Circuits with composite moduli

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Shiteng Chen

Dissertation Supervisor: Assistant Professor Periklis A. Papakonstantinou

May 2016

Circuits with composite moduli

by

Shiteng Chen

Submitted to the Institute for Interdisciplinary Information Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at

TSINGHUA UNIVERSITY

May 2016

Author . . . .
Institute for Interdisciplinary Information Sciences
May 4, 2016

Certified by . . . .
Periklis A. Papakonstantinou
Assistant Professor
Thesis Supervisor

Circuits with composite moduli

by

Shiteng Chen

Submitted to the Tsinghua University
on May 4, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This work studies the power of ACC circuits. These are circuits that have modular gates, in addition to the usual $\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$. We are particularly interested in circuits where the modulus is a composite number.

Our main result is that every ACC circuit of polynomial size and depth d can be reduced to a depth-2 circuit SYM$\circ$AND of size $2^{(\log n)^{O(d)}}$. This improves exponentially the previously best-known construction by Yao-Beigel-Tarui, which has size blowup $2^{(\log n)^{2^{O(d)}}}$. Therefore, depth-reduction for composites asymptotically matches the circuit size in the Allender-Hertrampf construction for primes from 1989.

An immediate corollary of our depth-reduction is that NEXP cannot be computed by families of non-uniform ACC circuits of size quasi-polynomial and depth up to $o(\log n/\log\log n)$. This is a nearly-exponential improvement of the previous best depth lower bound in Williams' program. In fact, it pushes William's program to the NC^1 barrier.

Our depth-reduction works also for a special form of exponential size ACC circuits. This is used to obtain the first lower bound for general ACC circuits in the interactive compression setting of Chattopadhyay, Oliverra, and Santhanam.

The dissertation concludes with a technical remark about the limitations of the correlation method, which is an important tool for proving circuit lower bounds. Green, Roy, and Straubing asked in 2005 whether correlation bounds can be generally used forcircuits with composite modular gates. We provide a partial negative answer to this question.

Dissertation Supervisor: Assistant Professor Periklis A. Papakonstantinou

Acknowledgments

I would like to give my special thanks to my supervisor Periklis A. Papakonstantinou. Without his help and his supervision I would never have been able to achieve these results. I also want to give my thanks to Andrew Yao and other professors and postdoctors in IIIS for providing a good research environment.

I want to thank my coauthors: Eric Allender, Joshua Brody, Zhiyi Huang, Kai Jin, Sampath Kannan, Francis Chi-Moon Lau, Tiancheng Lou, Periklis A. Papakonstantinou, Dominik Scheder, Hao Song, Xiaoming Sun, Navid Talebanfard, Haisheng Tan, Bangsheng Tang, Pingzhong Tang, Elad Verbin, Yuexuan Wang, Wei Yu. It is a great experience working with these talented researchers. In particular, I would like to thank Periklis for sharing his brilliant conjectures and working together with me to prove them.

I would also like to thank for Eric Allender, Paul Beame, Stephen Cook, Russell Impagliazzo, Rahul Santhanam and Xiaoming Sun for their invaluable feedback during the review of the thesis. Let me also thank Richard Beigel, Arkadev Chattopadhyay, Frederic Green, Kristoffer A. Hansen, Richard Lipton, Shachar Lovett, Kenneth Regan, Emanuele Viola, Chengu Wang, Avi Wigderson, Ryan Williams, and Xin Yang for enlightening discussions and very helpful comments regarding my work.

Finally, I want to thank my family and my friends for their support in daily life.

Contents

List of Figures

Definitions and notation

Standard notations

$\log n$: the base 2 logarithm of n

$\mathbb{Z}$: the set of integers

$\mathbb{R}$: the set of real numbers

$\mathbb{C}$: the set of complex numbers

$\mathbb{Z}_n$: the set of integers modulo n (same as $\mathbb{Z}/n\mathbb{Z}$)

$\mathbb{F}_p$: the finite field of characteristic p

$\mathbb{Z}_n^*$: multiplicative group of integers modulo n (same as $(\mathbb{Z}/n\mathbb{Z})^*$)

Functions and Boolean gates

MOD_n .... page 11

SYM .... page 11

ANY .... page 22

MAJ .... page 15

$\mu(n)$: Mobius function .... page 30

$\mathrm{Corr}(f, g)$: correlation .... page 14

$e_m(X)$: $e^{\frac{2\pi X i}{m}}$, where $e^{\frac{2\pi i}{m}}$ is the mth root of unity

$\bmod n$: arithmetical modulo operation

$\Delta(m)$: number of distinct prime divisor of m .... page 25

Computational models

circuit size .... page 1

circuit depth .... page 1

$\mathfrak{C}$-compression game .... Definition 3.2.1 on page 42

Complexity classes

AC . . . . page 3

AC^0 . . . . page 3

ACC . . . . page 3

ACC^0 . . . . page 3

ACC_m . . . . page 4

$\mathrm{ACC}(s, d)$ . . . . page 24

NC^1 . . . . page 20

NEXP . . . . page 16

$\mathrm{NTime}(f(n))$ . . . . page 19

P/Poly: class of decision problems characterized by non-uniform polynomial size families of circuits

P: class of decision problems solvable in uniform polynomial time

NP: class of decision problems nondeterministically solvable in uniform polynomial time

PSPACE: the class of decision problems solvable in uniform polynomial space

EXP: the class of decision problems solvable in uniform time $2^{n^{O(1)}}$

Chapter 1

Introduction

Boolean circuits are abstract models of computation for digital circuits in electronic engineering. This model is also widely used for describing parallel algorithms. As a field of study, circuit complexity predates the construction of the first computer ENIAC. The beginings of circuit complexity can be found in the work of Shannon and even 100 years before that in the work of Boole was doing 1847.

Circuit complexity is one of the most challenging and fruitful branches of complexity theory. It aims to understand the computational power of Boolean circuits. A Boolean circuit with n input bits is an acyclic directed graph in which every node is either an input node or a Boolean gate. An input node has in-degree 0 and is labeled by one of the n input bits. A Boolean gate is usually an AND gate, an OR gate, or a NOT gate. One of these gates is designated as the output gate. This model is very powerful. A straightforward way to construct a circuit for any Boolean function $f : \{0,1\}^n \to \{0,1\}$ is by simply representing its truth table. We often use two parameters to measure resources for a given circuit: the *size* of the circuit, which is the number of wires, i.e. directed edges of the graph, and the *depth*, which is the length of the longest path from the output to any input. Our goal is to either design circuits for explicit functions, or to obtain provable limitations; i.e. prove lower bounds for the resources required to compute functions. As with most other branches of complexity theory, these two main goals are closely related. The main contributions of this thesis regard this interplay. For instance, as our main

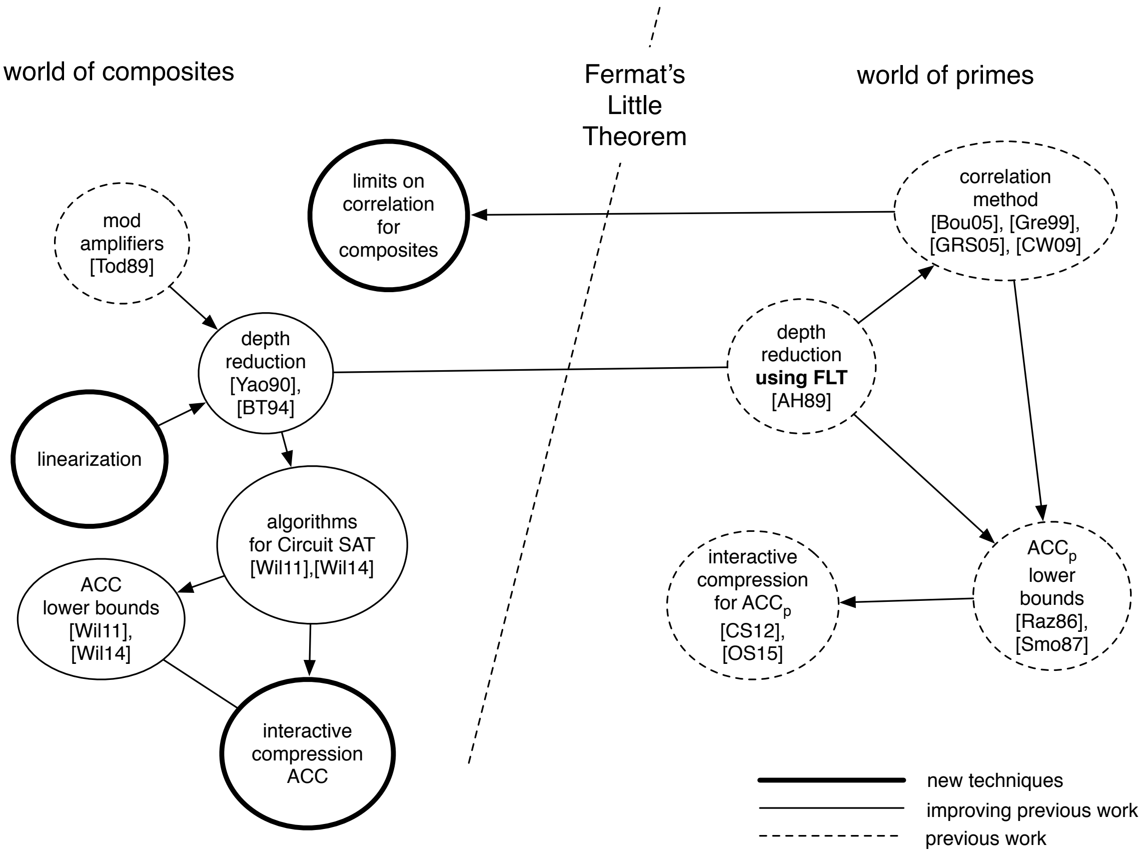
world of composites
Fermat's Little Theorem
world of primes
limits on correlation for composites
correlation method [Bou05], [Gre99], [GRS05], [CW09]
mod amplifiers [Tod89]
depth reduction [Yao90], [BT94]
depth reduction **using FLT** [AH89]
linearization
algorithms for Circuit SAT [Wil11],[Wil14]
ACC lower bounds [Wil11], [Wil14]
interactive compression ACC
interactive compression for ACC_p [CS12], [OS15]
ACC_p lower bounds [Raz86], [Smo87]
new techniques
improving previous work
previous work

Figure 1-1: Results of this dissertation and relation to previous work. The main contribution is an exponential size improvement on the depth-reduction for composites.

contribution we exponentially improve the size of the depth-reduction algorithm of Yao-Beigel-Tarui [Yao90, BT94]. Depth-reduction is an algorithm that reduces the depth of some specific kinds of circuits to 2. Together with the somewhat recently introduced program of Williams [Wil11, Wil14], this algorithm yields new, nearly exponentially-stronger depth lower bounds. Figure 1-1 summarizes this and the rest of our contributions.

To put things in context and discuss our motivation we first revisit the relevant seminal developments in circuit complexity.

1.1 From circuits to small-depth circuits

The central problem in circuit complexity is to understand the computational power of circuits with restricted resources, e.g. circuit size and depth. The first circuit size lower bound was given in 1949 in the seminal work of Shannon [Sha49], where it was shown that almost every n-bit input Boolean function requires a circuit of size at least $\Omega(2^n/n)$ to be computed. Subsequently, Lupanov [Lup58] showed that this bound is tight by an explicit construction. For more than half a century, proving size lower bounds, especially super-polynomial lower bounds for explicit Boolean functions has remained one of the most interesting open problems of circuit complexity. Unfortunately, after so many years, no super-polynomial size lower bounds have been given for any explicit function for general forms of circuits.

On the other hand, super-polynomial size lower bounds have been given for restricted circuits and in particular for families of circuits of constant depth. The parity function, i.e. the sum of the input bits modulo 2, it was proved to be incomputable by any constant depth polynomial size circuits with AND gates, OR gates and NOT gates, also known as AC^0 circuits. This was shown independently by Ajtai [Ajt83] and by Furst, Saxe and Sipser [FSS84]. Later, this result was improved by Hastad [Hås87] who showed the parity function can not be computed by constant depth subexponential sized circuits composed only with AND gates, OR gates, and NOT gates.

After this first success in proving lower bounds, the next natural question was: what happens if we give circuits the power of computing the parity function? That is, what if we can use a gate with polynomially many input bits that adds them up modulo 2, or more generally modulo m? We write MOD_m to denote a Boolean gate, i.e. with 0/1 output, which outputs 1 if and only if the number of 1s in its input is a multiple of m. This is very different compared to $\bmod m$ integer functions, since the output of MOD_m is always 0 or 1. The kinds of circuits constructed by AND, OR, NOT and MOD are called ACC circuits. Circuits with modular gates play a special role in this dissertation. Starting from the next section, we discuss our motivation and explain our developments. Understanding the power of ACC circuits is an extremely challenging and fascinating task. Researchers acknowledge the huge gap between the things we know for the power of prime modular gates and modular gates where m is composite. The results in this dissertation make this gap smaller.

1.2 Circuits with moduli: the theme of this work

In 1986, Razborov [Raz86] proved that MOD_q gates for an *odd* prime number q, can not be computed by a constant depth polynomial sized circuit composed of AND, OR, NOT and Parity gates, i.e. ACC_2^0 circuits in our notation. Smolensky [Smo87] improved this result by proving Razborov's conclusions still hold even when given MOD_p gates, for a prime number $p \neq q$. Presently, we give the formal statement of this theorem.

Theorem *For any two distinct prime numbers p and q we have* $\mathrm{MOD}_q \notin \mathrm{ACC}_p^0$

Here, ACC_p^0 stands for the complexity class of functions characterized by families of constant depth polynomial sized circuits composed of AND, OR, NOT and MOD_p gates – we write ACC_p both to refer to specific circuits and to the complexity class characterized by families of these circuits. The heart of the arguments which prove the theorems of Razborov and Smolensky stem from the view of ACC_p circuits as low-degree polynomials over the $\mathbb{F}_p$ field. These results are shown in two simple steps, which we outline below (although we do not use them later on in our technical developments):

First, show that for any ACC_p^0 circuit C with n bit input, there exists a degree $o(\sqrt{n})$ polynomial P, such that $P(x) \mod p$ agree with $C(x)$ on at least a $1 - o(1)$ fraction of inputs.

Second, show that for any prime q co-prime with p, there does not exist any degree $o(\sqrt{n})$ polynomial P such that $P(x) \mod p$ agrees with $\mathrm{MOD}_q(x)$ on 0.999 fraction of inputs.

The two steps directly imply that $\mathrm{MOD}_q \notin \mathrm{ACC}_p^0$.

Smolensky conjectured that the same holds for any co-prime moduli m and r.

Conjecture 1.2.1 ([Smo87]). *For any two co-prime integers m and r we have that* $\mathrm{MOD}_r \notin \mathrm{ACC}^0_m$

This conjecture is a central motivation for our work. The big technical problem is that the first step does not work when m is composite, since $\mathbb{Z}_m$ ceases to be a field.

The two aforementioned steps gave rise to two very important technical tools in circuit complexity: the *depth-reduction* and the *correlation method.*

1.2.1 Depth-reduction

Depth-reduction is an algorithm that compresses a low depth ACC circuit, referred to as a highly parallel algorithm, into a depth 2 circuit, which is an extremely parallel algorithm. As an algorithm, it is very important in parallel computation. Besides depth-reduction's importance in its own right, it is also widely used in complexity theory. We already saw that the ACC_p lower bounds [Raz86, Smo87] make (implicit) use of it and more recently in the proof of circuit lower bounds for NEXP; i.e. the family of Boolean functions computable by an algorithm nondeterministically in time $2^{n^{O(1)}}$, in Williams' Program [Wil11, Wil14]. In this dissertation, we improve the depth-reduction of Yao-Beigel-Tarui [Yao90, BT94] and improve the circuit lower bound of Williams [Wil11, Wil14] via the depth-reduction algorithm.

1.2.1.1 For prime moduli

The idea of representing an ACC_p circuit as a low degree polynomial (as previously discussed in Section 1.2 in regard to Razborov and Smolensky [Raz86, Smo87]) is also discussed in the work of Allender and Hertrampf [AH94] where an explicit construction of this polynomial is given. On the other hand, the property that the depth of ACC_p circuits can be reduced is a fundamental difference between ACC_p and AC circuits. Yao [Yao90] and Håstad [Hås87] showed that the depth of AC circuits can not be reduced without an exponential size blowup even from depth k to $k-1$.

This was proved using random restrictions. The same holds true even in the average-case, and it was proved through the so-called random projections technique in the recent breakthrough work by Rossman, Servedio, and Tan [RST15].

The following theorem is a simplified version of the depth-reduction result for prime moduli.

Theorem 1.2.2 ([Smo87, AH94])**.** *Given an* ACC_p *circuit* C *with prime modulus* p*, of depth* d*, input length* n*, and size* $s \geq n$ *(for* ACC^0*, the size is polynomial and depth is constant), there exists a degree* $\log^{O(d)} s$ *polynomial* P*, such that for any input* x*,* $P(x) \mod p$ *agrees with* C *on at least a* $1 - o(1)$ *fraction of inputs.*

We will now provide the (elegant) details of the construction in proving this theorem. Parts of this "warm-up construction" are going to be used later on in our main depth-reduction algorithm 2.2.6. We will treat each gate of the ACC_p circuit as a low degree polynomial over $\mathbb{F}_p$. We will also treat the Boolean value 0 and 1 as 0 and 1 in field $\mathbb{F}_p$, and all of the polynomial evaluations are over $\mathbb{F}_p$.

NOT For any NOT gate with input bit X:

$$\mathrm{NOT}(X) = 1 - X$$

MOD For a MOD_p gate with k bit input $\{X_1, X_2, \ldots, X_k\}$, by FLT (Fermat's Little Theorem), i.e. for any integer X and a prime p, $X \neq 0 \mod p$ implies $X^{p-1} = 1$ mod p, we get the following:

$$\mathrm{MOD}_p(X_1, X_2, \ldots, X_k) = 1 - (\sum_{1 \leq i \leq k} X_i)^{p-1}$$

Here, X_i are elements of $\mathbb{F}_p$.

Note that the application of FLT is a key step of turning the MOD gates into low degree polynomials. In our depth-reduction a composite will be decomposed into its prime factors and thus a size loss same as in the above step will occur in our construction as well.

AND and OR AND and OR gates with k bits input $\{X_1, X_2, \dots, X_k\}$ can be easily represented by the following formulas:

$$\text{AND}(X_1, X_2, \dots, X_k) = \prod_{1 \le i \le k} X_i$$

$$\text{OR}(X_1, X_2, \dots, X_k) = 1 - \prod_{1 \le i \le k} (1 - X_i)$$

For a ACC$_p$ circuit, however, the fan-in, i.e. the number of the input bits, of an AND or OR gate can be as big as polynomial of n. So the representations above are of too high degree. In order to reduce the degree, we replace these AND and OR gates with some randomized polynomials, i.e. polynomial with random coefficients. These polynomials equal to the given gates with high probability. We show this technique for OR gates as a representative example.

For any fixed input $\{X_1, X_2, \dots, X_k\}$, denote $R_i(X_1, X_2, \dots, X_k) = \sum_{1 \le j \le k} r_{i,j} X_j$ where $r_{i,j}$ is random number uniformly sampled from $\mathbb{F}_p$, then:

If $X_1 = X_2 = \cdots = X_k = 0$, then $\text{OR}(X_1, X_2, \dots, X_k) = 0$, $R_i(X_1, X_2, \dots, X_k) = 0$ with probability 1.

Otherwise, $\text{OR}(X_1, X_2, \dots, X_k) = 1$, $R_i(X_1, X_2, \dots, X_k) = 0$ with probability $\frac{1}{p}$. By FLT, $R_i^{p-1}(X_1, X_2, \dots, X_k) = 0$ with probability $\frac{1}{p}$ and $R_i^{p-1}(X_1, X_2, \dots, X_k) = 1$ with probability $1 - \frac{1}{p}$.

In order to increase the probability of correctness, we use independent copies of these polynomials. We sample $R_1, R_2, \dots, R_l$ independently, where l is another parameter which will be determined in a later argument. We denote the polynomial as $P' = 1 - \prod_{1 \le i \le l}(1 - R_l^{p-1})$.

Thus, we have that for any fixed input $\{X_1, X_2, \dots, X_k\}$, with probability at least $1 - p^{-l}$ we have that $\text{OR}(X_1, X_2, \dots, X_k) = P'(X_1, X_2, \dots, X_k)$.

For the whole circuit For an ACC$_p$ circuit C with size s, replace each gate with a polynomial as described above. In the end, we obtain a polynomial P''. By the union bound we have

$$\Pr(C(x) = P''(x)) \geq 1 - \frac{s}{p^l}$$

Fix $l = \kappa \cdot \log s$ for a large enough constant κ and thus we have:

$$\Pr(C(x) = P''(x)) = 1 - o(1)$$

on any fixed input x. Pick polynomial P as the polynomial from the sample space of P'', which is the closest C; i.e. it agrees with C on the largest fraction of inputs. Then, it is easy to see that P agrees with C on at least a $1 - o(1)$ fraction of inputs. The degree of this polynomial is also not very high:

$$\deg(P) \leq (pl)^d = \log^{O(d)} s = o(\sqrt{n})$$

In the work of Allender and Hertrampf [AH94], the above argument was improved via an explicit construction of such a polynomial using the sample space constructed by Valiant and Vazirani [VV85].

1.2.1.2 For composite moduli

For composite modulus m, the argument becomes much more complicated. One technical issue arises because MOD_m can not be represented as a constant degree polynomial over any finite field or even a ring. Let $m = \prod_{1 \leq i \leq \Delta(m)} p_i^{\alpha_i}$ be the prime factorization of m. In order to identify MOD gates by a polynomial, we decompose the MOD_m gates as a circuit with AND, OR, NOT, and MOD_{p_1}, MOD_{p_2}, $\ldots$, $\mathrm{MOD}_{p_{\Delta(m)}}$ gates. This preprocessing step is stated formally as part of Lemma 2.1.1 on page 25 without proof; for details see [BT94].

After this preprocessing, we obtain a circuit with different kinds of MOD gates. We also arrange this circuit such that there is only one kind of gate at each layer. Now, these gates can be represetined by low degree polynomials over different fields. In order to link these polynomials over different fields, we use Mod-Amplifiers, introduced by Toda [Tod89] for proving $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$ and improved by Yao [Yao90] and

Beigel and Tarui [BT94].

Lemma 1.2.3 (Mod-Amplifiers [BT94], weaker forms in [Tod89, Yao90], restated as Lemma 2.1.3 on page 26). *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 $X = 1 \mod m$. We call this* MP_k *as the kth Mod-Amplifier.*

Such a polynomial is constructed as follows:

First, we show that if $P(X)$ is a polynomial with integer coefficients and there exist two polynomials with integer coefficients $f(X)$ and $g(X)$ such that $P(X) = X^k f(X) = (X-1)^k g(X) + 1$, then P is one of the kth Mod-Amplifier

For any integer m, if $X = 0 \mod m$, then $m^k | X^k$, which implies $m^k | P(X) = X^k f(X)$, which implies $P(X) = 0 \mod m^k$.

If $X = 1 \mod m$, that implies $m^k | (X-1)^k$, which implies $m^k | P(X) - 1 = (X-1)^k g(X)$, which implies $P(X) = 1 \mod m^k$.

We presently construct such a polynomial. Let $P_k(X) = 1-(1-X)^k \cdot \sum_{0\le i<k} \binom{-k}{i}(-1)^i X^i$, then the degree of P_k is $2k-1$. On one hand, $P_k(X) = (X-1)^k \cdot (-1)^{k-1} \sum_{0\le i<k} \binom{-k}{i}(-1)^i X^i + 1$; on the other hand, given that the Taylor series of $(1-X)^{-k} = \sum_{0\le i} \binom{-k}{i}(-1)^i X^i$, then

$$\begin{aligned}
P_k(X) &= 1 - (1-X)^k \cdot \sum_{0\le i<k} \binom{-k}{i}(-1)^i X^i \\
&= 1 - (1-X)^k \cdot \sum_{0\le i} \binom{-k}{i}(-1)^i X^i + (1-X)^k \cdot \sum_{k\le i} \binom{-k}{i}(-1)^i X^i \\
&= 1 - (1-X)^k \cdot (1-X)^{-k} + (1-X)^k \cdot \sum_{k\le i} \binom{-k}{i}(-1)^i X^i \\
&= (1-X)^k \cdot \sum_{k\le i} \binom{-k}{i}(-1)^i X^i
\end{aligned}$$

Comparing the LHS (left hand side) and the RHS (right hand side) of the equation above, we find that the coefficients for the X^i terms of P_k for $i < k$ are all 0, which means there exists a polynomial with integer coefficients $f(X)$ such that $P_k(X) =$

$X^k f(X)$.

So, there exist two polynomials with integer coefficients $f(X)$ and $g(X)$ such that $P_k(X) = X^k f(X) = (X-1)^k g(X) + 1$, which means P_k is a kth Mod-Amplifier.

These Mod-Amplifiers are high degree polynomials for which we can amplify the modulus to its power without changing the remainder. We give a simple example to show how to use these polynomials to collapse 2 layers of circuits with different moduli.

Example (Mod-Amplifiers) Let $x_1, x_2, x_3, x_4 \in \{0, 1\}$. Then,

$$\begin{aligned}&(((x_1+x_2+x_3) \mod 2) + ((x_2+x_3+x_4) \mod 2) + ((x_3+x_4+x_1) \mod 2)) \mod 3\\ =&(((x_1+x_2+x_3)^2 \mod 4) + ((x_2+x_3+x_4)^2 \mod 4) + ((x_3+x_4+x_1)^2 \mod 4)) \mod 3\end{aligned}$$

The second line of the equation above is a Mod-Amplification step. It amplifies the inner modulus from 2 to 4, and the following equation is true.

$$\begin{aligned}&(((x_1+x_2+x_3)^2 \mod 4) + ((x_2+x_3+x_4)^2 \mod 4) + ((x_3+x_4+x_1)^2 \mod 4)) \mod 3\\ &\qquad \text{(Since } (x_1+x_2+x_3)^2 \mod 4,\ (x_2+x_3+x_4)^2 \mod 4,\ (x_3+x_4+x_1)^2 \mod 4 \text{ are all 0 or 1)}\\ &\qquad \text{(Then } ((x_1+x_2+x_3)^2 \mod 4 + (x_2+x_3+x_4)^2 \mod 4 + (x_3+x_4+x_1)^2 \mod 4) \in \{0,1,2,3\}\text{)}\\ =&\Big(((x_1+x_2+x_3)^2 \mod 4) + ((x_2+x_3+x_4)^2 \mod 4)\\ &+ ((x_3+x_4+x_1)^2 \mod 4) \mod 4\Big) \mod 3\\ =&(((x_1+x_2+x_3)^2 + (x_2+x_3+x_4)^2 + (x_3+x_4+x_1)^2) \mod 4) \mod 3\end{aligned}$$

Example (counter-example for Mod-Amplifiers) Here is what happens if we do not amplify the modulus.

$$\begin{aligned}&(((x_1+x_2+x_3) \mod 2) + ((x_2+x_3+x_4) \mod 2) + ((x_3+x_4+x_1) \mod 2)) \mod 3\\ \neq&(((x_1+x_2+x_3) + (x_2+x_3+x_4) + (x_3+x_4+x_1)) \mod 2) \mod 3\end{aligned}$$

Note that for $(x_1, x_2, x_3, x_4) = (1, 1, 0, 0)$ the LHS (left hand side) of the formula above equals 2, but the RHS equals 0. The reason for this difference is that 4 is bigger than the value of the formula before $\mod 3$, i.e. $((x_1+x_2+x_3)^2 \mod 4)+((x_2+x_3+x_4)^2 \mod 4) + ((x_3 + x_4 + x_1)^2 \mod 4)) < 4$, but 2 is not.

By amplifying the modulus, we push the (mod) function in the inner layer of the formula out to the outer layer.

After using these Mod-Amplifiers to collapse all of the layers of the circuit, the authors of [Yao90, BT94] get a depth 2 circuit SYM $\circ$ AND, which is described in the following theorem. A symmetric gate SYM is a gate whose output only depends on the number of 1s in its input. Every gate involved in the technical part of this dissertation is symmetric, for example AND, OR, MOD, MAJ.

Theorem 1.2.4 (The depth-reduction algorithm by Yao and Beigel and Tarui [Yao90, BT94])**.** *For any constant m, there is an efficient algorithm that, given an* ACC_m *circuit depth d, input length n, and size $s \geq n$, outputs a depth-2 circuit* SYM$\circ$AND *of size $2^{(\log s)^{2^{O(d)}}}$, with* AND *gate fan-in* $(\log s)^{2^{O(d)}}$*, where* SYM *is a gate whose output depends only on the number of* 1*s in its input.*

Notice that the SYM gate is a function whose value only depends on the Hamming weight of its input, and the Hamming weight of its input is in fact the evaluation of a polynomial function of the inputs of the circuit, and this polynomial function has degree $(\log s)^{2^{O(d)}}$. So the SYM $\circ$ AND circuit can be written as $f(P(x))$, where $f : \{0, 1, 2, 3, \ldots, \delta_{\mathrm{SYM}}\} \to \{0, 1\}$ and P is a $(\log s)^{2^{O(d)}}$ degree polynomial. Here δ_{SYM} stands for the fan-in of the symmetric gate on top of the circuit, which is bounded by $2^{(\log s)^{2^{O(d)}}}$. This is a generalized version of Theorem 1.2.2. This construction pays a lot in terms of size for the Mod-Amplifiers, since the degree of the polynomial increases from $(\log s)^{O(d)}$, which is the case for prime moduli, to $(\log s)^{2^{O(d)}}$, which is what happens here for composites. In fact, every time we use the Mod-Amplifiers to collapse one layer of the circuit inductively, the degree of Mod-Amplifiers increases together with the circuit size. Further more, we have to face the products of many Mod-Amplifiers, and in addition the numbers of Mod-Amplifiers that appear in the product also increase together with the circuit size. These kinds of "double increases" make the final circuit size triple exponential in the depth of the original circuit.

1.2.1.3 Our contribution on depth-reduction

Can depth-reduction be done for composite moduli as efficiently as for prime moduli? This was unknown for more than 25 years, and in this dissertation, we give a positive answer.

Theorem 1.2.5 (formally stated and proved as Theorem 2.2.6 on page 36)**.** *For any constant m, there is an efficient algorithm that takes an* ACC_m *circuit of depth d, input length n, and size $s \geq n$, and 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* 1*s in its input.*

Notice that the explicit construction mentioned in the above theorem is also an efficient algorithm. The main difference between our algorithm and the algorithm of YBT [Yao90, BT94] is an extra step of linearization. We remove one AND layer. Note that this step, initially, makes the size even bigger. A high-level view is given in Figure 1-2.

The removing of the AND layer is done by using a new technique which we call *linearization*, formally stated as Theorem 2.2.1 on page 27. With this technique, we can remove the AND gate layer from a $\mathrm{SYM} \circ \mathrm{AND} \circ \mathrm{MOD}$ structure, making it $\mathrm{SYM} \circ \mathrm{MOD}$ by increasing the size of the circuit a lot (but not hugely). After that, we can use the Mod-Amplifiers to link the SYM gate and MOD gates layer together. Although the linearization step increases the circuit size, at the Mod-Amplifying step, the products of Mod-Amplifiers can be avoided in the Mod-Amplifying step. Thus, we are able to eliminate the "double increasing" and in effect remove one tower of exponentials compared to Yao-Beigel-Tarui [Yao90, BT94].

The linearization step is done by using the exponential sum technique, i.e. an analytic tool introduced by Green [Gre99] for proving circuit lower bounds, and will be discussed further in Chapter 2 and Chapter 4. Through exponential sums we write

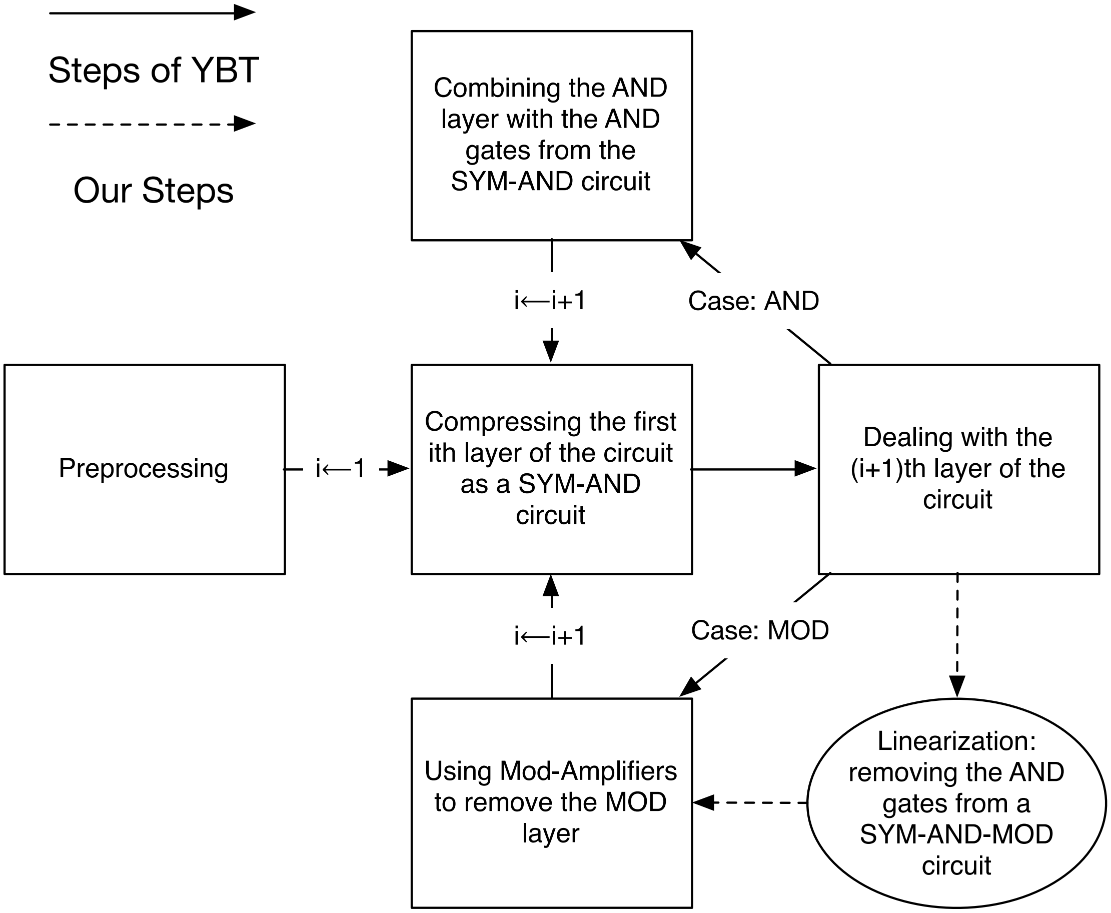
Steps of YBT
Our Steps
Combining the AND layer with the AND gates from the SYM-AND circuit
i⟵i+1
Case: AND
Preprocessing
i⟵1
Compressing the first ith layer of the circuit as a SYM-AND circuit
Dealing with the (i+1)th layer of the circuit
i⟵i+1
Case: MOD
Using Mod-Amplifiers to remove the MOD layer
Linearization: removing the AND gates from a SYM-AND-MOD circuit

Figure 1-2: Our depth-reduction algorithm vs YBT

MOD gates as sums of roots of unity and then by appropriately manipulating the products we can go back to a different form of MOD gates. Note that although the exponential sum technique has been widely used in proving correlation bounds, this is to the best of our knowledge its first use (Theorem 2.2.6) in getting an algorithm.

1.2.2 The correlation method

The main tool in proving lower bounds for restricted and small-depth circuits is proving correlation upper bounds [Bou05, CGT96, Gre99, GRS05, HMP$^+$87, CW09, CL11]. The notion of *correlation* quantifies the distance of two functions and was introduced by Hajnal et al. [HMP$^+$87]. One intuitive way to show a circuit lower bound for computing a given function using certain kind of circuits, is to prove that this kind of circuit correlates little with the given function. The smaller the correlation between the circuit and a function the larger the circuit size to compute this function. Recall the second step of the proof of the theorem of Smolensky 1.2. There it is in fact shown that the correlation the functions $\mathrm{MOD}_p(P(x))$ and MOD_q is small when $\deg(P) = o(\sqrt{n})$.

In Chapter 4, we show a limitation of the correlation method, aiming to answer the question of Green et al. [GRS05]. They asked whether it is possible to prove correlation upper bounds that yield size lower bounds for circuits of the form $\mathrm{MOD}_m \circ \mathrm{AND}_{[\omega(\log n)]}$. Note that these circuits correspond to functions $\mathrm{MOD}_m(P(x))$, for a polynomial P of degree $\omega(\log n)$. We show a correlation *lower bound* between MOD_r and $\mathrm{MOD}_m(P(x))$ where m, r are co-prime integers and P is of any degree. Previously, Green [Gre02] and Viola [Vio09] discussed correlation lower bounds that differ from ours. Viola's argument is for the correlation between symmetric functions and polynomials of degree $\sqrt{n}$ (i.e. high degree) over F_2 (in fact, $\mathbb{F}_p$ for prime p), whereas Green's argument is only about MOD_2 and MOD_3.

The correlation of the Boolean functions is defined as $\mathrm{Corr}(f,g) = |\Pr_x(f(x) = 1 \mid g(x) = 1) - \Pr_x(f(x) = 1 \mid g(x) = 0)| = |\frac{\mathbb{E}_x(f(x)\cdot g(x))}{\Pr_x(g(x)=1)} - \frac{\mathbb{E}_x(f(x)\cdot(1-g(x)))}{\Pr_x(g(x)=0)}|$. We extend the definition for $f : \{0,1\}^n \to \mathbb{C}$ and $g : \{0,1\}^n \to \{0,1\}$ so that $\mathrm{Corr}(f,g) = |\frac{\mathbb{E}_x[f(x)\cdot g(x)]}{\Pr_x[g(x)=1]} - \frac{\mathbb{E}_x[f(x)\cdot(1-g(x))]}{\Pr_x[g(x)=0]}|$.

Related to our lower bound previous work Hajnal et al. [HMP$^+$93] showed the discriminator lemma, according to which upper bounded correlation of f, g implies a lower bound for circuits of the form MAJ $\circ f$ that compute g. Recall that the Boolean gate MAJ outputs 1 if and only if the number of 1s in its input is more than half of its input wires.

Lemma 1.2.6 (discriminator lemma [HMP$^+$87], restated as Lemma 4.2.2 on page 52)**.** *Let T be a circuit consisting of a majority gate over sub-circuits $C_1, C_2, \dots, C_s$ each taking n-bit inputs. Let f be the function computed by this circuit. For every $i = 1, \dots, s$, if $\mathrm{Corr}(C_i(x), f(x)) \leq \epsilon$ then $s \geq 1/\epsilon$.*

Cai et al. [CGT96] studied depth 3 circuits of the form MAJ $\circ$ MOD_m $\circ$ AND and introduced the analytic study of *exponential sums*, which is important for our work as well. Their results were for symmetric MOD functions, later generalized by Green [Gre99], whereas Bourgain [Bou05] (for odd moduli) and Green et at [GRS05] and Chattopadhyay [Cha07] finally showed an exponential size lower bound for MAJ $\circ$ $\mathrm{MOD}_m \circ \mathrm{AND}_{[O(1)]}$ computing MOD_q, when m, q are co-prime, i.e. $(m, q) = 1$.

For two layers of MOD gates, Grolmusz et al. [GT98] and Caussinus [Cau96] studied $\mathrm{MOD}_m \circ \mathrm{MOD}_r$ circuits computing the AND function and proved, for any m, r, exponential circuit size lower bounds. Barrington and Straubing [BS99] considered $\mathrm{MOD}_p \circ \mathrm{MOD}_m$ circuits and proved a exponential size lower bound for such circuits computing MOD_q, where p is a prime and $(p, q) = (m, q) = 1$. Straubing and Thèrien [ST06] introduced a finite field representation of MOD gates and simplified the previous proofs [BS99, GT98]. Chattopadhyay et al. [CGPT06] studied $\mathrm{MOD}_r \circ \mathrm{MOD}_m$ to compute MOD_q, where $(r, q) = (m, q) = 1$, for composite r. The authors proved that the fan-in of the output MOD_r gate, or any ANY gate, must be $\Omega(n)$.

1.2.3 Some obstacles in proving Smolensky's conjecture

Theorem 1.2.5 corresponds to a depth-reduction algorithm for composite moduli, which works as efficiently as the algorithm in Theorem 1.2.2 for prime moduli. There are substantial differences between these two algorithms. We note that the SYM gate, constructed by running the depth-reduction algorithm on ACC_p for prime modulus p, is a MOD_p gate; but the SYM gate constructed by Theorem 2.2.6 is much more complicated. In fact, this SYM gate is of the form: $\bmod p_1^{\beta_1} \bmod p_2^{\beta_2} \ldots \bmod p_{O(d)}^{\beta_{O(d)}} > c$ where each p_i is a prime divisor of the modulus m. One possible way in proving Smolensky's conjecture is to obtain further knowledge about this "MOD tower".

Finally, note that the works of Williams [Wil11, Wil14], provide an alternative way of using the depth-reduction algorithm for proving lower bounds.

1.3 NEXP vs ACC: Williams' program

Since proving circuit lower bounds for explicit functions is a conceptually difficult problem, researchers are also trying to understand the relation between classical time-space complexity classes and circuit complexity classes. For example, is NP contained in (non-uniform) ACC^0? This is also an extremely hard problem that is open for about 30 years. It also appears that we are currently clueless about how to tackle it. In fact, for ACC_6 circuits of depth 2, it is still open whether they can compute the whole NP. In 2011, Williams [Wil11] obtained a weaker result than NP vs ACC. Instead of NP, Williams show that NEXP, i.e. the class of nondeterministic exponential ($2^{P(n)}$ where P is a polynomial) time computable Boolean functions, does not have constant depth, polynomial size ACC circuits, i.e. $\mathrm{NEXP} \not\subseteq \mathrm{ACC}^0$.

The known relation between the circuit complexity classes and time-space complexity classes is depicted in the Hasse-like diagram of Figure 1-3. From this diagram it is evident that there is a very big gap in our knowledge about these relations.

In this section, we briefly introduce the work of Williams and show the relation between this result and our depth-reduction algorithm. Before that we discuss at a high-level the diagonalization methods, an important ingredient of Williams' program.

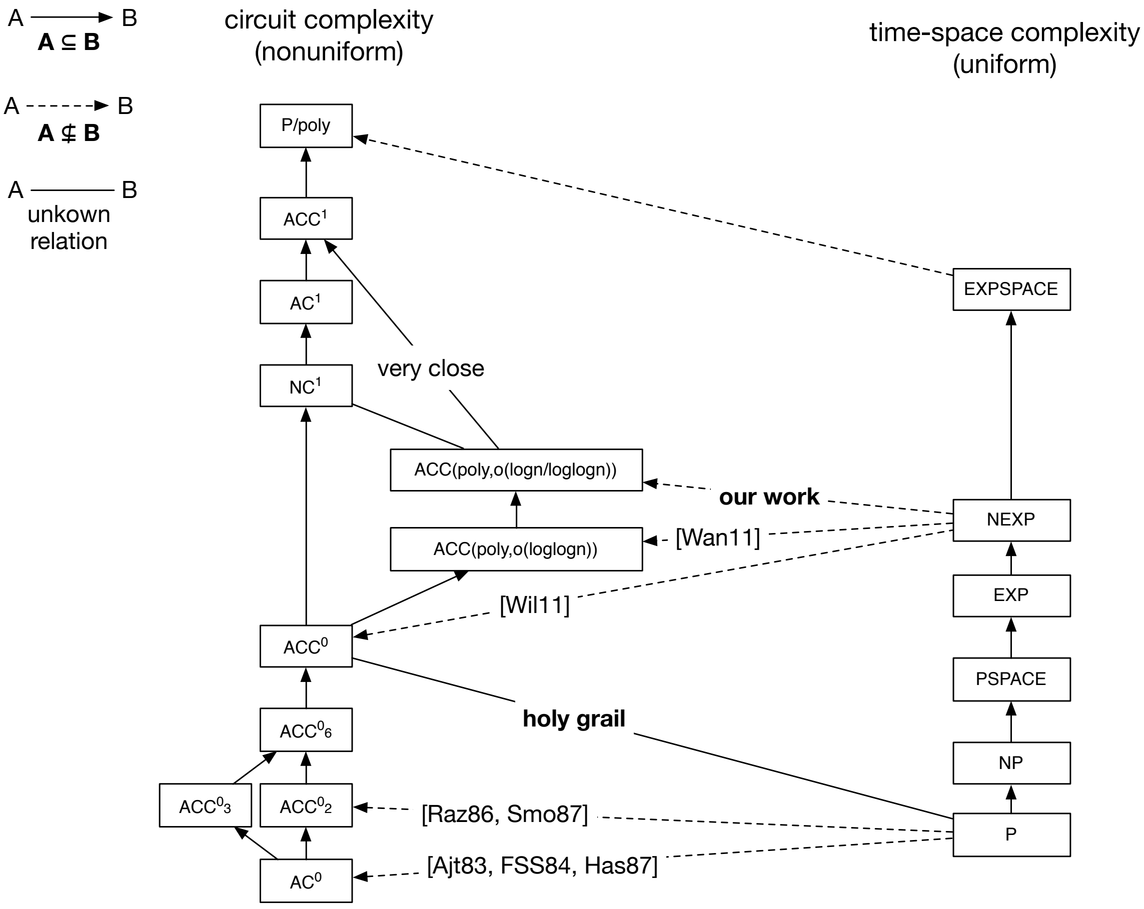

Figure 1-3: Complexity Classes

1.3.1 Self-referencing and uniformity

Can God create a stone that he can not lift? If he can not, he is not almighty, since there is still something he can not do. If he can, he is still not almighty, since he can not lift that stone. In the phrase "God create a stone that he can not lift", God is working to "create a stone that he can not lift", where the task is described using himself. The idea of using such self-referencing statements to yield contradictions has been used in different branches of mathematics, e.g. Cantor's proof of the uncountability of the real numbers, Russell's paradox, Gödel's incompleteness theorems, the undecidability of the halting problem, and time-space hierarchy.

But this method will not work when comparing circuit complexity classes. Circuit complexity classes are nonuniform classes, i.e. the circuits for computing the same function with 10 bits input and with 100 bits input can be completely different. The statement "a Boolean function f is contained in P" means that there is one algorithm that will terminate in polynomial time, and this algorithm computes f on any number of input bits. The statement "a Boolean function f is contained in ACC" means that there is an infinite family of circuits $\{C_1, C_2, \dots\}$, such that for any integer n, $C_n \in$ ACC and C_n can compute f with n bits input (with potentially completely different circuits for different input lengths). If we attempt to use self-referencing on circuit complexity, the extra description of the circuit itself will increase the input length, which obviously makes it a different circuit. Williams [Wil11, Wil14] provides a subtle way of using diagonal arguments to prove circuit lower bounds, not between circuit classes but rather between NEXP and circuit classes. We discuss this in the following section.

1.3.2 From algorithms to circuit lower bounds

Williams [Wil11, Wil14] shows lower bounds for NEXP against ACC and small variants of it. This is an important step towards proving Smolensky's conjecture 1.2.1.[1].

[1] Note that above NEXP, and in particular in EXPSPACE (or even at the 4th level of the exponential time analog of the polynomial hierarchy), we already have unconditional exponential size lower bounds for unrestricted circuits.

The author first constructs a slightly better-than-brute-force algorithm for ACC-circuits-SAT, i.e. the satisfiability problem for a given ACC circuit, based on the depth-reduction algorithm. Then, he shows that if NEXP $\subseteq$ ACC^0, ACC^0 circuits can be used as witnesses for computing functions from NTime($O(2^n)$). Here, NTime($f(n)$) stands for the family of Boolean functions that can be computed in time $f(n)$ nondeterministically. Then, using the ACC-circuit-SAT algorithm, the author can verify these witnesses a little faster than $o(2^n/n)$, which implies NTime($O(2^n)$) $\subseteq$ NTime($o(2^n/n)$). This contradicts the Cook's nondeterministic time-hierarchy [Coo73]. That means, the assumption that NEXP $\subseteq$ ACC^0 is not correct. This argument in fact is an indirect diagonal argument, since the nondeterministic time-hierarchy [Coo73] itself is proved by diagonal argument. In this argument, these ACC circuits are not used as some computational resources but as witnesses of the nondeterministic computation, and the circuit-SAT algorithm is used for verifying the witnesses.

More generally, [Wil11] initiated a program for circuit lower bounds for NEXP, where the existence of a "slightly better-than-brute-force" algorithm for $\mathfrak{C}$-SAT implies NEXP $\not\subseteq \mathfrak{C}$; see below for some restrictions on $\mathfrak{C}$.

Theorem 1.3.1 (restated as Theorem 3.1.5 at page 42)**.** *[Wil11] Let $\mathfrak{C}$ be any Boolean 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* NEXP $\not\subseteq \mathfrak{C}$*.*

A crucial step of the circuit-SAT algorithms in [Wil11, Wil14] is that the depth-reduced circuit can be of any up to slightly sub-exponential size. Therefore, the triple-exponential blowup in [BT94] implies [Wan11] an NEXP lower bound, i.e. NEXP $\not\subseteq$ ACC($2^{\log^k n}, o(\log\log n)$) for every constant $k > 0$.

1.3.3 Our improvement

The new depth-reduction (Theorem 2.2.6) yields a nearly-exponentially better depth lower bound over the previous best-known one [Wan11].

Theorem 1.3.2 (formally restated and proved as Theorem 3.1.1 on page 40)**.** NEXP $\not\subseteq$ 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 1.3.3 (formally restated as Corollary 3.1.2 on page 40)**.** *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. NC^1 is the class of functions characterized by families of circuits of constant fan-in, polynomial size, and depth $O(\log n)$. It is rather simple to see that NC^1 can be computed by circuits of polynomial size, depth $O(\log n/\log\log n)$, and unbounded fan-in. Therefore, any $\omega(1)$ improvement on the depth bound directly implies $\mathrm{NEXP} \not\subseteq \mathrm{NC}^1$. In fact, the barrier we are facing now is much stronger than just NC^1, since we allow MOD_m gates.

1.4 Interactive compression: a hybrid model

1.4.1 The model and known results

One way of strengthening Theorem 1.2 is to consider a hybrid model between circuit complexity and communication complexity, which was introduced by Chattopadhyay and Santhanam [CS12], named interactive compression. In this model, Alice is given an n-bit Boolean string x, and she wants to compute some function f on input x. But the computational power of Alice is quite limited, e.g. she can only compute ACC, so she needs the help of an almighty oracle Bob to do the computation. But Alice does not want to share too much information with Bob, so she wants to minimize the number of bits she needs to send to Bob.

Let us call these type of communication games $\mathfrak{C}$-*compress games*, when Alice only has the power of computing functions from circuit family $\mathfrak{C}$. The communication cost of this game is defined as the total number of bits sent from Alice to Bob. Sometimes, we restrict the number of rounds of the communication, i.e. in the ith round, Alice sends a string y_i that only depends on x and all of the bits received in previous rounds from Bob, and then Bob sends string z_i back to Alice. All of these y_i and z_i are computed by fixed circuits and functions. When $\mathfrak{C} = \mathrm{ACC}_p^0$ for some prime p, Chattopadhyay and Santhanam [CS12] proved a $O(\sqrt{n}/\log^{O(1)} n)$ communication lower bound for 1-round protocol. For details and definitions see Subsection 3.2 and [CS12, OS15]. This lower bound can also be proved in the same "depth-reduction and correlation bound" scheme as Theorem 1.2. Later, Oliveira and Santhanam [OS15] improved this result by removing the 1-round restriction.

The influential works of Chattopadhyay, Oliveira, and Santhanam [CS12] showed communication lower bounds for explicit functions, such as MOD_q [CS12, OS15] and the majority function MAJ [OS15]. Both of these works are based on correlation bounds between ACC_p circuits and explicit functions (see above). However, no such correlation bounds are known for composite moduli (see above) 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 on the other hand we weaken the conclusion to deriving NEXP lower bounds, which reaches the limits of current knowledge.

1.4.2 Our contribution and its relation with depth-reduction

In this dissertation, we devise another technique of proving interactive communication lower bounds for ACC circuits with composite moduli, and show the first communication lower bounds for ACC_m-compress games for any integer m. This result is stated as follows:

Theorem 1.4.1 (restated and proved as Theorem 3.2.2 on page 43)**.** *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 $\varepsilon > 0$.*

This theorem also strengthens our NEXP lower bound Theorem 1.3.2 by showing that NEXP is not only hard to compute by ACC circuits. In fact, with the help of the all-powerful Bob, Alice, who can only compute ACC circuits, still needs to send almost the whole input to Bob.

The main idea of this technique is to turn the ACC-compress game into a circuit, and use the Williams' program to prove circuits lower bounds. We briefly outline this technique below.

The first step is to turn this protocol into circuits. Since Bob is an almighty oracle, which can compute any Boolean function, we can only describe the computation of Bob as some ANY gate, i.e. any Boolean gate is possible here .

The only restriction of these ANY gates is that the fan-in of these gates is bounded by the total number of bits sent from Alice to Bob, i.e. the communication cost of this game. So if the communication cost of the k-round ACC_m-compress game is bounded by l, we can turn the communication protocol into a ACC_m circuit with k layers of $\mathrm{ANY}_{[l]}$ gates, i.e. $\mathrm{ACC}_m \circ \mathrm{ANY}_{[l]} \circ \cdots \circ \mathrm{ACC}_m \circ \mathrm{ANY}_{[l]} \circ \mathrm{ACC}_m$ circuits.

Now, the only thing left to to prove circuit lower bounds for this kind of circuits, which seems an impossible mission since there are a lot of ANY gates with only fan-in restriction in this circuits. But an interesting discovery was made by carefully analyzing our depth-reduction algorithm 2.2.6. That is, we found that the algorithm not only can compress ACC circuits, but also have the potential to compress much more complicate circuits, e.g. these circuits which describe the ACC_m-compress game. So besides the direct implications here, one can imagine there might be lots of implications to be found in the future. This generalized depth-reduction algorithm is formally stated in Section 3.2 Theorem 3.2.4.

In the same way as [Wil11, Wil14] and Theorem 1.3.2, this lower bound is proved also by constructing a fast circuit-SAT algorithm using the generalized depth-reduction algorithm.

Chapter 2

Depth reduction for composite moduli

In this chapter, we show the core result of this dissertation, the depth reduction algorithm of Theorem 1.2.5. This theorem is formally restated and proved as Theorem 2.2.6 on page 36. For the motivation and detailed comparison to previous work cf. Section 1.2.1.3 on page 12.

This chapter is organized as follows. Section 2.1 lists some preliminary notation. The same notation will be used in the subsequent chapters as well. This section also describes the pre-existing technical tools used in our depth-reduction construction. In Section 2.2.1 we show the linearization technique, which is the main technical difference between our algorithm and Yao-Beigel-Tarui [Yao90, BT94] algorithm (see also Figure 1-3). In Section 2.2.2 and Section 2.2.3 we put everything together to obtain the final depth-reduction.

2.1 Notation and existing tools

We denote by ACC^0_m the class of Boolean functions of the form $\{f_n : \{0,1\}^n \rightarrow \{0,1\}\}_{n\in\mathbb{Z}^+}$ 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, $\mathrm{ACC}^0 := \cup_{m\in\mathbb{Z}^+}\mathrm{ACC}^0_m$. 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 has 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 every other symmetric gate we may also consider the input to be an integer, as in $\mathrm{MOD}_m(x)=\mathrm{MOD}_m(||x||_1)$ as input $||x||_1$, i.e. we write $\mathrm{MOD}_m(||x||_1)$.

The $\mathrm{MOD}_m(||x||_1)$ gates, which evaluate 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 to more than one.

All operations in this dissertation 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 [Wil11].

Our construction in Section 2.2 uses a preprocessing step from [BT94]. 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 2.1.1 does this preprocessing efficiently.

Lemma 2.1.1 ([BT94, AG93, Wil14])**.** *There is an explicit construction that for every number of input bits n and modulus $m \ge 2$ and $m = \log^{O(1)} n$, 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[1].

ii. The size of the circuit is $s \cdot 2^{O(m^3 \log s_{\mathrm{AND}} \cdot \log^2 \delta_{\mathrm{AND}})} = 2^{O\left((m \log s)^3\right)}$.

iii. The fan-in of every AND gate in the circuit is $O(m^2 \log s_{\mathrm{AND}} \cdot \log \delta_{\mathrm{AND}}) = O\left((m \log s)^2\right)$.

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.

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

Remark 2.1.2. *The algorithm in the proof of Lemma 2.1.1 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*

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

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* $\leq \delta_{\mathrm{AND}}$ *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 [VV85]. Then, we 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* AND*s and leaving the rest intact. This property will be used in the interactive compression bounds in Subsection 3.2.*

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 2.1.1 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 [Tod89] for proving $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$.

Lemma 2.1.3 (Lemma 1.2.3 restated, Mod-Amplifiers [BT94], weaker forms in [Tod89, Yao90])**.** *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.

2.2 The depth-reduction

We now present the depth-reduction construction and prove Theorem 1.2.5. Theorem 1.2.5 is formally restated and proved at the end of this section as Theorem 2.2.6.

The same proof presented here, is used to obtain a stronger form of Theorem 2.2.6, which we need in the interactive compression setting of Section 3.2.

The depth-reduction is presented in three parts: (i) the linearization lemma (Lemma 2.2.1), (ii) a single step of our iterative depth-reduction construction (Lemma 2.2.5), and (iii) the use of Mod-Amplifiers (Theorem 2.2.6).

2.2.1 Linearization: eliminating products

Lemma 2.2.1 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 2.2.1 relies on the power of composite arithmetic, since a $(\mod m)$ is added even if it was not there originally.[2] When we later use Lemma 2.2.1 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 [Yao90, BT94].

Lemma 2.2.1 (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 2.2.1, the MOD_r's take inputs from the previous layer; Let the outputs of the gates of the previous layer bits be a binary vector $y \in \{0, 1\}^N$.

[2] In particular, even if we use our method instead of Allender-Hertrampf [AH94] 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).

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 2.2.2 inside the following proof).

Proof of Lemma 2.2.1. The construction of the L_i's and its analysis is shown in four parts.

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

$$\begin{aligned}\prod_{1\leq i\leq k} \mathrm{MOD}_r(L_i) &= \prod_{1\leq i\leq k}\left(\frac{1}{r}\sum_{0\leq j<q} e_r(j\cdot L_i)\right)\\ &=\frac{1}{r^k}\sum_{(j_1,\ldots,j_k)\in\mathbb{Z}_r^k} e_r\Bigg(\sum_{1\leq i\leq k}(j_iL_i)\Bigg)\end{aligned}$$

Remark 2.2.2. *We can write $\prod_{1\leq i\leq k}\mathrm{MOD}_r(L_i)$ as a sum with complex coefficients by observing that $\prod_{1\leq i\leq k}\mathrm{MOD}_r(L_i)=\sum_{1\leq i\leq s}c_i\mathrm{MOD}_r(L_i'(x))$, $c_i\in\mathbb{C}$, since for every $Y\in\mathbb{Z}^+$, $e_r(Y)=\sum_{0\leq i<r}e_r(i)\mathrm{MOD}_r(Y-i)$. 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.*

$$r^k\prod_{1\leq i\leq k}\mathrm{MOD}_r(L_i)=\sum_{(j_1,\ldots,j_k)\in\mathbb{Z}_r^k} e_r\Bigg(\sum_{1\leq i\leq k}(j_iL_i)\Bigg)$$

Since $\gcd(m,r)=1$ there exists an inverse $(r^k)^{-1}$ of r^k in the ring $\mathbb{Z}_m$.

$$\prod_{1\leq i\leq k}\mathrm{MOD}_r(L_i)=(r^k)^{-1}\sum_{(j_1,\ldots,j_k)\in\mathbb{Z}_r^k} e_r(\sum_{1\leq i\leq k}(j_iL_i)) \mod m \tag{2.1}$$

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

For every $u \in \mathbb{Z}_r$ and $v = (v_1, v_2, \ldots, v_k) \in \mathbb{Z}_r^k$, define $u \cdot v = (uv_1, uv_2, \ldots, uv_k)$, where the operation uv_i is in $\mathbb{Z}_r$.[3] 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[4] equivalence classes. These are also called the *orbits* of the group action. Let us denote each of the equivalence classes by $S_l = \mathbb{Z}_r^* \cdot (a_{l,1}, \ldots, a_{l,k})$. Regarding explicitness, in our construction each S_l can be computed by enumeration in time $r^{O(k)}$.

Then,

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

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,

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

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 $x = y$.

[3] **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 it becomes a sum of primitive roots over a scaled "line".

[4] These are less than r^k. The exact number can be computed by Burnside's Lemma; cf. [Lan02].

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

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 divisors 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. $\mu(x) = (-1)^s$, if there is no square-of-a-prime dividing x. Thus, $x = \prod_{1\le i\le s} q_i$, where q_i are the s-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.

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

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

$$\sum_{(j_1,\ldots,j_k)\in S_l} e_r(\sum_{1\leq i\leq k}(j_iL_i)) = \sum_{d|\kappa_{S_l,r}} \kappa_{S_l,r}\frac{\mu(d)}{d}\mathrm{MOD}_r(\sum_{1\leq i\leq k} d\cdot a_{l,i}\cdot L_i) \qquad (2.2)$$

Put (2.1) and (2.2) together

$$\begin{aligned}
&\prod_{1\le i\le k} \mathrm{MOD}_r(L_i)\\
&= (r^k)^{-1} \sum_{(j_1,\dots,j_k)\in\mathbb{Z}_r^k} e_r\Big(\sum_{1\le i\le k}(j_iL_i)\Big) \mod m\\
&= (r^k)^{-1} \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\dots,a_{l,k})} \left(\sum_{(j_1,\dots,j_k)\in S_l} e_r(j_iL_i))\right) \mod m\\
&= (r^k)^{-1} \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\dots,a_{l,k})} \left(\sum_{d\mid\kappa_{S_l,r}} \kappa_{S_l,r}\frac{\mu(d)}{d}\mathrm{MOD}_r\Big(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i\Big)\right) \mod m\\
&= \sum_{S_l=\mathbb{Z}_r^*\cdot(a_{l,1},\dots,a_{l,k})} \left(\sum_{d\mid\kappa_{S_l,r}} \underbrace{\left(\kappa_{S_l,r}\frac{(r^k)^{-1}\mu(d)}{d} \mod m\right)}_{\text{integer}}\mathrm{MOD}_r\Big(\sum_{1\le i\le k} d\cdot a_{l,i}\cdot L_i\Big)\right) \mod m
\end{aligned}$$

□

Remark 2.2.3 (Aside remark)**.** *Here are two aside (not used later in this dissertation) remarks.*
In depth-reduction we use Lemma 2.2.1 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* $\mathrm{MOD}_6 \circ \mathrm{ANY}_{[o(n)]} \circ \mathrm{MOD}_{35}$.[5]

Recall that every function can be written as a polynomial with $2^{O(k)}$ terms and thus we can obtain the *Generalized Linearization Lemma.*

Lemma 2.2.4 (Generalized linearization lemma)**.** *Given positive integers $m, r \in \mathbb{Z}^+$, any Boolean gate with k bit input* $\mathrm{ANY}_{[k]}$, $\gcd(m, r) = 1$ *and k positive integers $L_1, \dots, L_k$, there exist at most $s \le r^{O(k)}$ integral linear combinations $L'_1, \dots, L'_s$, i.e. $L'_i := \ell_i(L_1, \dots, L_k)$ for linear form ℓ_i, and integers $c_1, \dots, c_s \in \{0, 1, 2, \dots, m-1\}$*

[5] The generalization of Lemma 2.2.1 was suggested to us by Ryan Williams (personal communication). Our argument can be modified to work substituting in places with Fourier analytic techniques for the case of primes (personal communication with Kristoffer Hansen and Ryan Williams).

such that

$$\mathrm{ANY}_{[k]}(\mathrm{MOD}_r(L_1), \mathrm{MOD}_r(L_2), \ldots, \mathrm{MOD}_r(L_k)) = \sum_{1 \le i \le s} c_i \mathrm{MOD}_r(L_i') \mod m$$

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).

2.2.2 Inside a single iteration: using linearization & mod-amplification

Now, we show how to use the construction of Lemma 2.2.1 and the preprocessing Lemma 2.1.3 to perform a single step (described in Lemma 2.2.5) of an iterative construction (described in Lemma 2.2.6). Note that N denotes the number of input bits to a layer and n the circuit input length.

Lemma 2.2.5 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 [BT94, Tod89, Yao90]. Instead, we first use Lemma 2.2.1 to remove the AND layer. Although, this causes an even further increase in size later on we have huge overall gains.

Lemma 2.2.5. *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 2.2.1.

To apply Lemma 2.2.1 we take the $\mod m$ of the output of the $\text{AND} \circ \text{MOD}_q$ 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 $s' > \delta_{\text{SYM}}$ and $(s', q) = 1$. Then,

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

Then, by Lemma 2.2.1

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

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 2.1.3. Choose integer $k = \left\lceil \log\left(\delta_{\text{SYM}} \cdot s' \cdot \frac{q^{\delta_{\text{AND}}-1}}{q-1}\right) / \log q \right\rceil \leq (\delta_{\text{AND}} + 2\log\delta_{\text{SYM}})$. Then, $q^k > \sum_{1\leq i\leq\delta_{\text{SYM}}} \sum_{1\leq j\leq \frac{q^{\delta_{\text{AND}}-1}}{q-1}} c_{i,j}$. Then,

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

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

Step 3 Represent the formula as a SYM $\circ$ AND circuit.

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

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

Now, the original function can be represented as a circuit whose top layer is a symmetric gate $f'((\sum_{A \subseteq \{1,2,\dots,N\}, |A| \leq 2(q-1)(\delta_{\text{AND}} + 2\log \delta_{\text{SYM}})} \sum_{1 \leq j \leq b'_A} Y_{A,j}) \mod q^k)$ 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 $2(q-1)(\delta_{\text{AND}} + 2\log \delta_{\text{SYM}})$.

□

2.2.3 From single to multiple iterations

We conclude by applying Lemma 2.2.5 in each iterative step of our depth-reduction.

Theorem 2.2.6 (Theorem 1.2.5 on page 12 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* SYM ∘ AND *of size* $2^{(m \log s)^{O(\Delta(m)d)}}$ *where the fan-in of each* AND *gate is* $(m \log s)^{O(\Delta(m)d)}$*, where $\Delta(m)$ is the number of distinct prime divisors of m. More precisely, if the size of the circuit is $2^{\log^k n}$, 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 2.1.1 to construct a SYM ∘ ACC circuit with depth $2\Delta(m) \cdot d$ size $2^{m^3 \log^3 s}$ AND gate fan-in $m^2 \log^2 s$, where $\Delta(m)$ is the number of distinct prime divisors of m. Recall that each layer has only one type of gate: AND or MOD_q, where q is a prime divisor of m. We do the depth-reduction inductively from top to bottom (input level) and reduce the whole circuit into a $\mathrm{SYM}_{[2^{(m\log s)^{10\Delta(m)\cdot d}}]} \circ \mathrm{AND}_{[(m\log s)^{10\Delta(m)\cdot d}]}$ circuit. Denote by $\delta_{\mathrm{SYM},i}$ the fan-in of the symmetric gate we get from reducing the first i layers, $\delta_{\mathrm{AND},i}$ 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 SYM $\circ$ AND. Then, $\delta_{\mathrm{SYM},1} \leq 2^{(m\log s)^{1\cdot 5}}$, $\delta_{\mathrm{AND},1} \leq (m\log s)^{1\cdot 5}$

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

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 SYM $\circ$ AND circuit with $\delta_{\mathrm{SYM},i+1} = \delta_{\mathrm{SYM},i} = 2^{(m\log s)^{i\cdot 5}} \leq 2^{(m\log s)^{(i+1)\cdot 5}}$, $\delta_{\mathrm{AND},i+1} \leq (m\log s)^2 \cdot \delta_{\mathrm{AND},i} \leq (m\log s)^{(i+1)\cdot 5}$ by the 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 are obtained by compression, from the induction hypothesis form a SYM $\circ$ AND $\circ$ MOD_q circuit. We use Lemma 2.2.5 to compress. Then, $\delta_{\mathrm{SYM},i+1} \leq (2^{(m\log s)^3})^{2q(\delta_{\mathrm{AND},i}+2\log\delta_{\mathrm{SYM},i})} \leq 2^{(m\log s)^{(i+1)\cdot 5}}$, and $\delta_{\mathrm{AND},i+1} \leq 2(q-1)(\delta_{\mathrm{AND},i} + 2\log\delta_{\mathrm{SYM},i}) \leq (m\log s)^{(i+1)\cdot 5}$ by the induction hypothesis and Lemma 2.2.5.

Thus, after reducing the depth $2\Delta(m)\cdot d$ of the circuit, we get a SYM $\circ$ 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 SYM $\circ$ AND circuit to which is equivalent with the given ACC_m circuit. For ACC_6, the size and degree would be $2^{\log^{20d} s}$ and $\log^{20d} s$.

Chapter 3

Implications of depth reduction

We discuss two implications of the new depth-reduction (See Chapter 2).

Section 3.1 shows a near-exponentially better depth lower bound in Williams' program. This is a direct implication of the construction. Motivation and previous work has been discussed in detail in Section 1.3.

The second main implication is non-immediate one and is discussed in Section 3.2. This is an application of our depth-reduction construction (but not the statement of Theorem 2.2.6). Motivation and previous work has been explained in Section 1.4. 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 obstacle is that we reduce the depth of an exponentially big circuit.

Besides the above two results, there are a number of implications in the realm of immediate consequence. This includes all previous results that scale with depth-reduction.

Example of an immediate consequence Following [BT94] (p. 8, Section 6) if $\mathrm{MAJ} \in \mathrm{ACC}(2^{(\log n)^{O(1)}}, o(\log n/\log\log n))$ we conclude that TC^0 is computable by $\mathrm{ACC}(2^{(\log n)^{O(1)}}, o(\log n/\log\log n))$. This is shown by simply replacing every MAJ gate in the given TC^0 circuit by an $\mathrm{ACC}(2^{(\log n)^{O(1)}}, o(\log n/\log\log n))$ circuit. Since $\mathrm{ACC}(2^{(\log n)^{O(1)}}, o(\log n/\log\log n))$ can be compressed into a sub-exponential size $\mathrm{SYM}\circ$ AND circuit, and since a SYM gate can be computed by a depth-2, TC circuit, we conclude that TC^0 is computable by TC circuits of sub-exponential size and depth 3.

3.1 A new NEXP lower bound

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

Our improved depth-reduction (Theorem 2.2.6) yields a nearly-exponentially better depth lower bound over the previously best-known one.

Theorem 3.1.1. $\mathrm{NEXP} \not\subseteq \mathrm{ACC}(2^{\log^k n}, o(\frac{\log n}{\log\log n}))$ *for every constant k.*

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

Corollary 3.1.2. *For a fixed modulus m, and constant k, there exists 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. This has already been explained in detail in Section 1.3.3.

Proof outline of Theorem 3.1.1. 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 3.1.3 (from Theorem 2.2.6)**.** *Given an arbitrary $2^{(\log n)^{O(1)}}$-size and $o(\log n/\log\log n)$-depth* ACC *circuit, there is an explicit construction of an equivalent $2^{o(n)}$-size* $\mathrm{SYM} \circ \mathrm{AND}$ *circuit.*

Now, we state two theorems from [Wil11] that enable us to conclude Theorem 3.1.1.

Theorem 3.1.4 ([Wil11])**.** *Let $\mathfrak{C}$ be any Boolean circuit class, for which $\mathrm{OR}_{[n^{\omega(1)}]} \circ \mathfrak{C}$ can be computed by an 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 3.1.3 and Theorem 3.1.4 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.

The following Theorem 3.1.5 suffices to conclude Theorem 3.1.1.

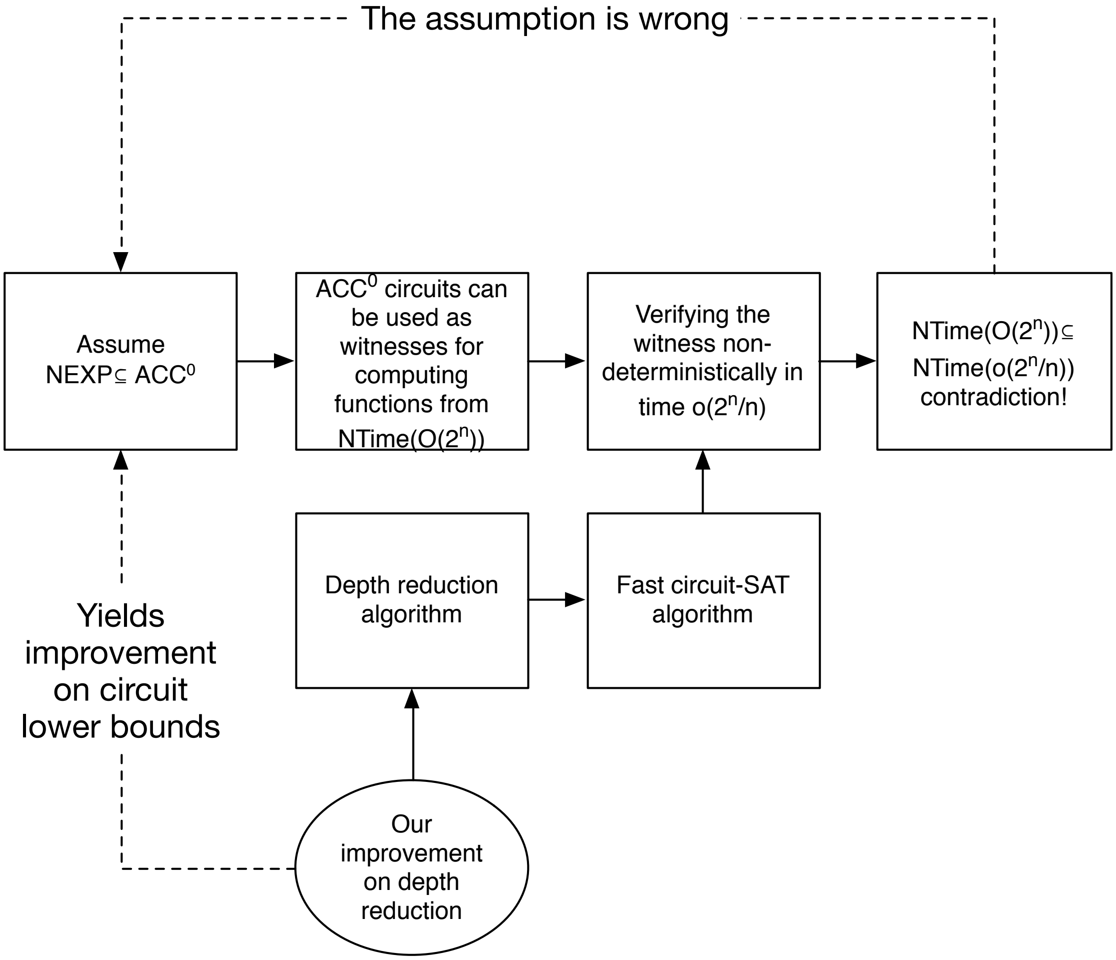
The assumption is wrong
Assume NEXP⊆ ACC^0
ACC^0 circuits can be used as witnesses for computing functions from NTime(O(2^n))
Verifying the witness non-deterministically in time o(2^n/n)
NTime(O(2^n))⊆ NTime(o(2^n/n)) contradiction!
Depth reduction algorithm
Fast circuit-SAT algorithm
Yields improvement on circuit lower bounds
Our improvement on depth reduction

Figure 3-1: Outline of the proof by Williams

Theorem 3.1.5 (see [Wil11], Theorem 1.3.1 restated). *Let $\mathfrak{C}$ be any Boolean circuit class which is closed under composition and contains* AC^0*. If $\mathfrak{C}$-SAT has a $\frac{2^n}{n^{\omega(1)}}$ running time algorithm, then* $\mathrm{NEXP} \not\subseteq \mathfrak{C}$*.*

□

3.2 Interactive compression for composites

We show the first interactive compression lower bound for general ACC circuits which was stated as Theorem 1.4.1 on page 21 and restated and proved as Theorem 3.2.2 on page 43. We begin with the formal description of this model.

3.2.1 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. [CS12].

Definition 3.2.1. *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.

3.2.2 Our interactive compression lower bound

We prove the following theorem, which is a strengthened version of the NEXP lower bound of Theorem 3.1.1.

Theorem 3.2.2. *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 $\varepsilon > 0$.*

Proof outline of Theorem 3.2.2

First, we realize the entire interaction as a circuit, replacing the bits sent back from Bob to Alice with a bounded fan-in arbitrary ANY gate.

After this, we use our depth-reduction construction to compress the circuit. By a careful analysis of the construction in Theorem 2.2.6, we can show that the same construction compresses an almost exponentially large but "highly imbalanced" circuit. The details of this strengthened theorem are in the proof of Theorem 3.2.4 below.

This way, we are able to compress this huge but of restricted form circuits and thus by [Wil14] we have that the depth-reduction implies faster-than-exhaustive-search #SAT algorithm for the circuit class described in the interactive compression procedure. This #SAT algorithm implies NEXP lower bounds.

Proof of Theorem 3.2.2

Here is how to represent the interaction as a circuit. Suppose l is the cost of this game. Lets us denote the computation of the bits sent from Bob to Alice by Boolean gates $g_1, g_2, \ldots, g_s$, where $s = 2^{\log^{O(1)} n}$ since Alice is restricted to make decisions computed by quasi-poly size ACC-circuits. Then, the fan-in of each g_i is at most l. Since we have

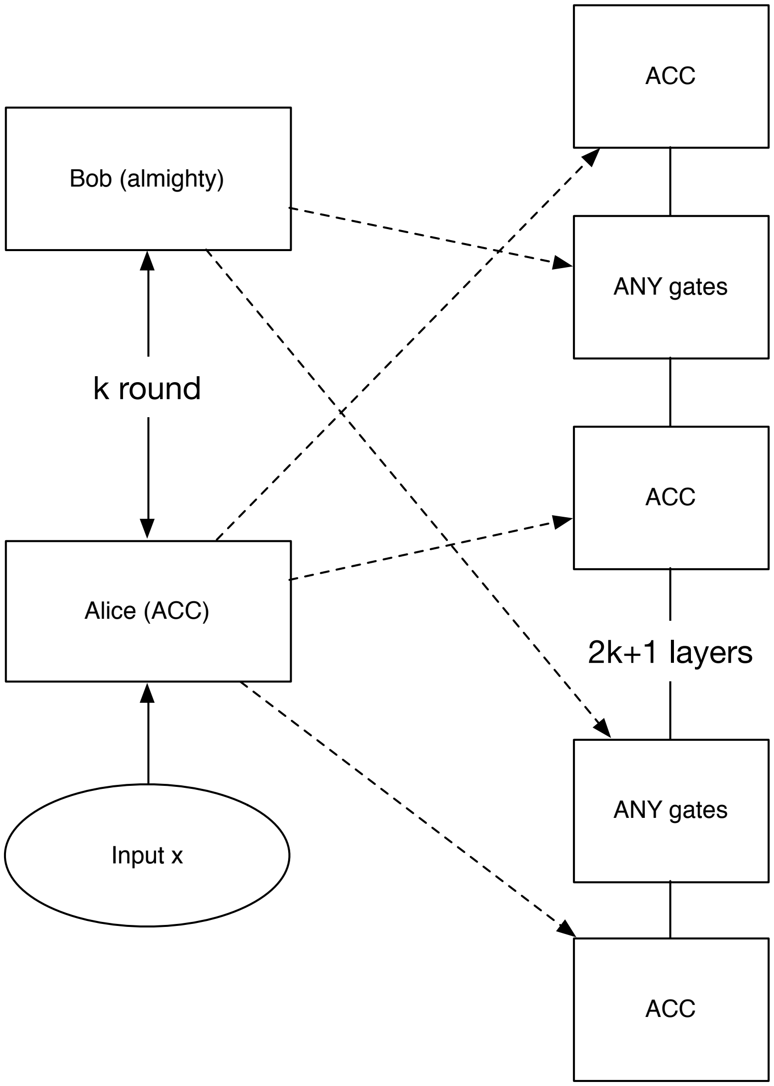

Figure 3-2: Mapping interactive compression protocol to a circuit

multiple rounds, where the result of the communication in one determines the next one, the whole computation becomes a circuit $\mathrm{ACC} \circ \mathrm{ANY} \circ \mathrm{ACC} \circ \mathrm{ANY} \cdots \circ \mathrm{ACC}$. For a k-round protocol the number of the layers of ANY gates is k and the fan-in of each ANY gate is at most l. Note that, each of the ANY gates describes the actions of (unbounded) Bob in the communication protocol.

We further modify this circuit by replacing each ANY gate by an appropriate $\mathrm{MOD}_2 \circ \mathrm{MOD}_3$ gadget. It is easy to see (and folklore) that $\mathrm{MOD}_2 \circ \mathrm{MOD}_3$ can be used to encode the truth table of any Boolean function; i.e. it is universal. For completeness we show this in Lemma 3.2.3 below. After this, we have a potentially very big ACC circuit. The issue is that the above circuit might be too large to compress (reduce its depth) using Theorem 2.2.6. After Lemma 3.2.3 we will explain how to deal with this issue.

Lemma 3.2.3. *Every* ANY *gate (Boolean function) of fan-in l can be represented by a* $\mathrm{MOD}_2 \circ \mathrm{MOD}_3$ *circuit of size $O(3^m)$. The coefficients can be computed in time $3^{O(m)}$ if you have access to the ANY gate(say, as an oracle)*

Proof. Suppose $f : \{0,1\}^m \to \{0,1\}$ is the function for the ANY gate.

We begin by representing f as a $\mathrm{MOD}_2 \circ \mathrm{AND}$ circuit. Since $\{\prod_{i,y_i=1} x_i \prod_{i,y_i=0}(1-x_i) \bigm| y \in \{0,1\}^l\}$ is the standard basis (not to be confused with the Fourier basis) of all of the Boolean functions defined on $\{0,1\}^l$, we have that $f(x) = \sum_{y\in\{0,1\}^m} f(y) \prod_{i,y_i=1} x_i \prod_{i,y_i=0}(1-$

x_i). Then,

$$\begin{aligned}
f(x) &= \mathrm{MOD}_2(1 + f(x)) \\
&= \mathrm{MOD}_2(1 + \sum_{y\in\{0,1\}^l} f(y) \prod_{i,y_i=1} x_i \prod_{i,y_i=0} (1 - x_i)) \\
&= \mathrm{MOD}_2(\sum_{y\in\{0,1\}^l} f(y) \prod_{i,y_i=1} x_i \prod_{i,y_i=0} (1 + x_i)) \\
&= \mathrm{MOD}_2(\sum_{y\in\{0,1\}^l} f(y) \sum_{z\geq y, z\in\{0,1\}^l} \prod_{i,z_i=1} x_i) \\
&= \mathrm{MOD}_2(\sum_{z\in\{0,1\}^l} (\sum_{y\in\{0,1\}^m, y\leq z} f(y)) \prod_{i,z_i=1} x_i) \\
&= \mathrm{MOD}_2(\sum_{z\in\{0,1\}^l} (\sum_{y\in\{0,1\}^m, y\leq z} f(y) \mod 2) \prod_{i,z_i=1} x_i)
\end{aligned}$$

Now, we replace the inner layer $\prod$ with AND gates and get a $\mathrm{MOD}_2 \circ \mathrm{AND}$ circuit. We conclude by representing it as a $\mathrm{MOD}_2 \circ \mathrm{MOD}_3$ circuit using Lemma 2.2.1. Since $x_i = \mathrm{MOD}_3(1 + 2x_i)$ we have the following.

$$\begin{aligned}
f(x) &= \mathrm{MOD}_2(\sum_{z\in\{0,1\}^l} (\sum_{y\in\{0,1\}^l, y\leq z} f(y) \mod 2)\mathrm{AND}_{i,z_i=1} x_i) \\
&= \mathrm{MOD}_2(\sum_{z\in\{0,1\}^l} (\sum_{y\in\{0,1\}^l, y\leq z} f(y) \mod 2)\mathrm{AND}_{i,z_i=1}\mathrm{MOD}_3(1 + 2x_i))
\end{aligned}$$

By Lemma 2.2.1 we remove the AND layer and complete the proof. □

As mentioned above, we shall now show how to do depth-reduction on the resulting circuit of size $2^{O(l)}$. It is too big for invoking Theorem 2.2.6 but we also observe that the resulting circuit is quite restricted. In particular, it is highly imbalanced, i.e. the width of each layer (the number of gates at a layer) is still very small, except the layers generated by representing the ANY gates. We introduce the following strengthened analysis of our depth-reduction, tailored for these restricted circuits.

Theorem 3.2.4. *Fix integer $m \in \mathbb{Z}^+$, a $\mathrm{SYM} \circ \mathrm{ACC}_m$ circuit of depth d, with* AND *gate fan-in $\leq s'$, and width, i.e. number of gates of layer i, w_i with $w_i > m$.*

Furthermore, in this circuit each layer either consists of : AND *gates or exclusively of* MOD_q *gates, where q is a prime divisor of m. Then, there exists an explicitly constructed equivalent circuit* $\mathrm{SYM} \circ \mathrm{AND}$ *circuit of size* $\leq 2^{s'^d \prod_{1\leq i\leq d} \log w_i}$*, and* AND *gate fan-in at most* $s'^d \prod_{1\leq i\leq d} \log w_i$.

Remark 3.2.5. *The only difference with Theorem 2.2.6 is in the calculation of the circuit size and* AND *gates fan-in in each iteration of the construction. In the proof of Theorem 2.2.6, we use the circuit size to bound the width, i.e. the number of gates of each layer. This is necessary for arbitrary* ACC *circuits. However, the circuits constructed using Lemma 3.2.3 to replace the* ANY *gates are special. The gates of this kind of circuit populate only several layers generated by* ANY *gates.*

Proof. We proceed inductively from top to bottom. The whole circuit will be compressed into a $\mathrm{SYM}_{[2^{s'^d \prod_{1\leq i\leq d} \log w_i}]} \circ \mathrm{AND}_{[s'^d \prod_{1\leq i\leq d} \log w_i]}$ circuit. Denote by norm_i the fan-in of the symmetric gate we get from compressing the first i layers, $\deg_i$ is the biggest AND gate fan-in.

The top layer of the circuit is a SYM gate, which is already a $\mathrm{SYM} \circ \mathrm{AND}$ circuit. $\mathrm{norm}_1 = w_1 \leq 2^{s' \cdot \log w_1}$, $\deg_1 = 1 \leq s' \cdot \log w_1$

Suppose that we have already compressed the first i layers into a $\mathrm{SYM} \circ \mathrm{AND}$ circuit. $\mathrm{norm}_i \leq 2^{s'^i \prod_{1\leq j\leq i} \log w_j}$, $\deg_i \leq s'^i \prod_{1\leq j\leq i} \log w_j$.

For the layer $i+1$:

Case: AND layer. Each gate of the $i+1$ layer y_t is the AND of some z from the $i+2$ layer. Then, we replace y with the products of z. We can get a $\mathrm{SYM} \circ \mathrm{AND}$ circuit with $\mathrm{norm}_{i+1} = \mathrm{norm}_i = 2^{s'^i \prod_{1\leq j\leq i} \log w_j} \leq 2^{s'^{i+1} \prod_{1\leq j\leq i+1} \log w_j}$, $\deg_{i+1} \leq s'^i \cdot \deg_i \leq s'^{i+1} \prod_{1\leq j\leq i+1} \log w_j$ by the induction hypothesis.

Case: MOD_q layer. We can think of the outputs of all the gates of layer $i+2$ are the inputs of the first $i+1$ layers of the circuit. Then the "input size" of layer $i+1$ is w_{i+1}. We can use Lemma 2.2.5 to compress the $\mathrm{SYM} \circ \mathrm{AND}$ circuit produced from the induction hypothesis and the layer $i+1$ together. The $\mathrm{norm}_{i+1} \leq (w_{i+1})^{2q(\deg_i + 2 \log \mathrm{norm}_i)} \leq 2^{s'^{i+1} \prod_{1\leq j\leq i+1} \log w_j}$, and $\deg_{i+1} \leq 2(q-1)(\deg_i + 2 \log \mathrm{norm}_i) \leq s'^{i+1} \prod_{1\leq j\leq i+1} \log w_j$ by the induction hypothesis and Lemma 2.2.5.

In the end, after reducing the depth d, we get a $\text{SYM} \circ \text{AND}$ circuit with norm at most $2^{s'^d \prod_{1 \le i \le d} \log w_i}$ and degree at most $s'^d \prod_{1 \le i \le d} \log w_i$. □

Putting together Lemma 3.2.3 and Theorem 3.2.4 obtains a construction that compresses an ACC circuit with one layer of small fan-in ANY gates.

Theorem 3.2.6. *Given a size s, depth d, ACC_m or $\text{SYM} \circ \text{ACC}_m$ circuit with k layer of ANY gates with fan-in at most l, there exists an explicit construction of an equivalent $\text{SYM}_{[2^{l^k \log^{O(d)} s}]} \circ \text{AND}_{[l^k \log^{O(d)} s]}$ circuit.*

Proof. The proof of this theorem follows closely the proof of Theorem 2.2.6, thus we are only outlining it here.

By using Lemma 2.1.1 and Remark 2.1.2 we reduce the AND gate fan-in. This way we obtain a $\text{SYM} \circ \text{ACC}_m$ circuit with k-many ANY-layers. The depth of this circuit is $O(d)$ and its size is quasi-polynomial. Each gate of this circuit is one of the following: (i) MOD_q gates, where q is a prime divisor of m, (ii) AND gate, where the fan-in of the gate is at most quasi-polynomial, (iii) ANY gate from the original circuit with fan-in at most l.

Then, using Lemma 3.2.3 we represent the ANY gates layer. Notice that the input size of each layer remains unchanged (still quasi-polynomial), but the MOD_2 gates in the layer that replaced the ANY gate can have fan-in as big as $2^{O(l)}$.

Thus, we have obtained a circuit with the following properties:

i. The depth of the circuit is $O(d)$.

ii. The "width" i.e. the number of gates of the ith layers is $w_i = 2^{\log^{O(1)} s}$ except k special layers, where $w_i = 2^{O(l)}$.

iii. The fan-in of every AND gate in the circuit is $\log^{O(1)} s$.

iv. Each MOD gate of the circuit is a MOD_q gate, where q is a prime divisor of m or a MOD_2 or MOD_3 gates. (there may be more than one type of MOD_q's inside the same circuit).

v. The circuit is layered, i.e. each layer contains gates of the same type.

We conclude by directly using Lemma 3.2.4 to do the depth-reduction. Since the "input size" of the ith layers is $w_i = 2^{\log^{O(1)} s}$ except k special layers, where $w_i = 2^{O(l)}$, the size of the output circuit will be $2^{\log^{O(d)} s \prod_{1 \le i \le d} \log w_i} = 2^{l^k \log^{O(d)} s}$ and AND gate fan-in at most $\log^{O(d)} s \prod_{1 \le i \le d} \log w_i = l^k \log^{O(d)} s$ □

Using the above depth-reduction construction and by following the same argument of [Wil14], we obtain a #SAT algorithm for the circuit class corresponding to the ACC-compression game.

Corollary 3.2.7. *Let* $\mathrm{C}_{\mathsf{inter}}$ *be the circuit class* $\mathrm{C}_{\mathsf{inter}} = \mathrm{ACC} \circ \mathrm{ANY}_{[l]} \circ \mathrm{ACC} \circ \mathrm{ANY}_{[l]} \circ \cdots \circ \mathrm{ACC}$*, with k-many layers of* ANY *gates,* $l \le n^{\frac{1}{k} - \varepsilon}$*, the circuit size is* $2^{(\log n)^{O(1)}}$ *size, and the depth is* $o(\frac{\log n}{\log \log n})$*. Then,* $\#\mathrm{C}_{\mathsf{inter}}$*-SAT can be solved in* $2^{n - \log^c n}$ *time for any constant c.*

Together with the following theorem from [Wil14]

Theorem 3.2.8 ([Wil14])**.** *Let* $\mathfrak{C}$ *be any Boolean circuit class which closed under* AND*, i.e.* $C_1 \in \mathfrak{C}$ *and* $C_2 \in \mathfrak{C}$ *implies* $C_1 \wedge C_2 \in \mathfrak{C}$*, and contains* AC^0*.For any constant c, if* $\#\mathfrak{C}$*-SAT has a* $2^{n - \log^c n}$ *running time algorithm, then* $\mathrm{NEXP} \not\subseteq \mathfrak{C}$*.*

we conclude that $\mathrm{NEXP} \not\subseteq \mathrm{C}_{\mathsf{inter}}$, which in turn implies Theorem 3.2.2.

Chapter 4

Limits of the correlation method

We prove a $2^{-O\left(\frac{n}{d(n)}\right)}$ lower bound on the correlation of $\mathrm{MOD}_m \circ \mathrm{AND}_{d(n)}$ and MOD_r, where m, r are positive integers. This is the first non-trivial lower bound on the correlation of such functions for arbitrary m, r. Motivation and detailed comparison with previous work has been discussed in Section 1.2.2

Section 4.1 outlines our technique. Section 4.2 lists the relevant preexisting tools together with the necessary additional notation. Finally, in Section 4.3 we state the technical result and give its proof.

4.1 Our technique

Our goal is to lower bound the correlation between MOD_r and any circuit $\mathcal{C}_{\text{simple}}$ with a single layer of MOD_m. We prove this in two steps. In the first step we obtain a correlation *upper bound* but for more complicated circuits $\mathcal{C}_{\text{multi-layer}}$, which in particular includes circuits with two MOD layers. This correlation upper bound implies a circuit size *lower bound* for $\mathcal{C}_{\text{multi-layer}}$. In the second step we do a reduction to obtain the *lower bound* on the correlation of a specific $\mathcal{C}_{\text{simple}}$ and MOD_r.

There is considerable success in using correlation upper bounds in obtaining circuit lower bounds. In our argument we need to lower bound the size of circuits of the form $\mathrm{MAJ} \circ \mathrm{ANY}_{[o(n)]} \circ \mathrm{AND} \circ \mathrm{MOD}_r \circ \mathrm{AND}_{[d(n)]}$, for which no previous lower bounds were known.

4.2 Additional notations and prerequisites for our correlation bound

Most of the notation we used in this chapter is introduced in Section 1.2.2 and 2.1. Now, let us state an observation we made, which is repeatedly used later on.

Observation 4.2.1 (sub-additivity). *Let functions $f_1, f_2 : \{0,1\}^n \to \mathbb{C}$ and let g be a Boolean function. Then,* $\mathrm{Corr}(f_1+f_2, g) \leq \mathrm{Corr}(f_1, g)+\mathrm{Corr}(f_2, g)$ *and* $\mathrm{Corr}(c\cdot f, g) = |c| \cdot \mathrm{Corr}(f, g)$*, for constant $c \in \mathbb{C}$.*

The main tool for proving MAJ ∘ ANY circuit lower bounds is the following lemma [HMP+87]. In fact, this lemma applies not only to MAJ but to any threshold gate.

Lemma 4.2.2 (discriminator lemma [HMP+87], Lemma 1.2.6 restated). *Let T be a circuit consisting of a majority gate over sub-circuits $C_1, C_2, \ldots, C_s$ each taking n-bit inputs. Let f be the function computed by this circuit. If* $\mathrm{Corr}(C_i(x), f(x)) \leq \epsilon$ *for each $i = 1, \ldots, s$, then $s \geq 1/\epsilon$.*

We use the above lemma together with elementary analytic techniques. The analytic machinery is explicit in the statement of the following Lemma 4.2.3 .

Lemma 4.2.3 (see [GRS05]). *For any $m, q, k \in \mathbb{Z}^+$, if $(m, q) = 1$, P is a polynomial function with integer coefficients,* $\deg(P) = O(1)$*, and $x \in \{0,1\}^n$, then*

$$\mathrm{Corr}(e_m(P(x)), \mathrm{MOD}_q(||x||_1) \leq 2^{-\Omega(n)}$$

We represent functions $f : \{0,1\}^n \to \{0,1\}$ as $f(x) = \sum_{S\subseteq\{1,2,\ldots,n\}} \alpha_S \prod_{x_i\in S} x_i$. This representation is unique since the functions $\{\prod_{i\in S} x_i | S \subseteq \{1, 2, \ldots, n\}\}$ form a function basis[1] for $\{0,1\}^n \to \mathbb{C}$. These basis functions are not to be confused with the Fourier basis, which consists of the characters written multiplicatively ($\{-1,1\}^n \to$

[1] Since $\prod_{i\in S} x_i \prod_{i\notin s}(1 - x_i)$ are easily shown to be orthogonal and the dimension of the function space is 2^n.

$\{-1,1\}$). We also introduce the definition of $\text{norm}(f) := \sum_S |\alpha_S|$, which is particularly useful for our purposes.

4.3 Our correlation result: statements and proofs

Our main results are Theorem 4.3.1, which states the circuit lower bound, and Theorem 4.3.2, which states the correlation lower bound. Note that Theorem 4.3.1 is used to show Theorem 4.3.2.

To simplify expression we represent a family of functions $\{g_m\}_m$ by one $g \in \{g_m\}_m$.

Theorem 4.3.1. *Let n be the input length to circuits and $\deg_g = o(n)$. Fix arbitrary $g : \{0,1\}^{\deg_g} \to \{0,1\}$ and $m, q \in \mathbb{Z}^+$, where $(m,q) = 1$. If a $\text{MAJ} \circ g \circ \text{AND} \circ \text{MOD}_m \circ \text{AND}_{[O(1)]}$ circuit computes MOD_q, then the fan-in of the MAJ gate on the top is $2^{\Omega(n)}$.*

Theorem 4.3.2. *For every $d \in \mathbb{Z}^+$ and every $m, q \in \mathbb{Z}^+, (m,q) = 1$ there exists a degree d polynomial P such that $\text{Corr}(\text{MOD}_m(P(x)), \text{MOD}_q(||x||_1)) \geq 2^{-O(\frac{n}{d})}$.*

4.3.1 Proof of Theorem 4.3.1: via a correlation upper bound

First, the sub-additive properties of correlation (Observation 4.2.1) yield the following lemma.

Lemma 4.3.3 (bounded correlation amplifier). *For every $d, m, q \in \mathbb{Z}^+, (m,q) = 1$ and every $g : \{0,1\}^{\deg_g} \to \{0,1\}$ and polynomial functions $P_i(x)$, $x \in \{0,1\}^n$, whose degrees are $\deg(P_i(x)) \leq d$ we have*

$$\begin{aligned}
&\text{Corr}(g(\text{MOD}_m(P_1(x)), \text{MOD}_m(P_2(x)), \ldots, \text{MOD}_m(P_{\deg_g}(x))), \text{MOD}_q(||x||_1)) \\
&\qquad \leq \text{norm}(g) \cdot \max_{P(x) \in \mathbb{Z}[x], \deg(P) \leq d} (\text{Corr}(e_m(P(x)), \text{MOD}_q(||x||_1)))
\end{aligned}$$

In particular, for $P_i(x) = O(1)$ we have

$$\text{Corr}(g(\text{MOD}_m(P_1(x)), \text{MOD}_m(P_2(x)), \ldots, \text{MOD}_m(P_{\deg_g}(x))), \text{MOD}_q(||x||_1)) \leq \text{norm}(g) \cdot 2^{-\Omega(n)}$$

Proof. Let $y_i = \mathrm{MOD}_m(P_i(x))$ and $y = (y_1, y_2, \ldots, y_{\deg_g})$ the input to g. Now, let $g(y) = \sum_{S \subseteq \{1,\ldots,\deg_g\}} \alpha_S \prod_{i \in S} y_i$. Therefore we have the following.

$$\begin{aligned}
&\mathrm{Corr}(g(\mathrm{MOD}_m(P_1(x)), \mathrm{MOD}_m(P_2(x)), \mathrm{MOD}_m(P_3(x)), \ldots, \mathrm{MOD}_m(P_{\deg_g}(x))), \mathrm{MOD}_q(||x||_1))\\
&= \mathrm{Corr}(g(y), \mathrm{MOD}_q(||x||_1))\\
&= \mathrm{Corr}(\sum_{S \subseteq \{1,\ldots,\deg_g\}} \alpha_S \prod_{i \in S} y_i, \mathrm{MOD}_q(||x||_1))\\
&\leq \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \mathrm{Corr}(\prod_{i \in S} y_i, \mathrm{MOD}_q(||x||_1)) \qquad \text{(by Observation 4.2.1)}\\
&= \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \mathrm{Corr}(\prod_{i \in S} \mathrm{MOD}_m(P_i(x)), \mathrm{MOD}_q(||x||_1))
\end{aligned}$$

$$\begin{aligned}
&= \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \mathrm{Corr}(\prod_{i \in S} (\frac{1}{m} \sum_{0 \leq j \leq m-1} e_m(j \cdot P_i(x))), \mathrm{MOD}_q(||x||_1))\\
&= \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \mathrm{Corr}(\frac{1}{m^{|S|}} \sum_{\substack{i_1 \cdots i_{|S|} \in S \\ 0 \leq j_{i_1} \cdots j_{i_{|S|}} < m}} e_m(j_{i_1} \cdot P_{i_1}(x) + \cdots + j_{i_{|S|}} \cdot P_{i_{|S|}}(x))), \mathrm{MOD}_q(||x||_1))\\
&\leq \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \frac{1}{m^{|S|}} \sum_{\substack{i_1 \cdots i_{|S|} \in S \\ 0 \leq j_{i_1} \cdots j_{i_{|S|}} < m}} \mathrm{Corr}(e_m(j_{i_1} \cdot P_{i_1}(x) + \cdots + j_{i_{|S|}} \cdot P_{i_{|S|}}(x))), \mathrm{MOD}_q(||x||_1))\\
&\qquad \text{(by Observation 4.2.1)}\\
&\leq \sum_{S \subseteq \{1,\ldots,\deg_g\}} |\alpha_S| \cdot \max_{\substack{P(x) \in \mathbb{Z}[x] \\ \deg(P) \leq d}} (\mathrm{Corr}(e_m(P(x)), \mathrm{MOD}_q(||x||_1)))\\
&\qquad \text{(because } \deg(j_{i_1} \cdot P_{i_1}(x) + \cdots + j_{i_{|S|}} \cdot P_{i_{|S|}}(x)) \leq d\text{)}\\
&= \mathrm{norm}(g) \cdot \max_{\substack{P(x) \in \mathbb{Z}[x] \\ \deg(P) \leq d}} (\mathrm{Corr}(e_m(P(x)), \mathrm{MOD}_q(||x||_1)))
\end{aligned}$$

The second part of the statement follows by Lemma 4.2.3. □

The above lemma shows the relation between correlation bounds and norm bounds. Now, we show a norm bound, which together with Lemma 4.3.3 concludes Theorem 4.3.5 below.

Lemma 4.3.4. *For every* $g : \{0,1\}^{\deg_g} \to \{0,1\}$ *we have* $\mathrm{norm}(g) \leq 3^{\deg_g}$.

Proof. We proceed by induction on $\deg_g$. If $\deg_g = 0$ then $g = 0$ or $g = 1$, that is $norm(g) = 0$ or 1. Suppose the predicate holds for $\deg_g \leq k$. For $\deg_g = k + 1$ let the polynomial representation of g be $g(x_1, x_2, \ldots, x_{k+1}) = P_1(x_1, x_2, \ldots, x_k) + x_{k+1} \cdot P_2(x_1, , \ldots, x_k)$, i.e. $g|_{x_{k+1}=0} = P_1$, $g|_{x_{k+1}=1} = P_1 + P_2$. Then, $P_1 = g|_{x_{k+1}=0}$ and $P_2 = g|_{x_{k+1}=1} - g|_{x_{k+1}=0}$. Since $g|_{x_{k+1}=1}$ and $g|_{x_{k+1}=0}$ are Boolean function on k variables, by the induction hypotehsis we have $\text{norm}(g|_{x_{k+1}=1}) \leq 3^k$ and $\text{norm}(g|_{x_{k+1}=0}) \leq 3^k$. Then, $\text{norm}(g) = \text{norm}(g|_{x_{k+1}=0} + x_{k+1} \cdot (g|_{x_{k+1}=1} - g|_{x_{k+1}=0})) \leq 3 \cdot 3^k = 3^{k+1}$. □

Theorem 4.3.5. *Fix arbitrary* $g : \{0,1\}^{\deg_g} \to \{0,1\}$ *where* $\deg_g = o(n)$ *and* $(m,q) = 1$. *Then, the correlation between* $g \circ \text{MOD}_m \circ \text{AND}_{[O(1)]}$ *circuit and* $\text{MOD}_q(||x||_1)$ *is* $2^{-\Omega(n)}$.

Proof. By Lemma 4.3.4 we have $\text{norm}(g) \leq 2^{\Omega(\deg_g)} = 2^{o(n)}$, and thus the above observation yields $\text{norm}(g \circ \text{AND}) \leq \text{norm}(g) \leq 2^{o(n)}$. Finally, by Lemma 4.3.3 we have that
$Corr(g \circ \text{MOD}_m \circ \text{AND}_{[O(1)]}, \text{MOD}_q(||x||_1)) \leq \text{norm}(g) \cdot 2^{-\Omega(n)} \leq 2^{-\Omega(n)}$. □

We strengthen this theorem by observing that $\text{norm}(g \circ \text{AND}) \leq \text{norm}(g)$, which holds true since $\prod_{1 \leq i \leq k} x_i$ is simply a monomial on x. Thus, Theorem 4.3.5 is strengthened for circuits of the form $g \circ \text{AND} \circ \text{MOD}_m \circ \text{AND}_{[O(1)]}$, and by Lemma 4.2.2 we immediately conclude the proof of Theorem 4.3.1.

4.3.2 Proof of Theorem 4.3.2: the correlation lower bound

We stated the lower bound of Theorem 4.3.1 in the most general form we could obtain (since it is also of independent interest). Now, we give the proof of Theorem 4.3.2, where we only need to show how to write MOD_q as a $\text{ANY} \circ \text{MOD}_m \circ \text{AND}_{[d]}$ circuit, for a function $\text{ANY} = g$ that we determine later.

Here is the main tool used to obtain Theorem 4.3.2.

Theorem 4.3.6. *For every* $d \in \mathbb{Z}^+$ *and* $m, q \in \mathbb{Z}^+, (m,q) = 1$ *there exists a degree* d *polynomial* P, *such that* $\text{Corr}(e_m(P(x)), \text{MOD}_q(||x||_1)) \geq 2^{-O\left(\frac{n}{d}\right)}$.

Proof. Let M_d be such that for every $d \in \mathbb{Z}^+$ and $m, q \in \mathbb{Z}^+, (m,q) = 1$ we have

$$\max_{P(x)\in\mathbb{Z}[x],\deg(P)\leq d}(\mathrm{Corr}(e_m(P(x)), \mathrm{MOD}_q(||x||_1)) = M_d$$

Split $\{x_1, x_2, \dots, x_n\}$ into n/d subsets $S_i = \{x_{id+1}, x_{id+2}, \dots, x_{(i+1)d}\}$ for $i = 1, 2, \dots, n/d$, where for simplicity we assume $d\big|n$. Now, use $\log q$ bits (all logarithms are of base 2) to encode the value of each $(\sum_{j\in S_i} x_j)$ mod q. Thus, using $\frac{n\log q}{d}$ bits denoted by $b_{1,1}, b_{1,2}, \dots, b_{1,\log q}, b_{2,1}, \dots, b_{\frac{n}{d},\log q}$ we can compute $\mathrm{MOD}(||x||_1)$. We define g such that $\mathrm{MOD}_q(||x||_1) = g(b_{1,1}, b_{1,2}, \dots, b_{1,\log q}, b_{2_1}, \dots, b_{\frac{n}{d},\log q})$. Since $\mathrm{MOD}_m(1-y) = y$ for any $y \in \{0,1\}$ we have $\mathrm{MOD}_q(||x||_1) = g(\mathrm{MOD}_m(1 - b_{1,1}), \mathrm{MOD}_m(1-b_{1,2}), \dots, \mathrm{MOD}_m(1-b_{\frac{n}{d},\log q}))$. Since $b_{i,j}$ is a function on variables $\{x_{id+1}, x_{id+2}, \dots, x_{(i+1)d}\}$, we can represent $1 - b_{i,j}$ as a polynomial $P_{i,j}$ on d variables and hence $\deg(P_{i,j}) \leq d$. Thus,

$$\mathrm{MOD}_q(||x||_1) = g(\mathrm{MOD}_m(P_{1,1}), \mathrm{MOD}_m(P_{1,2}), \dots, \mathrm{MOD}_m(P_{\frac{n}{d},\log q}))$$

which we use to obtain the following.

$$\begin{aligned}
&\mathrm{Corr}(\mathrm{MOD}(||x||_1), \mathrm{MOD}(||x||_1))\\
&= \mathrm{Corr}(g(\mathrm{MOD}_m(P_{1,1}), \mathrm{MOD}_m(P_{1,2}), \dots, \mathrm{MOD}_m(P_{\frac{n}{d},\log q})), \mathrm{MOD}(||x||_1))\\
&\leq \mathrm{norm}(g)M_d \leq 2^{\Omega(\frac{n}{d})}M_d \qquad \text{(by Lemma 4.3.4 -- different parameters than in Theorem 4.3.5)}
\end{aligned}$$

On the other hand, by the definition of correlation we have that

$$\mathrm{Corr}(\mathrm{MOD}(||x||_1), \mathrm{MOD}(||x||_1)) = 1$$

Thus, $1 \leq 2^{\Omega(\frac{n}{d})}M_d$ that implies $M_d \geq 2^{-O(\frac{n}{d})}$. □

Since $e_m(X)$ is a linear combination of $\mathrm{MOD}(X), \mathrm{MOD}(X-1), \dots, \mathrm{MOD}(X - m + 1)$ we conclude Theorem 4.3.2.

Proof of Theorem 4.3.2. Let P' be a polynomial of degree at most d such that

$$\mathrm{Corr}(e_m(P'(x)), \mathrm{MOD}_q(||x||_1)) \geq 2^{-O\left(\frac{n}{d}\right)}$$

. Since $e_m(P'(x)) = \sum_{0 \leq i < m} e_m(i)\mathrm{MOD}(P'(x) - i)$, by Observation 4.2.1 we have $\frac{1}{2^{O\left(\frac{n}{d}\right)}} \leq \mathrm{Corr}(e_m(P'(x)), \mathrm{MOD}_q(||x||_1)) \leq \sum_{0 \leq i < m} \mathrm{Corr}(\mathrm{MOD}(P'(x) - i), \mathrm{MOD}(||x||_1))$.

Then, there exists $0 \leq i < m$ such that $\mathrm{Corr}(\mathrm{MOD}_m(P'(X) - i), \mathrm{MOD}_q(||x||_1)) \geq \frac{2^{-O\left(\frac{n}{d}\right)}}{m} = 2^{-O\left(\frac{n}{d}\right)}$. □

Bibliography

[AG93] Eric Allender and Vivek Gore. On strong separations from ac. *Advances in Computational Complexity Theory*, 13:21, 1993.

[AH94] Eric Allender and Ulrich Hertrampf. Depth reduction for circuits of unbounded fan-in. *Information and Computation*, 112(2):217–238, 1994. (also FOCS'89).

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

[Bou05] Jean Bourgain. Estimation of certain exponential sums arising in complexity theory. *Comptes Rendus Mathematique*, 340(9):627 – 631, 2005.

[BS99] David Mix Barrington and Howard Straubing. Lower bounds for modular counting by circuits with modular gates. *computational complexity*, 8(3):258–272, 1999.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4(4):350–366, 1994. (also FOCS'91).

[Cau96] Hervé Caussinus. A note on a theorem of barrington, straubing and thérien. *Information Processing Letters*, 58(1):31–33, 1996.

[CGPT06] Arkadev Chattopadhyay, Navin Goyal, Pavel Pudlak, and Denis Therien. Lower bounds for circuits with mod_m gates. pages 709–718, 2006.

[CGT96] Jin-Yi Cai, Frederic Green, and Thomas Thierauf. On the correlation of symmetric functions. *Mathematical systems theory*, 29(3):245–258, 1996.

[Cha07] Arkadev Chattopadhyay. Discrepancy and the power of bottom fan-in in depth-three circuits. In *Foundations of Computer Science (FOCS)*, pages 449–458. IEEE, 2007.

[CL11] Arkadev Chattopadhyay and Shachar Lovett. Linear systems over finite abelian groups. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 300–308. IEEE, 2011.

[Coo73] Stephen A Cook. A hierarchy for nondeterministic time complexity. *Journal of Computer and System Sciences*, 7(4):343–353, 1973.

[CS12] Arkadev Chattopadhyay and Rahul Santhanam. Lower bounds on interactive compressibility by constant-depth circuits. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 619–628. IEEE, 2012.

[CW09] Arkadev Chattopadhyay and Avi Wigderson. Linear systems over composite moduli. pages 43–52, 2009.

[FSS84] Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[Gre99] Frederic Green. Exponential sums and circuits with a single threshold gate and mod-gates. *Theory of Computing Systems*, 32(4):453–466, 1999.

[Gre02] Frederic Green. The correlation between parity and quadratic polynomials mod 3. In *Computational Complexity, 2002. Proceedings. 17th IEEE Annual Conference on*, pages 47–54. IEEE, 2002.

[GRS05] Frederic Green, Amitabha Roy, and Howard Straubing. Bounds on an exponential sum arising in boolean circuit complexity. *Comptes Rendus Mathematique*, 341(5):279–282, 2005.

[GT98] Vince Grolmusz and Gábor Tardos. Lower bounds for (mod p-mod m) circuits. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 279–288. IEEE, 1998.

[Hås87] Johan Håstad. Computational limitations of small-depth circuits. 1987.

[HMP$^+$87] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. In *Foundations of Computer Science (FOCS)*, pages 99–110. IEEE, 1987.

[HMP$^+$93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

[Lan02] Serge Lang. Algebra (revised third edition). *Graduate Texts in Mathematics*, 1(211), 2002.

[Lup58] Oleg B Lupanov. On a method of circuit synthesis. In *Izvesitya VUZ, Radiofizika Vol. 1*, pages 120–140, 1958.

[OS15] Igor Carboni Oliveira and Rahul Santhanam. Majority is incompressible by $\mathrm{AC}^0[p]$ circuits. In *Conference on Computational Complexity (CCC)*, pages 124–157, 2015.

[Raz86] Alexander 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*, 41:333–338, 1986.

[RST15] Benjamin Rossman, Rocco A Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1030–1048. IEEE, 2015.

[Sha49] Claude Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Symposium on Theory of Computing (STOC)*, pages 77–82. ACM, 1987.

[ST06] Howard Straubing and Denis Thérien. A note on MODp-MODm circuits. *Theory of Computing Systems*, 39(5):699–706, 2006.

[Tod89] Seinosuke Toda. On the computational power of PP and $\oplus$P. In *Foundations of Computer Science (FOCS)*, pages 514–519. IEEE, 1989.

[Vio09] Emanuele Viola. *On the power of small-depth computation.* Now Publishers Inc, 2009.

[VV85] Leslie G Valiant and Vijay V Vazirani. NP is as easy as detecting unique solutions. In *Symposium on Theory of Computing (STOC)*, pages 458–463. ACM, 1985.

[Wan11] Fengming Wang. NEXP does not have non-uniform quasipolynomial-size ACC circuits of $o(\log \log n)$ depth. In *Theory and Applications of Models of Computation*, pages 164–170. Springer, 2011.

[Wil11] Ryan Williams. Non-uniform ACC Circuit Lower Bounds. In *Annual IEEE Conference on Computational Complexity*, pages 115–125, 2011.

[Wil14] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 194–202. ACM, 2014.

[Yao90] Andrew Chi-Chih Yao. On acc and threshold circuits. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 619–627. IEEE, 1990.