Cubic Formula Size Lower Bounds Based on Compositions with Majority

Anna Gál[1]
The University of Texas at Austin, Austin, TX, USA
panni@cs.utexas.edu

Avishay Tal[2]
Stanford University, Palo Alto, CA, USA
avishay.tal@gmail.com

Adrian Trejo Nuñez
The University of Texas at Austin, Austin, TX, USA
atrejo@cs.utexas.edu
https://orcid.org/0000-0002-5658-9956

Abstract

We define new functions based on the Andreev function and prove that they require $n^3/\operatorname{polylog}(n)$ formula size to compute. The functions we consider are generalizations of the Andreev function using compositions with the majority function. Our arguments apply to composing a hard function with any function that agrees with the majority function (or its negation) on the middle slices of the Boolean cube, as well as iterated compositions of such functions. As a consequence, we obtain $n^3/\operatorname{polylog}(n)$ lower bounds on the (non-monotone) formula size of an explicit monotone function by combining the monotone address function with the majority function.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography, Theory of computation → Circuit complexity

Keywords and phrases formula lower bounds, random restrictions, KRW conjecture, composition

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.35

Acknowledgements We thank an anonymous referee of a previous version of this paper for suggesting to apply our results to prove cubic formula size lower bounds for a monotone function.

1 Introduction

We study the problem of proving lower bounds on the De Morgan formula size of explicit functions.

While it is known that almost all Boolean functions of n variables require formula size exponential in n, proving lower bounds on the formula size of specific functions remains a major challenge. The current largest lower bounds on De Morgan formula size for explicitly defined functions are of the form $n^{3-o(1)}$. Lower bounds for general formula size are weaker, throughout this paper we only consider De Morgan formulas, but sometimes we just refer to them as “formulas”.

[1] Part of this work was done while visiting the Simons Institute for the Theory of Computing.
[2] Supported by a Motwani Postdoctoral Fellowship and by NSF grant CCF-1763299. Part of this work was done while visiting the Simons Institute for the Theory of Computing.

10th Innovations in Theoretical Computer Science (ITCS 2019).
Editor: Avrim Blum; Article No. 35; pp. 35:1–35:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

History

Formula size lower bounds have a long history. One of the methods for proving formula size lower bounds is based on shrinkage of De Morgan formulas under random restrictions. This method was introduced by Subbotovskaya [17] who gave a $\Omega\big(n^{1.5}\big)$ lower bound on the De Morgan formula size of the parity function. The lower bound for parity has been improved by Khrapchenko [10] to $\Omega\big(n^2\big)$. However, it is also known that Khrapchenko's method cannot give larger than quadratic lower bounds. The method of random restrictions on the other hand has led to the currently known largest lower bounds on formula size. Andreev [1] used random restrictions to prove an $\Omega\big(n^{2.5-o(1)}\big)$ lower bound for a function obtained by composing parity with an arbitrary other function f where f is specified as part of the input. We give more formal definitions in Section 2. After improvements of the bound by [8, 13], Håstad [5] proved a lower bound of the form $n^{3-o(1)}$ for the Andreev function. Tal [18] improved the lower order terms to give a $\Omega\Big(\frac{n^3}{(\log n)^2 \log\log n}\Big)$ lower bound for the Andreev function, which is tight up to the $\log\log n$ term. Tal [19] gave a slightly larger lower bound of the form $\Omega\Big(\frac{n^3}{\log n(\log\log n)^2}\Big)$ for another function introduced by Komargodski and Raz [11]. This function is similar to the Andreev function, it still composes parity with other functions specified as part of the input. The difference is that instead of specifying the function f by its entire truth table as part of the input, an error correcting code is used to derive the truth table from the input. Bogdanov [2] showed that the same $\Omega\Big(\frac{n^3}{\log n(\log\log n)^2}\Big)$ lower bound can also be obtained for any "small-biased" function, that is any randomized function whose distribution of truth tables is small biased. He also noted that standard constructions of small biased sets yield explicit families of such functions. [3, 12] showed that parity in Andreev's function can be replaced with any good enough bit fixing extractor, and the resulting function still requires $n^3/\operatorname{polylog}(n)$ formula size.

Other than Bogdanov's functions, the only explicit function with $n^{3-o(1)}$ formula size lower bounds has been the Andreev function, and its variants using error correcting codes by [11, 19] or bit fixing extractors [3, 12].

Dinur and Meir [4] gave a new proof of $n^{3-o(1)}$ formula size lower bounds for the Andreev function, based on information theoretic arguments. The bound obtained by their argument is of the form $\Omega\Big(\frac{n^3}{2^{\sqrt{\log n}}\operatorname{poly}\log\log n}\Big)$ which is weaker in the lower order terms than the bounds of Håstad [5] and Tal [18]. But their goal was to give a proof that could possibly generalize to other function compositions, which would be important in light of the KRW conjecture [9] (see Section 5). Our results can be viewed as a step in this direction.

Our Results

In this paper we obtain $n^3/\operatorname{polylog}(n)$ lower bounds on a new class of functions. First we consider an extension of the Andreev function which we call "Generalized Andreev function with Majority", using the majority function instead of parity in the function compositions. We define the function formally in Section 2. As far as we know this function has not been studied before, and previous approaches do not directly work to obtain our bounds.

Next we extend our results to composing a hard function with any function that agrees with the majority function (or its negation) on the middle slices of the Boolean cube, as well as iterated compositions of such functions. Since parity agrees with majority on the two middle slices of the Boolean cube, our argument also applies to parity (the original Andreev function), and composing parity with majority in various ways.

As another consequence, we prove $n^3/\operatorname{polylog}(n)$ lower bounds on the (non-monotone) formula size of the monotone function obtained by combining the monotone address function of Wegener [21] with the majority function.

It was pointed out to us by Pavel Pudlak [14], that for any function $f : \{0,1\}^n \to \{0,1\}$, one can construct a function $f' : \{0,1\}^{2n} \to \{0,1\}$ that is monotone, and has formula size at least as large as f. Consider inputs of the form (x, y) where $x, y \in \{0,1\}^n$ and simply let $f'(x, y) = 1$ on all inputs of Hamming weight greater than n, $f'(x, y) = 0$ on all inputs of Hamming weight less than n, and $f'(x, y) = f(x)$ on inputs (x, y) with Hamming weight n. Observe that f' has formula size at least as large as f: identifying for each $i \in \{1, \ldots, n\}$ the literals x_i and $\neg y_i$, and similarly $\neg x_i$ and y_i, we get f from f'. However, as far as we know, our results give the first super-quadratic formula size lower bound with a direct proof for an explicitly defined monotone function.

Our argument gives a formal proof that the monotone formula size of the majority function is at least $n^{\Gamma_{\text{mon}}}/\operatorname{polylog} n$, where Γ_{mon} denotes the shrinkage exponent of monotone formulas under random restrictions. It is a long standing open problem to determine the value of Γ_{mon}. It is also open to obtain tight bounds on the formula size of majority, both in the monotone and non-monotone case. The current best lower bound for both monotone and non-monotone formulas computing majority of n bits is $\Omega(n^2)$. The best upper bound on the De Morgan formula size of majority on n bits is $\mathcal{O}\left(n^{3.91}\right)$ [16]. Considering monotone formulas for majority, the best upper bound remains the $\mathcal{O}\left(n^{5.3}\right)$ bound by Valiant [20]. Håstad [5] noted that determining the value of Γ_{mon} is likely to yield improved lower bounds on the monotone formula size of the majority function. Our results make this connection explicit, independently of how the value Γ_{mon} is obtained.

Our argument is based on random restrictions and analyzing the shrinkage of formula size under restrictions. However, the main obstacle in applying previous arguments is that we need random restrictions that leave each Majority undetermined. Previously considered restrictions are far from achieving this. Instead of standard random restrictions, we use "staged" random restrictions, and adjust their results to enforce more structure. The idea of building restrictions in stages appears before in [7, 3, 12]. The main difference in our approach is that we maintain the structure of the composed hard function with majority after each stage by performing some clean-up procedure.

In addition to worst case formula size lower bounds, average case lower bounds have been shown in [3, 11, 12, 19]. These bounds are quantitatively weaker than the $n^{3-o(1)}$ worst case bounds but provide high probability versions of the shrinkage results under certain structured random reductions. Tal [19] has shown that average case bounds can be used to obtain stronger worst case bounds, and in fact the current largest lower bounds of the form $\Omega\left(\frac{n^3}{\log n(\log\log n)^2}\right)$ by Tal [19] and Bogdanov [2] were obtained this way.

2 Definitions and Background

Given an n-bit string $\vec{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$, let $\operatorname{wt}(\vec{x})$ denote the Hamming weight of $\vec{x}$, defined as

$$\operatorname{wt}(\vec{x}) = |\{i : x_i = 1\}|$$

Let $\mathcal{B}_n = \{f : \{0,1\}^n \to \{0,1\}\}$ denote the set of all Boolean functions on n bits.

Given a bm-bit string $\vec{x}$, we can interpret $\vec{x}$ as a $b \times m$ matrix with rows $\vec{x}_1, \ldots, \vec{x}_b$ of m bits each. If $f \in \mathcal{B}_b$ and $g \in \mathcal{B}_m$ are arbitrary functions, let $f \circ g : \{0,1\}^{b\times m} \to \{0,1\}$ denote their composition, defined as

$$(f \circ g)(\vec{x}_1, \ldots, \vec{x}_b) = f(g(\vec{x}_1), \ldots, g(\vec{x}_b))$$

Given a function $f \in \mathcal{B}_n$, let $\mathrm{tt}(f)$ denote the truth table of f, defined as the string of length 2^n specifying the output of f on all strings $\vec{x} \in \{0,1\}^n$ in lexicographic order. We use f and $\mathrm{tt}(f)$ interchangeably when f is an input to another function.

Let $\oplus_m : \{0,1\}^m \to \{0,1\}$ denote the parity function on m bits.

Let $\mathrm{Maj}_m : \{0,1\}^m \to \{0,1\}$ denote the majority function on m bits, defined as

$$\mathrm{Maj}_m(\vec{x}) = \begin{cases} 1 & \text{if } \mathrm{wt}(\vec{x}) \geq \lceil \frac{m}{2} \rceil \\ 0 & \text{otherwise} \end{cases}$$

2.1 Andreev Function

Let $\mathcal{A}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the Andreev function on $2n$ bits. Let $b = \log n$ and $m = n/b = n/\log n$. If $f \in \mathcal{B}_b$, then $|\mathrm{tt}(f)| = 2^b = 2^{\log n} = n$.

The function $\mathcal{A}_n$ takes two inputs: an n-bit string representing the truth table of a function f on b bits, and an n-bit string $\vec{x}$, interpreted as a $b \times m$ matrix with rows $\vec{x}_1, \ldots, \vec{x}_b$. Then,

$$\mathcal{A}_n(f, \vec{x}) = (f \circ \oplus_m)(\vec{x}) = f(\oplus_m(\vec{x}_1), \ldots, \oplus_m(\vec{x}_b))$$

2.2 Generalized Andreev Function

Let $b = \log n$ and $m = n/b$ as before. If $g_m \in \mathcal{B}_m$ is an arbitrary function on m bits, then let $\mathcal{A}_n^{g_m} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the generalized Andreev function on $2n$ bits, defined analogously by

$$\mathcal{A}_n^{g_m}(f, \vec{x}) = (f \circ g_m)(\vec{x}) = f(g_m(\vec{x}_1), \ldots, g_m(\vec{x}_b))$$

In particular, $\mathcal{A}_n = \mathcal{A}_n^{\oplus_m}$.

Let $\mathcal{M}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the generalized Andreev function with Maj_m in place of g_m. That is

$$\mathcal{M}_n(f, \vec{x}) = \mathcal{A}_n^{\mathrm{Maj}_m}(f, \vec{x}) = (f \circ \mathrm{Maj}_m)(\vec{x}) = f(\mathrm{Maj}_m(\vec{x}_1), \ldots, \mathrm{Maj}_m(\vec{x}_b))$$

If $f \in \mathcal{B}_b$ is a fixed function, define $\mathcal{M}_{n,f} : \{0,1\}^n \to \{0,1\}$ as

$$\mathcal{M}_{n,f}(\vec{x}) = \mathcal{M}_n(f, \vec{x})$$

or equivalently, $\mathcal{M}_{n,f} = f \circ \mathrm{Maj}_m$.

2.3 De Morgan Formulas

Formulas are tree-like circuits, that is circuits where each gate has fan-out at most one. A De Morgan formula is a formula that uses only **AND**, **OR** and negation gates, where the gates have fan-in at most 2. Let $f \in \mathcal{B}_n$ be an arbitrary function. Define $\mathcal{L}(f)$ to be the formula complexity of f, which is the minimum number of leaves required by any De Morgan formula computing f.

It is known that almost all Boolean functions on n variables require De Morgan formula size at least $\frac{2^n}{2 \log n}$ [15].

2.4 Random Restrictions and Shrinkage

Consider a function $f \in \mathcal{B}_n$ and let $S = \{x_1, \ldots, x_n\}$ denote the variables of f. A restriction on S is a function $\rho : S \to \{0, 1, \star\}$. Let $f\lceil_\rho$ denote the function obtained from f by fixing inputs x_i to $\rho(x_i)$ if $\rho(x_i) \neq \star$, which depends only on the inputs x_i for which $\rho(x_i) = \star$. Given arbitrary functions $g \in \mathcal{B}_m$ and $f \in \mathcal{B}_n$ for $m \leq n$, we say that f computes g as a sub-function if g can be achieved as a restriction of f.

A random p-restriction on S is a randomly generated restriction ρ where

$$\Pr(\rho(x_i) = \star) = p$$
$$\Pr(\rho(x_i) = 0) = \Pr(\rho(x_i) = 1) = \frac{1-p}{2}$$

uniformly and independently for all $x_i \in S$. Let $\mathcal{R}_p$ denote the distribution of all uniformly generated random p-restrictions.

Subbotovskaya [17] proved that for any Boolean function $f \in \mathcal{B}_n$ it holds that

$$\mathop{\mathbb{E}}_{\rho \sim \mathcal{R}_p}\left[\mathcal{L}\left(f\lceil_\rho\right)\right] = \mathcal{O}(p^\Gamma \mathcal{L}(f))$$

for $\Gamma = 3/2$. The constant Γ is called the shrinkage exponent, which is the largest number for which the statement is true. After several improvements [8, 13], Håstad [5] proved that $\Gamma = 2$. The following version is due to Tal [18].

▶ **Theorem 2.1** (Shrinkage Lemma [18]). *Let $f \in \mathcal{B}_n$ be an arbitrary function. Then, $\forall p > 0$,*

$$\mathop{\mathbb{E}}_{\rho \sim \mathcal{R}_p}\left[\mathcal{L}\left(f\lceil_\rho\right)\right] \leq \mathcal{O}(1 + p^2 \mathcal{L}(f)) \tag{1}$$

▶ **Corollary 2.2.** *Let $f \in \mathcal{B}_n$ be an arbitrary function. Then, $\exists c > 0$ such that for $\forall p > 0$ and large enough n,*

$$\Pr_{\rho \sim \mathcal{R}_p}\left(\mathcal{L}\left(f\lceil_\rho\right) \geq 10c(1 + p^2 \mathcal{L}(f))\right) \leq \frac{1}{10} \tag{2}$$

Proof. Let $c > 0$ be chosen such that $\mathbb{E}_{\rho \sim \mathcal{R}_p}\left[\mathcal{L}\left(f\lceil_\rho\right)\right] \leq c(1 + p^2 \mathcal{L}(f))$. Then, by Markov's inequality: $\Pr_{\rho \sim \mathcal{R}_p}\left(\mathcal{L}\left(f\lceil_\rho\right) \geq 10c(1 + p^2 \mathcal{L}(f))\right) \leq \frac{1}{10}$. ◀

2.5 Concentration Inequalities

We use the following result on bounds of sums of random variables.

▶ **Theorem 2.3** (Hoeffding's Inequality [6]). *Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ for $1 \leq i \leq n$ and let $X = \sum_{i=1}^n X_i$. Then,*

$$\Pr\left(\left|X - \mathbb{E}[X]\right| \geq t\right) \leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \tag{3}$$

3 Composition with Majority

Let $\mathcal{M}_n$ be the generalized Andreev function with majority. Let $b = \log n$ and $m = n/b = n/\log n$ and assume $b, m \in \mathbb{N}$. Let $h \in \mathcal{B}_b$ be a function of maximum formula complexity and consider $\mathcal{M}_{n,h} = h \circ \mathrm{Maj}_m$. Since $\mathcal{M}_{n,h}$ is a sub-function of $\mathcal{M}_n$, we have $\mathcal{L}(\mathcal{M}_n) \geq \mathcal{L}(\mathcal{M}_{n,h}) = \mathcal{L}(h \circ \mathrm{Maj}_m)$. Thus, it suffices to prove a lower bound on the formula complexity of $h \circ \mathrm{Maj}_m$. Indeed, this will be our strategy (which is standard when proving lower bounds for Andreev-type functions). Our main result is the following general theorem, that may also be applied in other scenarios.

▶ **Theorem 3.1** (Formula Size of Composition with Majority). *Let $b, m \in \mathbb{N}$ and $h \in \mathcal{B}_b$ be non-constant. Then,*

$$\mathcal{L}(h \circ \mathrm{Maj}_m) \geq \mathcal{L}(h) \cdot m^2 / \operatorname{polylog}(b \cdot m)$$

Since the hardest functions on $b = \log n$ bits have formula complexity at least $\frac{n}{2 \log \log n}$ [15], Theorem 3.1 implies that

$$\mathcal{L}(\mathcal{M}_n) \geq \mathcal{L}(h \circ \mathrm{Maj}_m) \geq \mathcal{L}(h) \cdot m^2 / \operatorname{polylog}(b \cdot m) \geq n^3 / \operatorname{polylog}(n)$$

The rest of this section is devoted to the proof of Theorem 3.1.

Warmup

Let $n = mb$. The input $\vec{x} = (x_1, \ldots, x_n)$ is divided into b contiguous blocks $B_1, \ldots, B_b$ of m variables each. In order to apply a random restriction based argument to $h \circ \mathrm{Maj}_m$, we wish to prove that there exists a restriction ρ that leaves each Maj_m undetermined and the resulting formula shrinks by a factor of $\Omega(m^2 / \operatorname{polylog}(bm))$.

A single majority is left undetermined by ρ if the absolute difference between the number of variables assigned 0 and 1 is at most the number of unassigned variables. Otherwise, the majority value is already set and there are not enough remaining variables to flip it.

3.1 Random p-Restrictions

Previous random restriction based arguments typically use random p-restrictions defined in Section 2.4. We start by some observations about them. Let $\rho \in \mathcal{R}_p$ be a random p-restriction on $S = \{x_1, \ldots, x_n\}$ and let $B_i = \{x_{i_1}, \ldots, x_{i_m}\}$ be a fixed block of the input.

Let X_{ik} and Y_{ik} for $1 \leq k \leq m$ be the following random variables:

$$X_{ik} = \begin{cases} 1 & \text{if } \rho(x_{i_k}) = \star \\ 0 & \text{otherwise} \end{cases} \qquad Y_{ik} = \begin{cases} 1 & \text{if } \rho(x_{i_k}) = 0 \\ -1 & \text{if } \rho(x_{i_k}) = 1 \\ 0 & \text{if } \rho(x_{i_k}) = \star \end{cases}$$

Then,

$$X_i = \sum_{k=1}^{m} X_{ik} \qquad Y_i = \sum_{k=1}^{m} Y_{ik}$$

$$\mathbb{E}[X_i] = \sum_{k=1}^{m} \mathbb{E}[X_{ik}] = mp \qquad \mathbb{E}[Y_i] = \sum_{k=1}^{m} \mathbb{E}[Y_{ik}] = 0$$

We note that X_i and Y_i are not necessarily independent, since for any $\ell \geq 0$

$$X_i \geq \ell \implies |Y_i| \leq m - \ell$$

Since $0 \leq X_{ik} \leq 1$, Theorem 2.3 gives:

$$\Pr_{\rho \in \mathcal{R}_p} (|X_i - mp| > t) \leq 2 \exp(-2t^2/m)$$

To obtain lower bounds of the form $\mathcal{L}(h) \cdot \frac{m^2}{\mathrm{polylog}(mb)}$ by a single round of p-restrictions, one would need $p = \mathcal{O}(\mathrm{polylog}(mb)/m)$, thus $X_i = \Theta(\mathrm{polylog}(mb))$ would hold with high probability. Since $|Y_i|$ is typically $\Omega(\sqrt{m})$, it is likely that $\mathrm{Maj}_m(B_i\lceil_\rho)$ is constant.

Since one p-restriction cannot shrink the formula size sufficiently and leave each majority undetermined, we will build such a restriction incrementally instead.

3.2 Staged p-Restrictions

Proof of Theorem 3.1. Let c' be a large constant to be defined later. We first deal with the case that $\mathcal{L}(h) \leq 2c'$. Then, since for non-constant h, Maj_m (or its negation) is a sub-function of $h \circ \mathrm{Maj}_m$ and since $\mathcal{L}(\mathrm{Maj}_m) \geq \Omega(m^2)$ ([10]) we get

$$\mathcal{L}(h \circ \mathrm{Maj}_m) \geq \mathcal{L}(\mathrm{Maj}_m) \geq \Omega(m^2) \geq \Omega(\mathcal{L}(h) \cdot m^2)$$

which completes the proof in this case. In the following, we shall assume that $\mathcal{L}(h) > 2c'$.

We define the following procedure that runs in t stages: in the j-th stage, we generate a p_j-restriction ρ_j such that, with high probability, the formula has enough unrestricted variables to balance the number of 0's and 1's and leave enough variables unrestricted for stage $j+1$.

Setting Up Parameters

We set

$$m_1 = m$$

and

$$m_{j+1} = m_j^{0.6}$$

for $j \geq 1$ as long as $m_j \geq \log^5(4b)$. Let t be the last j such that $m_j \geq \log^5(4b)$. A small calculation shows that $t \leq 2 \log\log m$. For $j = 1, \ldots, t$ we set

$$p_j = 4m_j^{-0.4} = 4m_{j+1}/m_j$$

Shrinkage In t Stages

Denote by $\varphi_1 = h \circ \mathrm{Maj}_m$. For $j = 1, \ldots, t$, we show how to construct φ_{j+1} over variables S_{j+1} from φ_j over S_j. We show by induction that $\varphi_{j+1} = h \circ \mathrm{Maj}_{m_{j+1}}$ (up to a renaming of the variables) and that

$$\mathcal{L}(\varphi_{j+1}) \leq c \cdot \left(\frac{m_{j+1}}{m_j}\right)^2 \cdot \mathcal{L}(\varphi_j),$$

for some large enough universal constant $c > 0$.

Let $j \in \{1 \ldots, t\}$. Let $\rho_j \in \mathcal{R}_{p_j}$ be a random p_j-restriction over S_j. Let

$$X_i^j = \sum_{k=1}^{m_j} X_{ik}^j \qquad\qquad Y_i^j = \sum_{k=1}^{m_j} Y_{ik}^j$$

for $i = 1, \ldots, b$ be defined analogously as in the previous section for block B_i. Then,

$$\mathbb{E}\left[X_i^j\right] = m_j p_j = 4m_{j+1}$$

Let $E_{j,i}$ denote the event that $\left|X_i^j - m_j p_j\right| \le \frac{1}{4} m_j p_j$ and let $F_{j,i}$ denote the event that $\left|Y_i^j\right| \le \frac{1}{2} m_j p_j$. By Theorem 2.3, using the assumption $m_j \ge \log^5(4b)$,

$$\Pr_{\rho_j \in \mathcal{R}_{p_j}}(E_{j,i}) \ge 1 - 2e^{-2\frac{m_{j+1}^2}{m_j}} = 1 - 2e^{-2m_j^{0.2}} \ge 1 - \frac{1}{4b} \tag{4}$$

$$\Pr_{\rho_j \in \mathcal{R}_{p_j}}(F_{j,i}) \ge 1 - 2e^{-2\frac{(2m_{j+1})^2}{4m_j}} = 1 - 2e^{-2m_j^{0.2}} \ge 1 - \frac{1}{4b} \tag{5}$$

By Corollary 2.2, there exists some constant $c' > 0$ such that

$$\Pr_{\rho_j \in \mathcal{R}_{p_j}}\left(\mathcal{L}\left(\varphi_j\lceil_{\rho_j}\right) \le c'(1 + p_j^2\,\mathcal{L}(\varphi_j))\right) \ge 0.9$$

Let H_j denote the event that $\mathcal{L}\left(\varphi_j\lceil_{\rho_j}\right) \le c' \cdot (1 + p_j^2 \cdot \mathcal{L}(\varphi_j))$. Thus $\Pr[H_j] \ge 0.9$. By the union bound, there exists a restriction ρ_j for which H_j and all $E_{j,i}, F_{j,i}$ for $i = 1, \ldots, b$ hold simultaneously. Fix such a restriction ρ_j. Now, since $E_{j,i}$ holds, then

$$X_i^j \ge \frac{3}{4} m_j p_j$$

Since $F_{j,i}$ also holds, then we can make the number of 0's and 1's equal by fixing at most $\frac{1}{2} m_j p_j$ variables appropriately, leaving at least $\frac{1}{4} m_j p_j = m_{j+1}$ unrestricted variables in the block. We restrict the remaining variables further to leave exactly m_{j+1} unrestricted variables by assigning an equal number of them 0 and 1 in some arbitrary process. Take φ_{j+1} to be the restricted function.

Since H_j holds, we get

$$\mathcal{L}(\varphi_{j+1}) \le \mathcal{L}\left(\varphi_j\lceil_{\rho_j}\right) \le c' \cdot (1 + p_j^2 \cdot \mathcal{L}(\varphi_j))$$

However, since h is a sub-function of φ_{j+1} and since we assumed that $\mathcal{L}(h) > 2c'$, we get that $c' < \frac{1}{2}\,\mathcal{L}(\varphi_{j+1})$. Thus,

$$\mathcal{L}(\varphi_{j+1}) < \tfrac{1}{2}\,\mathcal{L}(\varphi_{j+1}) + c' \cdot p_j^2 \cdot \mathcal{L}(\varphi_j)$$

which implies that $\mathcal{L}(\varphi_{j+1}) < 2c' \cdot p_j^2 \cdot \mathcal{L}(\varphi_j)$ and we get

$$\mathcal{L}(\varphi_{j+1}) \le 2c' \cdot \left(\frac{4m_{j+1}}{m_j}\right)^2 \cdot \mathcal{L}(\varphi_j) = c \cdot \left(\frac{m_{j+1}}{m_j}\right)^2 \cdot \mathcal{L}(\varphi_j)$$

for any $j \in \{1, \ldots, t\}$ by setting $c = 2c' \cdot 16$. Overall, we get

$$\mathcal{L}(\varphi_{t+1}) \le c^t \cdot \left(\frac{m_{t+1}}{m}\right)^2 \cdot \mathcal{L}(\varphi)$$

Since h is a sub-function of φ_{t+1}, the formula size of φ_{t+1} is at least $\mathcal{L}(h)$, which gives

$$\mathcal{L}(\varphi) \ge c^{-t} \cdot \left(\frac{m}{m_{t+1}}\right)^2 \cdot \mathcal{L}(h)$$

Using $m_{t+1} < \log^5(4b)$ and $t \le 2\log\log m$ we get

$$\mathcal{L}(\varphi) \ge c^{-2\log\log m} \cdot \left(\frac{m}{\log^5(4b)}\right)^2 \cdot \mathcal{L}(h) \ge \mathcal{L}(h) \cdot m^2/\operatorname{polylog}(b \cdot m) \qquad ◀$$

In the above proof, we only use two facts about the majority function. First, we use that the values of the m-bit majority function are 0 on inputs $\vec{x}$ with Hamming weight $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil - 1$ and 1 on inputs with $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil$. In addition, (at the beginning of our proof) we use that $\mathcal{L}(\mathrm{Maj}_m) \geq \Omega(m^2)$ [10]. Thus our argument extends to any function with these two properties. It turns out that the first condition we need implies the second. Let $g_m \in \mathcal{B}_m$ be any function such that $g_m(\vec{x}) = 0$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil - 1$, and $g_m(\vec{x}) = 1$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil$. Then by Khrapchenko's theorem [10] the De Morgan formula size of g_m is at least $\Omega(m^2)$.

One can also think of such functions as a partial function that generalizes both Majority and Parity. We obtain the following.

▶ **Theorem 3.2.** *Let $m = \frac{n}{\log n}$ and let $g_m \in \mathcal{B}_m$ be any function such that $g_m(\vec{x}) = 0$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil - 1$, and $g_m(\vec{x}) = 1$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil$. Then,*

$$\mathcal{L}(\mathcal{A}_n^{g_m}) \geq n^3 / \operatorname{polylog}(n)$$

4 Consequences

4.1 Composition with Other Threshold Functions

We also obtain lower bounds for compositions with arbitrary threshold functions $\mathrm{Th}_{m,k}$ instead of Maj_m. We use that $\mathrm{Th}_{2k+1,k}$ is a subfunction of $\mathrm{Th}_{m,k}$. Fixing arbitrary $m - (2k+1)$ bits to 0 in each block, our results immediately imply that $\mathcal{L}(h \circ \mathrm{Th}_{m,k}) \geq \mathcal{L}(h) \cdot k^2 / \operatorname{polylog}(k, b)$. We get stronger bounds by noticing that fixing the $m - (2k+1)$ heaviest variables in each block, the formula shrinks by a factor of $(2k+1)/m$. Thus, we get

$$\mathcal{L}(h \circ \mathrm{Th}_{m,k}) \geq \mathcal{L}(h) \cdot m \cdot k / \operatorname{polylog}(k, b)$$

This implies the following:

▶ **Theorem 4.1.** *Let $m = \frac{n}{\log n}$ and $k \leq m/2$. Then*

$$\mathcal{L}\left(\mathcal{A}_n^{\mathrm{Th}_{m,k}}\right) \geq n^2 \cdot k / \operatorname{polylog}(n)$$

4.2 Iterated Compositions

Since the composed function "Parity of Parities" is just Parity, considering iterated compositions in place of Parity in the original lower bound arguments for Andreev function did not give new functions. But taking iterated compositions of Majorities yield new functions, such as "Majority of Majorities", "Parity of Majorities", "Majority of Parities" and so on. Our results extend to the generalized Andreev function with iterated compositions in place of g_m. We obtain additional functions with cubic formula size lower bounds.

▶ **Theorem 4.2.** *Let $\mathcal{G}_m$ denote the set of functions $g_m \in \mathcal{B}_m$ such that $g_m(\vec{x}) = 0$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil - 1$, and $g_m(\vec{x}) = 1$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil$; or the other way around, that is $g_m(\vec{x}) = 1$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil - 1$, and $g_m(\vec{x}) = 0$ when $\mathrm{wt}(\vec{x}) = \lceil \frac{m}{2} \rceil$.*

Let $m = \frac{n}{\log n}$ and let $u \geq 2$ and $v \geq 2$ be integers such that $uv = m$. For any functions $f_u \in \mathcal{G}_u$ and $g_v \in \mathcal{G}_v$

$$\mathcal{L}(\mathcal{A}_n^{f_u \circ g_v}) \geq n^3 / \operatorname{polylog}(n)$$

Proof. Let $h \in \mathcal{B}_{\log n}$ be a function of maximum formula complexity. By Theorem 3.1

$$\mathcal{L}(h \circ f_u) \geq \mathcal{L}(h) \cdot u^2 / \operatorname{polylog}(b \cdot u)$$

where $b = \log n$. Let $b' = b \cdot u$. By Theorem 3.1

$$\mathcal{L}(h \circ f_u \circ g_v) \geq \mathcal{L}(h \circ f_u) \cdot v^2 / \operatorname{polylog}(b' \cdot v)$$

Thus,

$$\mathcal{L}(h \circ f_u \circ g_v) \geq \mathcal{L}(h) \cdot \frac{u^2}{\operatorname{polylog}(b \cdot u)} \cdot \frac{v^2}{\operatorname{polylog}(b \cdot u \cdot v)} \geq \mathcal{L}(h) \cdot \frac{m^2}{\operatorname{polylog}(n)} \geq \frac{n^3}{\operatorname{polylog}(n)} \quad ◀$$

The argument extends to repeated iterations. As the proof shows, we lose a polylog(n) factor from the n^3 lower bound at each iteration.

4.3 Cubic Formula Size Lower Bounds for an Explicit Monotone Function

A function $h : \{0,1\}^b \to \{0,1\}$ is called a *slice function* if on inputs $\vec{z} \in \{0,1\}^b$, $h(\vec{z}) = 1$ if $\operatorname{wt}(\vec{z}) \geq \lfloor \frac{b}{2} \rfloor + 1$, and $h(\vec{z}) = 0$ if $\operatorname{wt}(\vec{z}) < \lfloor \frac{b}{2} \rfloor$. Note that every slice function is monotone, and slice functions differ from each other only on inputs in the middle layer of the Boolean cube, that is on inputs with weight exactly $\lfloor \frac{b}{2} \rfloor$.

The *monotone address function* defined by Wegener [21] takes $b + n$ input bits where $n = \binom{b}{\lfloor \frac{b}{2} \rfloor}$. The n bits are interpreted to specify a slice function h on b bits. We denote by h both the n-bit string and the slice function specified by it. Then, on input (z, h) where $z \in \{0,1\}^b$ and $h \in \{0,1\}^n$, the output of the monotone address function is $h(z)$. Note that the monotone address function itself is monotone.

We are now ready to define a monotone function that requires cubic formula size. Let $n = \binom{b}{\lfloor b/2 \rfloor}$, and let $m = n/b$. Similarly to the Generalized Andreev Function, we define a function $\mathcal{F}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on $2n$ bits.

The function $\mathcal{F}_n$ takes two inputs: an n-bit string representing a slice function h on b bits, and an n-bit string $\vec{x}$, interpreted as a $b \times m$ matrix with rows $\vec{x}_1, \ldots, \vec{x}_b$. Then,

$$\mathcal{F}_n(h, \vec{x}) = (h \circ \operatorname{Maj}_m)(\vec{x}) = h(\operatorname{Maj}_m(\vec{x}_1), \ldots, \operatorname{Maj}_m(\vec{x}_b))$$

We can further generalize this as follows: If $g_m \in \mathcal{B}_m$ is an arbitrary function on m bits, then let $\mathcal{F}_n^{g_m} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the function on $2n$ bits, defined analogously by

$$\mathcal{F}_n^{g_m}(h, \vec{x}) = (h \circ g_m)(\vec{x}) = h(g_m(\vec{x}_1), \ldots, g_m(\vec{x}_b))$$

In particular, $\mathcal{F}_n = \mathcal{F}_n^{\operatorname{Maj}_m}$. Note that for any monotone function g_m, the function $\mathcal{F}_n^{g_m}$ is also monotone.

Since the number of De Morgan formulas of size s on b input bits is at most $(cb)^s$ for some constant c [15], and the number of different slice functions on b input bits is 2^n where $n = \binom{b}{\lfloor b/2 \rfloor}$, by a standard counting argument, there are slice functions on b bits that require formula size at least $\Omega(\frac{n}{\log b}) = \Omega(\frac{n}{\log \log n})$.

This implies the following bound on the formula size of the monotone function $\mathcal{F}_n$.

▶ **Theorem 4.3.**

$$\mathcal{L}(\mathcal{F}_n) \geq n^3 / \operatorname{polylog}(n)$$

4.4 Monotone Formula Size of Majority

Our results highlight again the question of determining the shrinkage exponent for monotone formulas, raised by Håstad [5]. It was pointed out by Håstad [5], that determining the shrinkage exponent for monotone formulas could potentially yield improved lower bounds on the monotone formula size of the Majority function. Our results make this connection explicit, without any dependence on how the value of the shrinkage exponent is obtained. More precisely, our arguments imply the following.

▶ **Theorem 4.4.** *Let Γ_{mon} denote the shrinkage exponent of monotone formulas. Then $\mathcal{L}_{\text{mon}}(\text{Maj}_n) \geq n^{\Gamma_{\text{mon}}} / \operatorname{polylog} n$, where $\mathcal{L}_{\text{mon}}$ denotes monotone formula complexity.*

Proof. To see this, notice that our argument in the proof of Theorem 3.1 can also be carried out when $b = 1$ and $h : \{0,1\}^1 \to \{0,1\}$ is the identity function, that is $h \circ \text{Maj}_m = \text{Maj}_m$, and we apply our staged restrictions on just one block.

Let $\Gamma = \Gamma_{\text{mon}}$. Then by definition,

$$\mathop{\mathbb{E}}_{\rho \sim \mathcal{R}_p}\left[\mathcal{L}_{\text{mon}}\left(f\lceil_\rho\right)\right] = \mathcal{O}\left(1 + p^\Gamma \mathcal{L}_{\text{mon}}(f)\right)$$

Let c' be a constant such that

$$\mathop{\mathbb{E}}_{\rho \sim \mathcal{R}_p}\left[\mathcal{L}_{\text{mon}}\left(f\lceil_\rho\right)\right] \leq c' \cdot (1 + p^\Gamma \mathcal{L}_{\text{mon}}(f))$$

Let $m_1 = m = n$. As in the proof of Theorem 3.1, we set $m_{j+1} = m_j^{0.6}$ for $j \geq 1$. Let t be the last j such that $m_j \geq 32$ and $\mathcal{L}_{\text{mon}}\left(\text{Maj}_{m_{j+1}}\right) \geq 2c'$ both hold. (Recall that $b = 1$, thus $\log^5(4b) = 2^5 = 32$.)

A small calculation shows that $t \leq \frac{1}{\log(10/6)} \log\log m \leq 2\log\log m$. For $j = 1, \ldots, t$ we set $p_j = 4m_j^{-0.4} = 4m_{j+1}/m_j$.

As in the proof of Theorem 3.1, our staged restrictions ensure that $\varphi_j = \text{Maj}_{m_j}$. Similarly to our previous argument, setting $c = 2c' \cdot 16$, and using that

$$\mathcal{L}_{\text{mon}}(\varphi_{j+1}) = \mathcal{L}_{\text{mon}}\left(\text{Maj}_{m_{j+1}}\right) \geq 2c'$$

for $j = 1, \ldots, t$, we get

$$\mathcal{L}_{\text{mon}}(\varphi_{t+1}) \leq c^t \cdot \left(\frac{m_{t+1}}{m}\right)^\Gamma \cdot \mathcal{L}_{\text{mon}}(\varphi_1)$$

Thus, we obtain

$$\mathcal{L}_{\text{mon}}(\text{Maj}_n) \geq c^{-t} \cdot \left(\frac{n}{m_{t+1}}\right)^\Gamma \cdot \mathcal{L}_{\text{mon}}\left(\text{Maj}_{m_{t+1}}\right)$$

By the definition of t above, at least one of $m_{t+1} < 32$ or $\mathcal{L}_{\text{mon}}\left(\text{Maj}_{m_{t+2}}\right) < 2c'$ must hold. The latter implies that $m_{t+2} < 2c'$, hence $m_{t+1} = m_{t+2}^{10/6} < (2c')^{10/6}$ and for $c'' = \max\left\{32, (2c')^{10/6}\right\}$ we have $m_{t+1} < c''$. Since $m_t \geq 32$ we also have $m_{t+1} \geq 8$. Using $8 \leq m_{t+1} < c''$ and $t \leq 2\log\log m$ we get

$$\mathcal{L}_{\text{mon}}(\text{Maj}_n) \geq c^{-2\log\log m} \cdot \left(\frac{n}{c''}\right)^\Gamma \cdot \mathcal{L}_{\text{mon}}\left(\text{Maj}_{m_{t+1}}\right) \geq n^\Gamma / \operatorname{polylog} n$$ ◀

5 Future Directions

A possible extension of our result would be to verify the KRW conjecture [9] for composing arbitrary functions with the majority function. The KRW conjecture essentially states that the formula size of composed functions is the product of their formula sizes, e.g. $\mathcal{L}(h \circ g) \geq \Omega(\mathcal{L}(h) \cdot \mathcal{L}(g))$. The conjecture has been verified for composing arbitrary functions with parity. Unfortunately, getting asymptotically tight bounds on the formula size of majority is still open. Currently, the best upper bound on the De Morgan formula size of majority is $\mathcal{O}\big(n^{3.91}\big)$ [16]. Our lower bound would verify the conjecture for composing arbitrary functions with majority if $\mathcal{L}(\mathrm{Maj}_n) = \mathcal{O}\big(n^2\big)$.

Another interesting direction is studying the average-case hardness of the Generalized Andreev function with Majority. Here, we expect a different behavior than the standard Andreev function that is hard to compute on $1/2 + \exp(-n^{\Omega(1)})$ fraction of the inputs [11] (under the uniform distribution). For $\mathcal{M}_n$ we could not hope to get such strong average-case hardness, as we argue next. Observe that a Majority function on the $\{x_1, \ldots, x_m\}$ agrees with the dictator function of x_1 on $1/2 + \Omega(1/\sqrt{m})$ fraction of the inputs. Replacing each majority in $\mathcal{M}_n$ with the appropriate dictator yields the address function, which has formula complexity $\Theta(n)$. A small calculation shows that that a linear size formula (computing the address function) has agreement at least $1/2 + \Omega(1/\sqrt{m})^{\log n} \geq 1/2 + 2^{-\log^2(n)}$ with $\mathcal{M}_n$. We conjecture that getting a much better agreement with $\mathcal{M}_n$, say $1/2 + 1/\operatorname{poly}(n)$, or even $1/2 + 2^{-o(\log^2 n)}$, requires almost cubic formula complexity.

References

1 A. E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-scheme. *Moscow University Mathematics Bulletin*, 42(1):63–66, 1987.

2 Andrej Bogdanov. Small Bias Requires Large Formulas. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 22:1–22:12, 2018. doi:10.4230/LIPIcs.ICALP.2018.22.

3 Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining Circuit Lower Bound Proofs for Meta-Algorithms. *Computational Complexity*, 24(2):333–392, June 2015. doi:10.1007/s00037-015-0100-0.

4 Irit Dinur and Or Meir. Toward the KRW Composition Conjecture: Cubic Formula Lower Bounds via Communication Complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 3:1–3:51, 2016. doi:10.4230/LIPIcs.CCC.2016.3.

5 Johan Håstad. The Shrinkage Exponent of de Morgan Formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998. doi:10.1137/S0097539794261556.

6 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

7 Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from Shrinkage. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 111–119, 2012. doi:10.1109/FOCS.2012.78.

8 Russell Impagliazzo and Noam Nisan. The effect of random restrictions on formula size. *Random Structures & Algorithms*, 4(2):121–133, 1993.

9 Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3-4):191–204, 1995.

10 V. M. Khrapchenko. Complexity of the realization of a linear function in the class of Π-circuits. *Mathematical notes of the Academy of Sciences of the USSR*, 9(1):21–23, January 1971. `doi:10.1007/BF01405045`.

11 Ilan Komargodski and Ran Raz. Average-case lower bounds for formula size. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 171–180, 2013. `doi:10.1145/2488608.2488630`.

12 Ilan Komargodski, Ran Raz, and Avishay Tal. Improved Average-Case Lower Bounds for De Morgan Formula Size: Matching Worst-Case Lower Bound. *SIAM Journal on Computing*, 46(1):37–57, 2017. `doi:10.1137/15M1048045`.

13 Michael S Paterson and Uri Zwick. Shrinkage of de Morgan formulae under restriction. *Random Structures & Algorithms*, 4(2):135–150, 1993.

14 P. Pudlak. Personal communication, 2018.

15 John Riordan and Claude E Shannon. The Number of Two-Terminal Series-Parallel Networks. *Studies in Applied Mathematics*, 21(1-4):83–93, 1942.

16 I. S. Sergeev. Complexity and depth of formulas for symmetric Boolean functions. *Moscow University Mathematics Bulletin*, 71(3):127–130, 2016.

17 Bella Abramovna Subbotovskaya. Realizations of linear functions by formulas using $+$, $*$, and $-$. *Doklady Akademii Nauk SSSR*, 136(3):553–555, 1961.

18 Avishay Tal. Shrinkage of De Morgan Formulae by Spectral Techniques. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 551–560, 2014. `doi:10.1109/FOCS.2014.65`.

19 Avishay Tal. Formula lower bounds via the quantum method. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1256–1268, 2017. `doi:10.1145/3055399.3055472`.

20 Leslie G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3):363–366, 1984.

21 I. Wegener. The critical complexity of all (monotone) Boolean functions and monotone graph properties. *Information and Control*, 67:212–222, 1985.