 (M.G.), 132812 and 251170 (M.J.), 252083 and 256287 (P.K.), and by Helsinki Doctoral Programme in Computer Science - Advanced Computing and Intelligent Systems (J.K.).

References

[1] N. Alon, M. Karchmer, and A. Wigderson. Linear circuits over GF(2). *SIAM Journal on Computing*, 19(6):1064–1067, 1990. doi:10.1137/0219074.

[2] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, J. Nederlof, and P. Parviainen. Fast zeta transforms for lattices with few irreducibles. In *Proceedings of the*

23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), pages 1436–1444. SIAM, 2012.

[3] B. Bollobás. *Random Graphs.* Number 73 in Cambridge studies in advanced mathematics. Cambridge University Press, 2nd edition, 2001.

[4] J. Boyar and M. G. Find. Cancellation-free circuits in unbounded and bounded depth. In *Fundamentals of Computation Theory*, volume 8070 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 2013. doi:10.1007/978-3-642-40164-0_17.

[5] J. Boyar, P. Matthews, and R. Peralta. Logic minimization techniques with applications to cryptology. *Journal of Cryptology*, 26:280–312, 2013. doi:10.1007/s00145-012-9124-7.

[6] C. Calabro, R. Impagliazzo, V. Kabanets, and R. Paturi. The complexity of unique k-SAT: An isolation lemma for k-CNFs. *Journal of Computer and System Sciences*, 74(3):386–393, 2008. doi:10.1016/j.jcss.2007.06.015.

[7] C. Calabro, R. Impagliazzo, and R. Paturi. The complexity of satisfiability of small depth circuits. In *4th International Workshop on Parameterized and Exact Computation (IWPEC 2009)*, pages 75–85. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-11269-0_6.

[8] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlstrom. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity (CCC 2012)*, pages 74–84. IEEE, 2012. doi:10.1109/CCC.2012.36.

[9] A. Gál, K. A. Hansen, M. Koucký, P. Pudlák, and E. Viola. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 479–494. ACM, 2012. doi:10.1145/2213977.2214023.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[11] S. B. Gashkov and I. S. Sergeev. On the complexity of linear Boolean operators with thin matrices. *Journal of Applied and Industrial Mathematics*, 5:202–211, 2011. doi:10.1134/S1990478911020074.

[12] M. I. Grinchuk and I. S. Sergeev. Thin circulant matrixes and lower bounds on complexity of some Boolean operators. *Diskretnyĭ Analiz i Issledovanie Operatsiĭ*, 18:38–53, 2011.

[13] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

[14] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

[15] M. Järvisalo, P. Kaski, M. Koivisto, and J. H. Korhonen. Finding efficient circuits for ensemble computation. In *Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT 2012)*, pages 369–382. Springer, 2012. doi:10.1007/978-3-642-31612-8_28.

[16] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[17] S. Jukna and I. Sergeev. Complexity of linear boolean operators. *Foundations and Trends in Theoretical Computer Science*, 9(1):1–123, 2013. doi:10.1561/0400000063.

[18] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison–Wesley, 3rd edition, 1998.

[19] J. Kollár, L. Rónyai, and T. Szabó. Norm-graphs and bipartite Turán numbers. *Combinatorica*, 16(3):399–406, 1996. doi:10.1007/BF01261323.

[20] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[21] E. A. Lamagna and J. E. Savage. Computational complexity of some monotone functions. In *IEEE Conference Record of 15th Annual Symposium on Switching and Automata Theory*, pages 140–144, 1974. doi:10.1109/SWAT.1974.9.

[22] O. B. Lupanov. On rectifier and switching-and-rectifier schemes. In *Doklady Akademii Nauk SSSR*, volume 111, pages 1171–1174, 1956. In Russian.

[23] K. Mehlhorn. Some remarks on Boolean sums. *Acta Informatica*, 12:371–375, 1979. doi:10.1007/BF00268321.

[24] É. I. Nechiporuk. On a Boolean matrix. *Systems Theory Research*, 21:236–239, 1971.

[25] M. Pătraşcu and R. Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 1065–1075. SIAM, 2010.

[26] N. Pippenger. On another Boolean matrix. *Theoretical Computer Science*, 11(1): 49–56, 1980. doi:10.1016/0304-3975(80)90034-1.

[27] N. Pippenger. On the evaluation of powers and monomials. *SIAM Journal on Computing*, 9(2):230–250, 1980. doi:10.1137/0209022.

[28] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2–3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.