Depth-reduction for composites

Shiteng Chen
* State Key Laboratory of Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing, China
Email: shitengchen@gmail.com

Periklis A. Papakonstantinou
†MSIS, RBS
Rutgers University
Piscataway, NJ, USA
Email: periklis.research@gmail.com

***Abstract*—We obtain a new depth-reduction construction, which implies a super-exponential improvement in the depth lower bound separating NEXP from non-uniform ACC.**

In particular, we show that every circuit with $\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$, and MOD_m gates, $m \in \mathbb{Z}^+$, of polynomial size and depth d can be reduced to a depth-2, $\mathrm{SYM} \circ \mathrm{AND}$, circuit of size $2^{(\log n)^{O(d)}}$. This is an exponential size improvement over the traditional Yao-Beigel-Tarui, which has size blowup $2^{(\log n)^{2^{O(d)}}}$. Therefore, depth-reduction for composite m matches the size of the Allender-Hertrampf construction for primes from 1989.

One immediate implication of depth reduction is an improvement of the depth from $o(\log\log n)$ to $o(\log n/\log\log n)$, in Williams' program for ACC circuit lower bounds against NEXP. This is just short of $O(\log n/\log\log n)$ and thus pushes William's program to the NC^1 barrier, since NC^1 is contained in ACC of depth $O(\log n/\log\log n)$. A second, but non-immediate, implication regards the strengthening of the ACC lower bound in the Chattopadhyay-Santhanam interactive compression setting.

Keywords: composite modulus, depth-reduction, circuit lower bound, Williams' program, interactive compression

I. Introduction

The development of computational complexity is vastly a history of conjectures, and gaps between these conjectures and what is actually proved. One such story regards the power of MOD_m gates in small-depth boolean circuits that also have $\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$ gates. A MOD_m gate outputs 1 if and only if the number of 1s in its input is a multiple of m. What is known for prime $m = p$ stands in sharp contrast to what is known for composite $m \in \mathbb{Z}^+$.

In a sense we settle the question about depth-reduction of ACC circuits.[1] Depth-reduction is an algorithm compressing a low-depth ACC circuit (highly parallel algorithm) of depth d into a depth-2 circuit (extremely parallel algorithm). Theorem 1 below states that for every ACC_m circuit, where m is composite, there is an equivalent depth-2 circuit of size $2^{(\log n)^{O(d)}}$. This size asymptotically matches the construction of Allender-Hertrampf [1] for prime moduli and improves exponentially the $2^{(\log n)^{2^{O(d)}}}$ size of the previously best-known Yao-Beigel-Tarui construction [2], [3].

Theorem 1 (formally stated on p. 8)**.** *There is an efficient algorithm that given a circuit with* AND*,* OR*,* NOT*,* MOD_m *gates, of depth d, input length n, and size $s \geq n$, the algorithm outputs a depth-2 circuit* $\mathrm{SYM} \circ \mathrm{AND}$ *of size $2^{(\log s)^{O(d)}}$, where* SYM *is a gate whose output depends only on the number of 1s in its input.*

Depth-reduction constructions are sensitive to the types of gates of the circuits. For instance, when we only consider circuits with only $\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$ gates then it is impossible to compress the depth even from depth $k+1$ to k, without suffering an exponential size blowup. This was proved in the worst case by Hastad and Yao [4], [5]. In a recent breakthrough by Rossman, Servedio, and Tan [6] it was shown that this irreducibility holds also on the average.

Depth-reduction is fundamental and also related to other fundamental questions in circuit complexity. We will explain the most relevant connections to depth-

[1]In the literature ACC^0 denotes the class of boolean functions computable by polynomial size $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, \mathrm{MOD}_m\}$ families of circuits of constant depth and $m \in \mathbb{Z}^+$. We will be referring to ACC^0 both as the class of boolean functions and the circuits characterizing it. Since, we consider circuits of different depth d and size s we will commonly refer to such circuits as ACC circuits (or ACC_m for a fixed modulus) of depth d and size s.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.20

reduction, after we first briefly recall what is already known for prime moduli $m = p$.

The celebrated works of Razborov and Smolensky [7], [8] showed that the boolean function MOD_q cannot be computed by ACC_p^0 circuits for primes $p \neq q$. This was technically achieved by viewing an ACC_p circuit as a polynomial over $\mathbb{F}_p$. This view appeared to be very fruitful and in particular in the depth-reduction of ACC_p circuits [1]. Thus, for prime modulus p (i) strong lower bounds and (ii) depth-reduction algorithms are known since the early 90s.

Smolensky conjectured [8] that the lower bound extends to MOD_r and ACC_m^0 for *every composite* coprime moduli m, r. This conjecture is still a holy grail for contemporary circuit complexity. Since then, there is a spate of important works, e.g. [9], [10], [11], [12], [13], [14], [15], [16], [17], that obtain lower bounds for restricted forms of depth-2 or depth-3 circuits. These works introduced a number of analytic techniques, at the same time shaping our understanding and goals of modern circuit complexity.

From depth-reduction to ACC *circuit lower bounds:* Smolensky's conjecture relates to our depth-reduction as follows. Note that *no* non-trivial limitations for general ACC circuits were known up until [18], which showed that non-uniform $\mathrm{ACC}^0 := \bigcup_m \mathrm{ACC}_m^0$ does not contain NEXP. Importantly, Williams [18], [19] introduced a program according to which an improved depth-reduction algorithm yields [18], [19] a lower bound for NEXP against circuits of higher depth – the smaller the size blowup in the depth-reduction the bigger the depth in the ACC lower bound.

Here are a few more details. In his seminal work, Williams first gives a slightly better-than-brute-force Circuit-SAT algorithm for ACC-circuits. Then, he shows that if $\mathrm{NEXP} \subseteq \mathrm{ACC}^0$, the depth-reduction algorithm can be used to imply for every problem in $\mathrm{NTIME}(2^n)$ a nondeterministic algorithm that runs in time $o(\frac{2^n}{n})$. This contradicts Cook's nondeterministic time-hierarchy [20], and thus $\mathrm{NEXP} \not\subseteq \mathrm{ACC}^0$. More generally, the existence of a "slightly better-than-brute-force" algorithm for $\mathfrak{C}$-SAT implies $\mathrm{NEXP} \not\subseteq \mathfrak{C}$; see Section IV-A for some restrictions on $\mathfrak{C}$. A crucial step of the circuit-SAT algorithms in [18], [19] is that the depth-reduced circuit can be of any up to slightly sub-exponential size. Therefore, the triple-exponential size blowup in the depth [3] implies [21] an NEXP lower bound, i.e. $\mathrm{NEXP} \not\subseteq \mathrm{ACC}(2^{\log^k n}, o(\log\log n))$ for every constant $k > 0$. Theorem 1 improves super-exponentially[2] the previously best-known $o(\log\log n)$ depth for ACC circuits to $o(\log n/\log\log n)$. More details are explained in Section IV-A.

A strengthening of the above circuit lower bound is given in the interactive compression setting of Chattopadhyay-Oliveira-Santhanam [22], [23]. This setting is interesting in its own right. In Section IV-B we present this result together with intuition, motivation, and comparison to previous work.

II. Notation and existing tools

We assume familiarity with the terminology in basic computational complexity, cf. [24]. All circuit classes in this paper are non-uniform. We denote by ACC_m^0 the class of boolean functions of the form $\{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{Z}^+}$ computable by families of circuits $\{C_n\}_{n\in\mathbb{Z}^+}$ where each C_n is of polynomial size $\mathrm{poly}(n)$, constant depth, and uses gates $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, \mathrm{MOD}_m\}$, where MOD_m is a boolean gate defined below. We measure *size* as the number of wires in the circuit, *depth* as the length of longest path from the output of the circuit to any input. Let also, $\mathrm{ACC}^0 := \cup_{m\in\mathbb{Z}^+}\mathrm{ACC}_m^0$. We denote by $\mathrm{ACC}_m(s,d)$ the class of boolean functions characterized by families of $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, \mathrm{MOD}_m\}$-circuits of size s and depth d. Let also $\mathrm{ACC}(s,d) := \cup_{m\in\mathbb{Z}^+}\mathrm{ACC}_m(s,d)$. In this notation, $\mathrm{ACC}^0 = \mathrm{ACC}\big(n^{O(1)}, O(1)\big)$.

We write ACC^0 *circuit* for a family of circuits characterizing a function in ACC^0, whereas ACC_m *circuit* designates a circuit family with $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, \mathrm{MOD}_m\}$ gates.

Families of *layered circuits* are denoted in the usual way. That is, $\mathrm{SYM} \circ \mathrm{AND} \circ \mathrm{MOD}_m$ denotes a family of depth-3 circuits (or one member of the family) where the output gate is a symmetric gate. A *symmetric gate* SYM is a boolean function whose output depends on the number of 1s in the input; e.g. the "MOD gate" (see below), "majority gate", "threshold gate". The maximum fan-in of a gate at a layer is written in brackets as a subscript, e.g. $\mathrm{MOD}_m \circ \mathrm{AND}_{[\delta_{\mathrm{AND}}]}$ the AND gates at the bottom (next to the input) layer have fan-in at most δ_{AND}.

We write $||x||_1 := \sum_{i=1}^n x_i$, treating x_i's as integers, for $x \in \{0,1\}^n$ and denote by MOD_m the boolean function (gate) that takes an N-bit input $x = (x_1, \ldots, x_N)$ and $\mathrm{MOD}_m(x) = 1 \iff m \big| ||x||_1$. For MOD_m and

[2] From $o(\log\log n)$ to $o(\log n/\log\log n)$ the increase is super-exponential, whereas from $O(\log\log n)$ to $o(\log n/\log\log n)$ sub-exponential as correctly pointed out by Oded Goldreich.

every other symmetric gate we will assume that take as input $||x||_1$, i.e. we write $\mathrm{MOD}_m(||x||_1)$.

The $\mathrm{MOD}_m(||x||_1)$, which is evaluated to $\{0,1\}$, should not be confused with the modulus over $\mathbb{Z}$, i.e. $||x||_1(\mod m)$. We restrict to a prime field $\mathbb{F}_q$ or ring $\mathbb{Z}_m$ using "$\mod q$" or "$\mod m$". This reduces notational clutter – distinct fields and rings, in a sense, coexist in the same circuit and our techniques simultaneously use and relate more than one.

All operations in this paper are over $\mathbb{C}$. For example, in evaluating a polynomial function $P : \{0,1\}^n \to \mathbb{Z}$ with integer coefficients the operations treat the inputs $0, 1$ as integers. Polynomial functions always take inputs $\{0,1\}^n$ and recall that MOD_m gates take inputs from $\mathbb{Z}$.

For $X \in \mathbb{Z}$ we write $e_m(X) := e^{X\frac{2\pi i}{m}}$, where $e^{\frac{2\pi i}{m}}$ is the m-th primitive root of 1. Then, observe that $\mathrm{MOD}_m(X) = \frac{1}{m}\sum_{0\le k<m} e_m(kX)$.

Preprocessing and Mod-Amplifiers: For depth-reduction and its applications we consider *explicit circuit constructions*, i.e. constructions computable in time polynomial (in fact, AC^0) in the size of the output circuit. Explicitness will be used in the applications of depth-reduction, including the extension of [18].

Our construction in Section III uses a preprocessing step from [3]. This is how we deal with big fan-in AND gates and initially replace MOD_m gates, where m is composite, by modular gates of prime modulus. Lemma 2 does this preprocessing efficiently.

Lemma 2 ([3], [25], [19]). *There is an explicit construction that for every number of input bits n and modulus $2 \le m \le \log^{O(1)} n$, given an ACC_m circuit of depth d and size s, where there are s_{AND} many AND gates each of fan-in at most δ_{AND}, the construction outputs a $\mathrm{SYM} \circ \mathrm{ACC}$ circuit with the following properties.*

- *i. The depth of the circuit is $2\Delta(m)d$, where $\Delta(m)$ is the number of distinct prime divisors of m[3].*
- *ii. The size of the circuit is $s \cdot 2^{O(m \log s_{\mathrm{AND}} \cdot m^2 \log^2 \delta_{\mathrm{AND}})} = 2^{O\left((m \log s)^3\right)}$.*
- *iii. The fan-in of every AND gate in the circuit is $O(m \log s_{\mathrm{AND}} \cdot m \log \delta_{\mathrm{AND}}) = O\left((m \log s)^2\right)$.*
- *iv. Each MOD gate of the circuit is a MOD_q gate, where q is a prime divisor of m (in general, many types of MOD_q's are inside the same circuit).*
- *v. The circuit is layered, i.e. each layer contains gates of the same type.*

More presisely when we furthermore consider an ACC_m circuit of size $2^{\log^k n}$. Then, the size of the constructed circuit is at most $2^{(m \log^k n)^3}$, the AND gate fan-in is at most $2^{(m \log^k n)^2}$, and the depth is at most $2\Delta(m)d$.

The above hold true if instead of an ACC_m circuit we are given an $\mathrm{SYM} \circ \mathrm{ACC}_m$ circuit.

Remark 3. *The algorithm in the proof of Lemma 2 is doing 3 things: (i) reduces the fan-in of AND gates to at most $\log s_{\mathrm{AND}} \cdot \log \delta_{\mathrm{AND}}$; (ii) decomposes the MOD_m gates into circuits with MOD_p gates one for each p, a prime divisor of m; (iii) layers the circuit, i.e. each layer only contains the same type of gates.*

To reduce AND gate fan-in we replace each AND gate of fan-in $\le \delta_{\mathrm{AND}}$ by a probabilistic $\mathrm{MOD}_p \circ \mathrm{AND}$ circuit, where the AND gates fan-in is at most $O(\log s_{\mathrm{AND}} \cdot \log \delta_{\mathrm{AND}})$, where all these probabilistic sub-circuits are sampling from a $2^{O(\log s_{\mathrm{AND}} \cdot \log^2 \delta_{\mathrm{AND}})}$ size sample space [26]. Then, we [26] derandomize through enumeration and majority vote, which can be implemented with $2^{O(\log s_{\mathrm{AND}} \cdot \log^2 \delta_{\mathrm{AND}})}$ copies of sub-circuits. This step only replaces the AND gates. Therefore, the same algorithm can be used in circuits with different types of gates, changing only the ANDs and leaving the rest intact. This property will be used in the interactive compression bounds in Subsection IV-B.

Note that the constant $2\Delta(m)$ in the depth is a universal constant and the same holds for the constants in the exponents of size and AND fan-in.

After the preprocessing of Lemma 2 we get a circuit with different kinds of MOD gates. Therefore, *a priori*, it is not clear how to express the circuit as one polynomial – expressing the circuit as a polynomial is how depth-reduction is typically done. To collapse different MOD gates we use Mod-Amplifiers to increase moduli. These Mod-Amplifiers are simply a special family of high degree polynomials, originally introduced by Toda [27] for proving $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$.

Lemma 4 (Mod-Amplifiers [3], weaker forms in [27], [2]). *For any integer k, there exists a degree $2k$ polynomial MP_k with integer coefficients such that for any integer $m > 1$, and any integer X, $\mathrm{MP}_k(X) = 0 \mod m^k$ if $X = 0 \mod m$; and $\mathrm{MP}_k(X) = 1 \mod m^k$ if $X = 1 \mod m$.*

Thus, Mod-Amplifiers amplify the modulus without changing the 0/1 value of the mod-function.

III. The depth-reduction

We now present the depth-reduction construction and prove Theorem 1. Theorem 1 is formally restated at the end of this section. The same proof presented here, is used to obtain a stronger form of Theorem 1, which

[3]We write $\Delta(m)$ instead of the typical $\omega(m)$ notation.

we need in the interactive compression setting of Section IV-B.

The depth-reduction is presented in three parts: (i) the linearization lemma (Lemma 5), (ii) a single step of our iterative depth-reduction construction (Lemma 8), and (iii) the use of Mod-Amplifiers (Theorem 1).

A. Linearization: eliminating products

Lemma 5 is an important technical tool, which might be also of independent interest. It shows that the AND-layer can be eliminated in a $\mathrm{MOD}_m \circ \mathrm{AND} \circ \mathrm{MOD}_r$ configuration, for m, r co-prime, i.e. $\gcd(m, r) = 1$. Lemma 5 *relies on the power of composite arithmetic*, since a $(\mod m)$ is added even if it were not there originally.[4] When we later use Lemma 5 we will see that although this construction initially blows up the size, at the end there is a huge payback (to the initial size-worsening in each application of the construction). Thus, we get an exponentially smaller construction compared to [2], [3].

Lemma 5 (Linearization lemma)**.** *Given positive integers $m, r \in \mathbb{Z}^+$, $\gcd(m, r) = 1$ and k indeterminates (variables) $L_1, \dots, L_k$, there exist r^{k+1} integral linear combinations $L'_1, \dots, L'_{r^{k+1}}$, i.e. $L'_i := \ell_i(L_1, \dots, L_k)$ for linear form ℓ_i, and integers $c_1, \dots, c_{r^{k+1}} \in \{0, 1, 2, \dots, m-1\}$ such that for all valuations of the L_i in $\mathbb{Z}^+$ we have the identity*

$$\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i) = \sum_{1 \le i \le r^{k+1}} c_i \mathrm{MOD}_r(L'_i) \mod m$$

The linear combinations L'_i and coefficients c_i can be computed in time $r^{O(k)}$ (when each arithmetic operation with the L_i's costs one time step).

When we apply Lemma 5 the MOD_r's take inputs from the previous layer; say that these outputs of the gates of the previous layer bits are the binary vector $y \in \{0, 1\}^N$. Since each L_i is the hamming weight of the input bits then both L_i and L'_i are integral linear combinations of the y_i's.

We stress out that integrality in the linear combinations and coefficients is necessary for using this construction in transforming circuits. If one merely cares to write the product of MOD as a sum then this is easy over complex $\mathbb{C}$ coefficients (see Remark 6 inside the following proof).

[4]In particular, even if we use our method instead of Allender-Hertrampf [1] for $\mathrm{ACC}_{\mathrm{prime}}$ circuits we still have to introduce a second type of MOD gates (two types of MODs is the same as one composite).

Proof of Lemma 5: The construction of the L_i's and its analysis is shown in four parts.

Represent $\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i)$ as an exponential sum:

$$\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i) = \prod_{1 \le i \le k} \left(\frac{1}{r} \sum_{0 \le j < r} e_r(j \cdot L_i) \right)$$
$$= \frac{1}{r^k} \sum_{(j_1, \dots, j_k) \in \mathbb{Z}_r^k} e_r\left(\sum_{1 \le i \le k} (j_i L_i) \right)$$

Remark 6. *We can write $\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i)$ as a sum with complex coefficients by observing that $\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i) = \sum_{1 \le i \le s} c_i \mathrm{MOD}_r(L'_i(x))$, $c_i \in \mathbb{C}$, since for every $Y \in \mathbb{Z}^+$, $e_r(Y) = \sum_{0 \le i < r} e_r(i) \mathrm{MOD}_r(Y - i)$. However, the statement of this lemma is about integral coefficients and linear combinations. To that end, we introduce a co-prime modulus m that enables us to compute ring inverses.*

$$r^k \prod_{1 \le i \le k} \mathrm{MOD}_r(L_i) = \sum_{(j_1, \dots, j_k) \in \mathbb{Z}_r^k} e_r\left(\sum_{1 \le i \le k} (j_i L_i) \right)$$

Since $\gcd(m, r) = 1$ there exists an inverse $(r^k)^{-1}$ of r^k in the ring $\mathbb{Z}_m$. Then, $\prod_{1 \le i \le k} \mathrm{MOD}_r(L_i)$

$$= (r^k)^{-1} \sum_{(j_1, \dots, j_k) \in \mathbb{Z}_r^k} e_r\Big(\sum_{1 \le i \le k} (j_i L_i) \Big) \mod m \quad (1)$$

Introduce a group action that partitions $\mathbb{Z}_r^k$ into well-behaved orbits:

For every $u \in \mathbb{Z}_r$ and $v = (v_1, v_2, \dots, v_k) \in \mathbb{Z}_r^k$, define $u \cdot v = (uv_1, uv_2, \dots, uv_k)$, where the operation uv_i is in $\mathbb{Z}_r$.[5] We define the binary relation $\equiv$ on $\mathbb{Z}_r^k$ such that for any $x, y \in \mathbb{Z}_r^k$, $x \equiv y$ if and only if $y \in \mathbb{Z}_r^* \cdot x$, where $\mathbb{Z}_r^*$ stands for the multiplicative group of integers modulo r. This is an equivalence relation on $\mathbb{Z}_r^k$, since $\mathbb{Z}_r^*$ is a group under multiplication. Then, $\equiv$ partitions $\mathbb{Z}_r^k$ into many[6] equivalence classes. These are also called the *orbits* of the group action. Let us denote each of the equivalence classes by $S_l = \mathbb{Z}_r^* \cdot (a_{l,1}, \dots, a_{l,k})$. Regarding explicitness, in our construction each S_l can be computed by enumeration in time $r^{O(k)}$.

[5]**Intuition:** The partitioning of interest are the orbits of this group action, which are just "lines". The benefit in restricting the summation inside each such "line" is that when MOD is written using an exponential sum, then itself becomes a sum of primitive roots over a scaled "line".

[6]These are less than r^k. The exact number can be computed by Burnside's Lemma; cf. [28].

Then,

$$\sum_{(j_1,\ldots,j_k)\in\mathbb{Z}_r^k} e_r(\sum_{1\le i\le k}(j_iL_i))$$
$$=\sum_l \sum_{(j_1,\ldots,j_k)\in S_l} e_r(\sum_{1\le i\le k}(j_iL_i))$$

Sum inside each orbit:

The following is a very important property regarding how the exponential sums behave inside each equivalence class (i.e. inside each orbit of our group action).

Fix an arbitrary equivalence class $S_l = \mathbb{Z}_r^* \cdot (a_{l,1}, a_{l,2}, \ldots, a_{l,k})$:

Let $\gcd(a_{l,1}, a_{l,2}, \ldots, a_{l,k}, r) = c$.

Let $a'_{l,i} = a_{l,i}/c$, $r' = r/c$ and thus $\gcd(a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k}, r') = 1$. Hence,

$$\begin{aligned} S_l =& \mathbb{Z}_r^* \cdot (a_{l,1}, a_{l,2}, \ldots, a_{l,k}) \\ =& \mathbb{Z}_r^* \cdot c(a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k}) \\ =& (c\mathbb{Z}_r^*) \cdot (a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k}) \end{aligned}$$

where $c\mathbb{Z}_r^* = c\{t \mid \gcd(t,r) = 1\} = \{t \mid \gcd(t,r) = c\}$. Since $\gcd(a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k}, r') = 1$, for any $x, y \in c\mathbb{Z}_r^*$, $x \cdot (a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k}) = y \cdot (a'_{l,1}, a'_{l,2}, \ldots, a'_{l,k})$ if and only if $x = y$.

$$\sum_{(j_1,\ldots,j_k)\in S_l} e_r(\sum_{1\le i\le k}(j_iL_i))$$
$$=\sum_{\gcd(t,r)=c,\ 0\le t<r} e_r(\sum_{1\le i\le k} t\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{\gcd(t',r')=1,\ 0\le t'<r'} e_r(\sum_{1\le i\le k} t'c\cdot a'_{l,i}\cdot L_i)$$
$$(t' = t/c)$$

This sum is over $\{\gcd(t',r') = 1,\ 0 \le t' < r'\}$ and thus it can be computed by inclusion-exclusion. We can first sum all of the terms corresponding to $0 \le t' < r'$ together. Then, subtract the sums of the terms corresponding to the t's divisible by a prime divisor p of r'. Then, add the terms corresponding to t's divisible by two distinct prime divisor p_i and p_j of r', and so on. This inclusion-exclusion calculation is greatly simplified using the Mobius function.

Mobius function is defined $\mu : \mathbb{Z} \to \{-1, 0, 1\}$ as follows.

i. $\mu(x) = 0$, if there exists prime q such that $q^2|x$.
ii. $\mu(x) = (-1)^{r^{k+1}}$, if there is no square-of-a-prime diving x. Thus, $x = \prod_{1\le i\le r^{k+1}} q_i$, where q_i are the r^{k+1}-many distinct prime divisors of x.

One observes that $\sum_{d|n}\mu(d) = 1$ if $n = 1$ and $\sum_{d|n}\mu(d) = 0$ otherwise.

Using these properties we bound the exponential sum inside the fixed S_l.

$$\sum_{(j_1,\ldots,j_k)\in S_l} e_r(\sum_{1\le i\le k}(j_iL_i))$$
$$=\sum_{\gcd(t,r)=c,\ 0\le t<r} e_r(\sum_{1\le i\le k} t\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{\gcd(t',r')=1,\ 0\le t'<r'} e_r(\sum_{1\le i\le k} t'c\cdot a'_{l,i}\cdot L_i)\ (t' = t/c)$$
$$=\sum_{\gcd(t',r')=1,\ 0\le t'<r'} e_{r'}(\sum_{1\le i\le k} t'\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{0\le t'<r'}\sum_{d|\gcd(t',r')}\mu(d)e_{r'}(\sum_{1\le i\le k} t'\cdot a'_{l,i}\cdot L_i)$$
$$(\sum_{d|\gcd(t',r')}\mu(d) = 1 \text{ if } \gcd(t',r') = 1 \text{ and } 0 \text{ otherwise})$$
$$=\sum_{0\le t'<r'}\sum_{d|t',d|r'}\mu(d)e_{r'}(\sum_{1\le i\le k} t'\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{d|r'}\mu(d)\sum_{d|t',0\le t'<r'} e_{r'}(\sum_{1\le i\le k} t'\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{d|r'}\mu(d)\sum_{0\le t''<r'/d} e_{r'}(\sum_{1\le i\le k} t''d\cdot a'_{l,i}\cdot L_i)\ (t'' = t'/d)$$
$$=\sum_{d|r'}\mu(d)\sum_{0\le t''<r'/d} e_{r'/d}(\sum_{1\le i\le k} t''\cdot a'_{l,i}\cdot L_i)$$
$$=\sum_{d|r'}\mu(d)\cdot\frac{r'}{d}\mathrm{MOD}_{\frac{r'}{d}}(\sum_{1\le i\le k} a'_{l,i}\cdot L_i)$$
$$=\sum_{d|r'}\mu(d)\cdot\frac{r'}{d}\mathrm{MOD}_r(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i)$$
$$=\sum_{d|\frac{r}{\gcd(a_{l,1},a_{l,2},\ldots,a_{l,k},r)}}\frac{\mu(d)r}{d\cdot\gcd(a_{l,1},a_{l,2},\ldots,a_{l,k},r)}$$
$$\mathrm{MOD}_r(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i)$$

By letting $\kappa_{S_l,r} := \frac{r}{\gcd(a_{l,1},a_{l,2},\ldots,a_{l,k},r)}$ we have

$$\sum_{(j_1,\ldots,j_k)\in S_l} e_r(\sum_{1\le i\le k}(j_iL_i)) = \sum_{d|\kappa_{S_l,r}}\kappa_{S_l,r}\frac{\mu(d)}{d}\mathrm{MOD}_r(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i) \tag{2}$$

Put (1) and (2) together:

$$\prod_{1\le i\le k}\mathrm{MOD}_r(L_i)$$

$$= (r^k)^{-1} \sum_{(j_1,\ldots,j_k)\in\mathbb{Z}_r^k} e_r\Big(\sum_{1\le i\le k} (j_i L_i)\Big) \mod m$$

$$= (r^k)^{-1} \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\ldots,a_{l,k})} \sum_{(j_1,\ldots,j_k)\in S_l} e_r\Big(\sum_{1\le i\le k} j_i L_i\Big) \mod m$$

$$= (r^k)^{-1} \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\ldots,a_{l,k})} \left(\sum_{d\mid\kappa_{S_l,r}} \kappa_{S_l,r}\frac{\mu(d)}{d} \mathrm{MOD}_r\Big(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i\Big)\right) \mod m$$

$$= \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\ldots,a_{l,k})} \left(\sum_{d\mid\kappa_{S_l,r}} \alpha\, \mathrm{MOD}_r\Big(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i\Big)\right) \mod m$$

where $\alpha = \underbrace{\left(\kappa_{S_l,r}\frac{(r^k)^{-1}\mu(d)}{d} \mod m\right)}_{\text{integer}}$. ■

Remark 7 (Aside remark)**.** *Here are two aside (not used later in this paper) remarks.*
(i) The AND *gate with fan-in* k *in the LHS,* $\mathrm{AND}_{[k]} \circ \mathrm{MOD}_r$ *can be replaced by* $\mathrm{ANY}_{[k]}$ *boolean function of fan-in* k*. Recall that every function can be written as a polynomial with* $2^{O(k)}$ *terms and thus we can obtain the* Generalized Linearization Lemma*.*
(ii) In depth-reduction we use Lemma 5 for $r = p$*, for prime* p*. The Generalized Linearization Lemma (and for general* m*) is of independent interest. For instance, an immediate consequence is that an exponential lower bound for* $\mathrm{MOD}_6 \circ \mathrm{MOD}_{35} \implies$ *exponential lower bound for* $\mathrm{MOD}_6 \circ \mathrm{ANY}_{[o(n)]} \circ \mathrm{MOD}_{35}$*.*[7]

B. Inside a single iteration: using linearization & mod-amplification

Now, we show how to use the construction of Lemma 5 and the preprocessing Lemma 4, to perform a single step (described in Lemma 8) of an iterative construction (described in Lemma 1). Note that N denotes the number of input bits to a layer and n the circuit input length.

Lemma 8 is critically different from the previous depth-reduction technology. Beigel-Tarui replaces each MOD_q gate by a Mod-Amplifier. The Mod-Amplifiers are quite high degree polynomials. Thus, the AND gates, i.e. products of Mod-Amplifiers, blow up very fast the degree and size [3], [27], [2]. Instead, we first use Lemma 5 to remove the AND layer. Although, this causes an even further increase in size later on we have huge overall gains.

Lemma 8. *For every* $\mathrm{SYM}_{[\delta_{\mathrm{SYM}}]} \circ \mathrm{AND}_{[\delta_{\mathrm{AND}}]} \circ \mathrm{MOD}_q$ *circuit on* N *input bits* $X = (X_1, X_2, \ldots, X_N)$*, where* q *is a prime number and* $N > q$*, there is an explicit construction of a* $\mathrm{SYM}_{[N^{2q(\delta_{\mathrm{AND}}+2\log\delta_{\mathrm{SYM}})}]} \circ \mathrm{AND}_{[2(q-1)(\delta_{\mathrm{AND}}+2\log\delta_{\mathrm{SYM}})]}$ *circuit, which computes the same function as the given circuit.*

Proof: Since the output of a symmetric gate is only a function of the hamming weight of the input, we will assume the given circuit is $f\left(\sum_{1\le i\le\delta_{\mathrm{SYM}}} \prod_{1\le j\le\delta_{\mathrm{AND}}} \mathrm{MOD}_q(l_{i,j}(X))\right)$. Here, the function $f : \{0,1,\ldots,\delta_{\mathrm{SYM}}\} \to \{0,1\}$ corresponds to the SYM gate of the top layer; $\prod_{1\le j\le\delta_{\mathrm{AND}}}$ corresponds to the next AND layer; $\mathrm{MOD}_q(l_{i,j})$ corresponds to the third MOD_q layer, where $l_{i,j}$ are integral linear functions on X, i.e. from $\{0,1\}^N$ to $\mathbb{Z}$ (equivalently, $l_{i,j}(X)$ is the inner product of X with an integral vector).

The "steps" below correspond to the steps of the algorithm realizing the construction.

Step 1: Remove the AND gates using Lemma 5.

To apply Lemma 5 we take the $\mod m$ of the output of the $\mathrm{AND} \circ \mathrm{MOD}_q$ circuit. Thus, we first modify the given symmetric function by adding a mod-layer and keeping the value unchanged.

Pick the smallest integer s' such that $s' > \delta_{\mathrm{SYM}}$ and $(s', q) = 1$. Then,

$$f\left(\sum_{1\le i\le\delta_{\mathrm{SYM}}} \prod_{1\le j\le\delta_{\mathrm{AND}}} \mathrm{MOD}_q(l_{i,j}(X))\right)$$

$$= f\left(\left(\sum_{1\le i\le\delta_{\mathrm{SYM}}} \prod_{1\le j\le\delta_{\mathrm{AND}}} \mathrm{MOD}_q(l_{i,j}(X))\right) \mod s'\right)$$

Then, by Lemma 5

$$\sum_{1\le i\le\delta_{\mathrm{SYM}}} \prod_{1\le j\le\delta_{\mathrm{AND}}} \mathrm{MOD}_q(l_{i,j}(X)) \mod s'$$

[7]The generalization of Lemma 5 was suggested to us by Ryan Williams (personal communication). Ryan Williams (personal communication) indicated that for prime r and in particular for ACC_6 linearization can be made to work with Fourier analytic techniques, whereas Kristoffer Hansen (personal communication) indicated that the same might be possible for every composite ACC_m. Regarding composite r, Richard Beigel (personal communication) came up recently with a beautiful, simplified inductive proof of our linearization – it achieves almost the same result as in our statement (but for a slightly worse constant than in our statement).

$$= \sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MOD}_q(l'_{i,j}(X)) \mod s'$$

where $c_{i,j}$ are integer coefficients between 0 and s', and l' are linear combinations of l. Then,

$$f\left(\sum_{1\le i\le \delta_{\text{SYM}}} \prod_{1\le j\le \delta_{\text{AND}}} \text{MOD}_q(l_{i,j}(X))\right) = f\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MOD}_q(l'_{i,j}(X)) \mod s'\right)$$

Define a symmetric f' as $f'(Y) = f(Y \mod s')$ and thus

$$f\left(\sum_{1\le i\le \delta_{\text{SYM}}} \prod_{1\le j\le \delta_{\text{AND}}} \text{MOD}_q(l_{i,j}(X)) \mod s'\right)$$
$$= f'\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MOD}_q(l'_{i,j}(X))\right)$$

Step 2: Use Mod-Amplifiers to remove the MOD_q layer.

By Fermat's little theorem, $\text{MOD}_q(l(X)) = (1 - l(X)^{q-1}) \mod q$. Thus, we can replace each MOD_q gate by a low degree polynomial over $\mathbb{F}_q$. Then, we "link" these polynomials on $\mathbb{F}_q$ with the symmetric gate on top by amplifying the moduli through Lemma 4. Choose integer $k = \left\lceil \log\left(\delta_{\text{SYM}} \cdot s' \cdot \frac{q^{\delta_{\text{AND}}}-1}{q-1}\right) / \log q \right\rceil \le (\delta_{\text{AND}} + 2\log \delta_{\text{SYM}})$. Then, $q^k > \sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}$. Then,

$$f'\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MOD}_q(l'_{i,j}(X))\right)$$
$$= f'\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}((1 - (l'_{i,j}(X))^{q-1}) \mod q)\right)$$
$$\text{(by Fermat's little theorem)}$$
$$= f'\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}(\text{MP}_k(1 - (l'_{i,j}(X))^{q-1}) \mod q^k)\right)$$
$$\text{(using Mod-Amplifiers)}$$
$$= f'\left(\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}(\text{MP}_k(1 - (l'_{i,j}(X))^{q-1}) \mod q^k)\right) \mod q^k\right)$$
$$\text{(since } q^k > \sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j})$$
$$= f'\left(\left(\sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MP}_k(1 - (l'_{i,j}(X))^{q-1})\right) \mod q^k\right)$$

Let us denote by $P(X) = \sum_{1\le i\le \delta_{\text{SYM}}} \sum_{1\le j\le \frac{q^{\delta_{\text{AND}}}-1}{q-1}} c_{i,j}\text{MP}_k(1 - (l'_{i,j}(X))^{q-1})$. Then, the original circuit becomes $f'(P(X) \mod q^k)$, $\deg(P) \le \deg(\text{MP}_k) \cdot (q-1) \le 2k(q-1) \le 2(q-1)(\delta_{\text{AND}} + 2\log \delta_{\text{SYM}})$.

Step 3: Represent the formula as a SYM ∘ AND circuit.

It is easy to see that P is a polynomial with integer coefficients. Since $\deg(P) \le 2(q-1)(\delta_{\text{AND}} + 2\log \delta_{\text{SYM}})$, we will assume $P(X) = \sum_{A\subseteq\{1,2,\dots,N\}, |A|\le 2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})} b_A \prod_{i\in A} X_i$, where the coefficients b_A are all integers. Let the integers b'_A be the $\mod q^k$ remainders of b_A, and thus $0 \le b'_A < q^k$. Then,

$$f'(P(x) \mod q^k)$$
$$= f'\left(\sum_{\substack{A\subseteq\{1,2,\dots,N\}\\ |A|\le 2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})}} b_A \prod_{i\in A} X_i \mod q^k\right)$$
$$= f'\left(\sum_{\substack{A\subseteq\{1,2,\dots,N\}\\ |A|\le 2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})}} b'_A \prod_{i\in A} X_i \mod q^k\right)$$
$$= f'\left(\sum_{\substack{A\subseteq\{1,2,\dots,N\}\\ |A|\le 2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})}} \sum_{1\le j\le b'_A} \prod_{i\in A} X_i \mod q^k\right)$$

Then, the original function can be represented as a circuit whose top layer is a symmetric gate $f'((\sum_{A\subseteq\{1,2,\dots,N\},|A|\le 2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})} \sum_{1\le j\le b'_A} Y_{A,j}) \mod q^k)$ and the next AND layer is $\prod_{i\in A} X_i$. The fan-in of the symmetric gate is at most

$q^k \cdot N^{2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})} \leq N^{2q(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})}$, and the fan-in of an AND gate is at most $2(q-1)(\delta_{\text{AND}}+2\log\delta_{\text{SYM}})$. ■

C. From single to multiple iterations

We conclude by applying Lemma 8 in each iterative step of our depth-reduction.

Theorem 1 (formally stated).: There is an explicit construction such that for every input length n of an arbitrary ACC_m circuit of depth d and size s, this construction outputs a depth 2 circuit $\mathrm{SYM}\circ\mathrm{AND}$ of size $2^{(m\log s)^{(10\Delta(m)d)}}$ where the fan-in of each AND gate is $(m\log s)^{10\Delta(m)d}$, where $\Delta(m)$ is the number of distinct prime divisors of m. More precisely, if the size of the circuit if $2^{\log^k n}$, then the size of the output circuit is $2^{(m\log^k n)^{10\Delta(m)d}}$.

Proof: Given an ACC_m circuit, we first use Lemma 2 to construct a $\mathrm{SYM}\circ\mathrm{ACC}$ circuit with depth $2\Delta(m)\cdot d$ size $2^{(m\log s)^3}$ AND gate fan-in $(m\log s)^2$, where $\Delta(m)$ is the number of distinct prime divisors of m. Recall that each layer have only one type of gates: AND or MOD_q, where q is a prime divisor of m. We do the depth-reduction inductively from top to bottom (input level) of the circuit and reduce the whole circuit into a $\mathrm{SYM}_{[2^{(\log s)^{10\Delta(m)\cdot d}}]}\circ\mathrm{AND}_{[(\log s)^{10\Delta(m)\cdot d}]}$ circuit. Denote by $\delta_{\text{SYM},i}$ the fan-in of the symmetric gate we get from reducing the first i layers, $\delta_{\text{AND},i}$ is the biggest AND gate fan-in.

The top layer of the circuit is a SYM gate (in fact, a "majority" gate), therefore the given circuit is of the form $\mathrm{SYM}\circ\mathrm{AND}$. Then, $\delta_{\text{SYM},1} \leq 2^{(m\log s)^{1\cdot 5}}$, $\delta_{\text{AND},1} \leq (m\log s)^{1\cdot 5}$

Suppose we have already reduced the first i layers into a $\mathrm{SYM}\circ\mathrm{AND}$ circuit. Then, $\delta_{\text{SYM},i} \leq 2^{(m\log s)^{i\cdot 5}}$, $\delta_{\text{AND},i} \leq (m\log s)^{i\cdot 5}$.

For the layer $i+1$:

Case: AND layer. Each gate of the $i+1$ layer is the AND of some gates from the $i+2$ layer. Simply replace the each gate of the $i+1$ layer with the products of its inputs. We can get a $\mathrm{SYM}\circ\mathrm{AND}$ circuit with $\delta_{\text{SYM},i+1} = \delta_{\text{SYM},i} = 2^{(m\log s)^{i\cdot 5}} \leq 2^{(m\log s)^{(i+1)\cdot 5}}$, $\delta_{\text{AND},i+1} \leq (m\log s)^2\cdot\delta_{\text{AND},i} \leq (m\log s)^{(i+1)\cdot 5}$ by induction hypothesis.

Case: MOD_q layer. We think of the outputs of all gates in layer $i+2$ as inputs to the first $i+1$ layers of the circuit. Then, the "input size" of layer $i+1$ is at most the size of the circuit i.e. $2^{O((m\log s)^3)}$. The first 3 layers of the circuit gotten by the compressing from induction hypothesis form a $\mathrm{SYM}\circ\mathrm{AND}\circ\mathrm{MOD}_q$ circuit. We use Lemma 8 to compress.[8] Then, $\delta_{\text{SYM},i+1} \leq (2^{(m\log s)^3})^{2q(\delta_{\text{AND},i}+2\log\delta_{\text{SYM},i})} \leq 2^{(m\log s)^{(i+1)\cdot 5}}$, and $\delta_{\text{AND},i+1} \leq 2(q-1)(\delta_{\text{AND},i}+2\log\delta_{\text{SYM},i}) \leq (m\log s)^{(i+1)\cdot 5}$ by induction hypothesis and Lemma 8.

Thus, after reducing the depth $2\Delta(m)\cdot d$ of the circuit, we get a $\mathrm{SYM}\circ\mathrm{AND}$ circuit with norm at most $2^{(m\log s)^{10\Delta(m)\cdot d}}$ and degree at most $(m\log s)^{10\Delta(m)\cdot d}$. ■

Thus, we got a $2^{(m\log s)^{10\Delta(m)d}}$ size and $(m\log s)^{10\Delta(m)d}$ degree $\mathrm{SYM}\circ\mathrm{AND}$ circuit to which is equivalent with the given ACC_m circuit. Especially for ACC_6, the size and degree would be $2^{\log^{20d} s}$ and $\log^{20d} s$.

IV. Some implications

We list two main implications of the new depth-reduction construction. Section IV-A shows a near-exponentially better depth lower bound in Williams' program. This is an immediate consequence of Theorem 1. Regarding non-immediate consequences, Section IV-B contains an application of our depth-reduction construction (but not the statement of Theorem 1). This is the first super-constant-depth lower bounds in a hybrid model of communication complexity and circuit complexity. Here, we still use depth-reduction. The technical challenge is to reduce the depth of an exponentially big circuit.

A. From depth $o(\log\log n)$ to $o(\log n/\log\log n)$ – a new barrier to Williams' program

As explained at the end of Section I, our improved depth-reduction (Theorem 1) yields a super-exponentially better depth lower bound over the previous best-known one.

Theorem 9. $\mathrm{NEXP} \not\subseteq \mathrm{ACC}(2^{\log^k n}, o(\frac{\log n}{\log\log n}))$ *for every constant k.*

In particular, for a fixed m we obtain the following detailed bound.

Corollary 10. *For a fixed modulus m, and a constant k, there exist a constant $c(m,k)$ such that* $\mathrm{NEXP} \not\subseteq \mathrm{ACC}_m(2^{\log^k n}, \frac{c(m,k)\log n}{\log\log n})$

Note, that the above lower bound pushes Williams' program to the NC^1 barrier. By this we mean that any $\omega(1)$ improvement on the depth bound directly implies $\mathrm{NEXP} \not\subseteq \mathrm{NC}^1$, since $\mathrm{NC}^1 \subseteq \mathrm{AC}(n^{O(1)}, O(\frac{\log n}{\log\log n}))$. In fact, the barrier we are facing now is much stronger since we allow MOD_m gates.

[8]Notice that the AND layers are generated by inductively using the depth reduction algorithm. So, the linearization step is necessary even when there are no AND gates in the original circuit.

Finally, we remark that after the depth-reduction step the Circuit-SAT algorithm is for circuits of the form $\mathrm{SYM} \circ \mathrm{AND}$. The fact that the top gate is SYM is crucial in e.g. [19] and it is not known whether restricted SYM gates can yield faster algorithms (and thus better lower bounds) – see in Green et al. [29] for a variant of Beigel-Tarui with the SYM gate restricted in the so-called MidBit form.

Proof outline of Theorem 9: Our depth-reduction algorithm can compress every ACC circuit of depth $o(\log n/\log\log n)$ to a sub-exponential depth-2 circuit.

Corollary 11 (from Theorem 1)**.** *Given an arbitrary $2^{(\log n)^{O(1)}}$-size and $o(\log n/\log\log n)$-depth* ACC *circuit, there is a explicit construction of an equivalent $2^{o(n)}$-size* $\mathrm{SYM} \circ \mathrm{AND}$ *circuit.*

Now, we state two theorems from [18] that enable us to conclude Theorem 9.

Theorem 12 ([18])**.** *Let $\mathfrak{C}$ be any circuit class, for which $\mathrm{OR}_{[n^{\omega(1)}]} \circ \mathfrak{C}$ can be computed by a equivalent $2^{o(n)}$ size* $\mathrm{SYM} \circ \mathrm{AND}$ *circuit. Then, $\mathfrak{C}$-SAT can be solved in $\frac{2^n}{n^{\omega(1)}}$ time.*

Thus, Corollary 11 and Theorem 12 imply a faster than exhaustive search circuit-SAT algorithm for $\mathrm{ACC}(2^{\log^k n}, o(\frac{\log n}{\log\log n}))$ for every integer k.

We conclude Theorem 9 through Theorem 13.

Theorem 13. *[18] Let $\mathfrak{C}$ be any circuit class which closed under composition and contains AC^0. If $\mathfrak{C}$-SAT has a $\frac{2^n}{n^{\omega(1)}}$ running time algorithm, then $\mathrm{NEXP} \not\subseteq \mathfrak{C}$.*

■

B. Interactive compression

One way to strengthen the ACC lower bound is to consider the following interactive setting, introduced by Chattopadhyay and Santhanam [22] for ACC_p, where p is prime. Here we show the first lower bound in this setting for composite modulus, i.e. for the general ACC.

The setting, coined as *interactive compression* [22], is a communication game between Alice and Bob. In this game, Alice holds an n-bit input x and she wants to decide whether $x \in L$ for some problem L. Her power is restricted to only access a circuit from a fixed class of circuits $\mathfrak{C}$ that cannot compute L. To that end, she is communicating with a computationally unbounded Bob. We call this communication game $\mathfrak{C}$*-compression game for* L. For a fixed protocol the *cost* of the game is the number of bits communicated. For details and definitions see [22], [23].

Our work, same as in [22], [23], is about unconditional lower bounds. Note that the work of Fortnow and Santhanam [30] and Dell and van Melkebeek [31] shows strong but conditional lower bounds in similar interactive compression settings.

Chattopadhyay and Santhanam [22], and the subsequent strengthening and simplification by Oliveira and Santhanam [23], proved communication lower bounds for explicit functions, such as MOD_q [22], [23] and the majority function MAJ [23]. Both of these works are based on correlation bounds between ACC_p circuits and explicit functions, originally shown by Razborov and Smolensky [7], [8]. However, no such correlation bounds are known for composite moduli, even for a depth-2, ACC circuits. Thus, on one hand we strengthen Alice's power by giving her access to ACC_m circuits for composite m, but also weaken the conclusion to deriving NEXP lower bounds (that reaches the limits of current knowledge).

We show an interactive compression NEXP-lower bound for an Alice that has the power of $\mathrm{ACC}(2^{(\log n)^{O(1)}}, o(\log n/\log\log n))$. To that end, we introduce a technique very different than [22], [23], which uses our depth-reduction construction together with [19].

1) Formalization of interactive compression: Let us begin with the definition of an interactive compression game. For background, examples (e.g. the parity upper bound), and formal definitions cf. [22].

Definition 14. *A* $\mathfrak{C}$-compression game *for a function $f : \{0,1\}^n \to \{0,1\}$ is a two-party communication game, where the first party, Alice, is given the entire input x and is restricted to make decisions computed by $\mathfrak{C}$-circuits, while the second party, Bob, is not given any input and is computationally unbounded. The two parties realize a $\mathfrak{C}$-bounded interactive communication protocol to decide the value of $f(x)$.*
Syntactically, a $\mathfrak{C}$-bounded protocol *consists of a sequence of finite circuits $\{C_n\}$, $C_n \in \mathfrak{C}$ that Alice is using to generate her messages. The computationally unbounded Bob is a function from sequences of messages to messages. Here is the description of the* computation of k-round $\mathfrak{C}$-protocol*: at the i-th round Alice sends a message $y_i \in \{0,1\}^*$ to Bob and if i is not the last round Bob replies with a message $z_i \in \{0,1\}^*$. The message y_i is generated by applying a number of consecutive (and fixed) $\mathfrak{C}$-circuits on $< x, z_1, z_2, \ldots, z_{i-1} >$, and z_i is generated by applying a number of fixed boolean functions on $< y_1, y_2, \ldots, y_i >$. At the end of the k-th round Bob applies a boolean function from messages to*

messages used to decide the value of f.
The communication cost *of the protocol is the maximum number of bits sent by Alice as a function of $n = |x|$.*

The number of bits sent by Bob is not counted in the communication cost. However, this number is bounded by the size of $\mathfrak{C}$-circuit, since the number of bits that can be accessed by Alice is bounded by the circuit size.

Due to the space limitation, we only state the prove outline here. See the full version [32] for details.

2) Our interactive compression lower bound: We prove the following theorem, which is a strengthened version of the NEXP lower bound of Theorem 9.

Theorem 15. *The cost of a k-round quasi-poly size, $o(\frac{\log n}{\log\log n})$ depth* ACC*-compression game for* NEXP *is at least $n^{\frac{1}{k}-\varepsilon}$ for every $\varepsilon > 0$.*

Full proofs of this theorem together with all full proofs are given in the full version [32].

Acknowledgments

We are most thankful to Kristoffer Hansen and to Ryan Williams for the very useful discussions, suggestions, and corrections in some of the first expositions of this work. We are also thankful to Paul Beame, Richard Beigel, Oded Goldreich, Dick Lipton, Ken Regan, Avi Wigderson, and Xin Yang for remarks, corrections, and various technical and stylistic suggestions.

References

[1] E. Allender and U. Hertrampf, "Depth reduction for circuits of unbounded fan-in," *Information and Computation*, vol. 112, no. 2, pp. 217–238, 1994, (also FOCS'89).

[2] A. C.-C. Yao, "On ACC and threshold circuits," in *Foundations of Computer Science (FOCS)*. IEEE, 1990, pp. 619–627.

[3] R. Beigel and J. Tarui, "On ACC," *Computational Complexity*, vol. 4, no. 4, pp. 350–366, 1994, (also FOCS'91).

[4] J. Håstad, "Computational limitations of small-depth circuits," Ph.D. dissertation, MIT, 1987.

[5] A. C.-C. Yao, "Separating the polynomial-time hierarchy by oracles," in *Foundations of Computer Science (FOCS)*. IEEE, 1985, pp. 1–10.

[6] B. Rossman, R. A. Servedio, and L.-Y. Tan, "An average-case depth hierarchy theorem for boolean circuits," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 2015, pp. 1030–1048.

[7] A. Razborov, "Lower bounds on the size of bounded depth networks over a complete basis with logical addition, mathematische zametki 41 pp. 598–607," *English Translation inMathematical Notes of the Academy of Sciences of the USSR*, vol. 41, pp. 333–338, 1986.

[8] R. Smolensky, "Algebraic methods in the theory of lower bounds for boolean circuit complexity," in *Symposium on Theory of Computing (STOC)*. ACM, 1987, pp. 77–82.

[9] J. Bourgain, "Estimation of certain exponential sums arising in complexity theory," *Comptes Rendus Mathematique*, vol. 340, no. 9, pp. 627 – 631, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1631073X05001093

[10] A. Chattopadhyay, N. Goyal, P. Pudlak, and D. Therien, "Lower bounds for circuits with mod_m gates," pp. 709–718, 2006.

[11] A. Chattopadhyay, "Discrepancy and the power of bottom fan-in in depth-three circuits," in *Foundations of Computer Science (FOCS)*. IEEE, 2007, pp. 449–458.

[12] A. Chattopadhyay and A. Wigderson, "Linear systems over composite moduli," in *Foundations of Computer Science (FOCS)*. IEEE, 2009, pp. 43–52.

[13] A. Chattopadhyay and S. Lovett, "Linear systems over finite abelian groups," in *Conference on Computational Complexity (CCC)*. IEEE, 2011, pp. 300–308.

[14] F. Green, A. Roy, and H. Straubing, "Bounds on an exponential sum arising in boolean circuit complexity," *Comptes Rendus Mathematique*, vol. 341, no. 5, pp. 279–282, 2005.

[15] V. Grolmusz and G. Tardos, "Lower bounds for (mod p-mod m) circuits," in *Foundations of Computer Science (FOCS)*. IEEE, 1998, pp. 279–288.

[16] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán, "Threshold circuits of bounded depth," in *Foundations of Computer Science (FOCS)*. IEEE, 1987, pp. 99–110.

[17] H. Straubing and D. Thérien, "A note on MODp-MODm circuits," *Theory of Computing Systems*, vol. 39, no. 5, pp. 699–706, 2006.

[18] R. Williams, "Non-uniform ACC Circuit Lower Bounds," in *Conference on Computational Complexity (CCC)*, 2011, pp. 115–125.

[19] ——, "New algorithms and lower bounds for circuits with linear threshold gates," in *Symposium on Theory of Computing (STOC)*. ACM, 2014, pp. 194–202.

[20] S. A. Cook, "A hierarchy for nondeterministic time complexity," *Journal of Computer and System Sciences*, vol. 7, no. 4, pp. 343–353, 1973.

[21] F. Wang, "NEXP does not have non-uniform quasipolynomial-size acc circuits of o (loglogn) depth," in *Theory and Applications of Models of Computation*. Springer, 2011, pp. 164–170.

[22] A. Chattopadhyay and R. Santhanam, "Lower bounds on interactive compressibility by constant-depth circuits," in *Foundations of Computer Science (FOCS)*. IEEE, 2012, pp. 619–628.

[23] I. C. Oliveira and R. Santhanam, "Majority is incompressible by $ac^0[p]$ circuits," in *Conference on Computational Complexity (CCC)*, 2015, pp. 124–157.

[24] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[25] E. Allender and V. Gore, "On strong separations from AC," *Advances in Computational Complexity Theory*, vol. 13, p. 21, 1993.

[26] L. G. Valiant and V. V. Vazirani, "NP is as easy as detecting unique solutions," in *Symposium on Theory of Computing (STOC)*. ACM, 1985, pp. 458–463.

[27] S. Toda, "On the computational power of pp and? p," in *Foundations of Computer Science (FOCS)*. IEEE, 1989, pp. 514–519.

[28] S. Lang, "Algebra (revised third edition)," *Graduate Texts in Mathematics*, vol. 1, no. 211, 2002.

[29] F. Green, J. Kobler, K. W. Regan, T. Schwentick, and J. Torán, "The power of the middle bit of a# p function," *Journal of Computer and System Sciences*, vol. 50, no. 3, pp. 456–467, 1995.

[30] L. Fortnow and R. Santhanam, "Infeasibility of instance compression and succinct pcps for np," in *Symposium on Theory of Computing (STOC)*. ACM, 2008, pp. 133–142.

[31] H. Dell and D. van Melkebeek, "Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses," in *Symposium on Theory of Computing (STOC)*. ACM, 2010, pp. 251–260.

[32] S. Chen and P. A. Papakonstantinou, "Depth reduction for composites," ECCC, Tech. Rep. 85, May 2016 http://eccc.hpi-web.de/report/2016/085/.