UNIVERSITY OF CALIFORNIA SAN DIEGO

# Connections Between and Techniques For Circuit Meta-Complexity Problems and Lower Bounds

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Kenneth Donald Hoover III

Committee in charge:

Professor Russell Impagliazzo, Chair
Professor Samuel Buss
Professor Shachar Lovett
Professor Ramamohan Paturi
Professor Jacques Verstraete

2022

The Dissertation of Kenneth Donald Hoover III is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF ALGORITHMS

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Russell Impagliazzo for all the help and advice he's given over the years. I would also like to thank our co-authors, Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, Ivan Mihajlin, and Alexander Smal. I want to give a special thanks to my labmates, Marco Carmosino, Jessica Sorrell, Rex Lei, Sam McGuire, Sasank Mouli, Anant Dhyal, and Ivan Mihajlin, for all the ideas we bounced off each other over the years. I want to thank my family, without whom I wouldn't be giving this defense today. Finally, I want to thank my fiancée, Kylyn Estoesta, for all the support and patience she's given me (especially over the past few months), and her family for being so welcoming to me here in San Diego.

Chapters 3, 4, 5, and 7, in part, are based on material as it appears in "Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for Constant-Depth Circuits and Applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

Chapter 8, in part, is based on material as it appears in "Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander V. Smal. Half-Duplex Communication Complexity. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation (ISAAC 2018)*, volume 123 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

VITA

2016 Bachelor of Science, University of Toronto

2018 Master of Science, University of California San Diego

2022 Doctor of Philosophy, University of California San Diego

ABSTRACT OF THE DISSERTATION

Connections Between and Techniques For Circuit Meta-Complexity Problems and Lower Bounds

by

Kenneth Donald Hoover III

Doctor of Philosophy in Computer Science

University of California San Diego, 2022

Professor Russell Impagliazzo, Chair

This dissertation presents some circuit complexity results and techniques. Circuit complexity is a branch of computational complexity dealing with classes of circuit families as opposed to classes of Turing Machines. Recent research has shown that there are rich connections between circuit complexity and other areas in theoretical computer science: Carmosino *et al.* construct learning algorithms from "natural" circuit lower bounds; Murray and Williams show that slightly better than brute-force SAT algorithms lead to circuit lower bounds against NP, and; Ilango, Ren, and Santhanam show that the existence of one-way functions is equivalent to hard distributions with certain properties existing for the Minimum Circuit Size Problem (MCSP).

A common theme throughout these results is the concept of "meta-algorithms," algorithms which take functions as input and attempt to either construct objects computing the function in some way (e.g. construct a circuit that well-approximates the function), or finding some computationally relevant quantity (e.g. what is the minimum size of a circuit computing the function).

This dissertation will focus on circuit complexity and MCSP for classes of low-depth circuits, particularly those of bounded fan-in. Here, we will present a lifting theorem from small-constant-depth bounded bottom fan-in circuits to larger-constant-depth bounded bottom fan-in circuits, leading to a reduction between MCSP for the corresponding classes. As part of this, we also present a new switching lemma, which may be of independent interest. We then demonstrate that MCSP for depth-2 bounded bottom fan-in circuits is NP-hard to compute, and is approximable within a factor of $O(\log N)$. After, we give a barrier result stating that "natural" reductions between MCSP for different fixed-depth circuit classes yields unexpectedly fast MCSP algorithms or new circuit lower bounds against these classes. Finally, we introduce a family of new models of communication complexity and give some upper and lower bounds in these models, with an eye to separating P from fan-in 2 $O(\log n)$-depth circuits.

# Chapter 1

# Introduction

We will begin with some historical background on the results this thesis presents. Computational complexity has two fundamental problems: lower bounding the amount of resources (e.g. time, space, randomness) needed to compute a function, and upper bounding the amount. These problems are immensely important, defining what sorts of tasks and problems we can or cannot expect computers to solve. Early on in the field, Cobham and Edmonds arrived at "using a polynomial amount of time, relative to the input size," the complexity class $\mathsf{P}$, being the threshold for what makes a problem efficient to solve [25, 32]. As for what makes a problem inefficient to solve, Hartmanis and Stearns showed that there is a hierarchy of time complexity classes, where higher classes are strict supersets of the lower classes [40]. However, the functions defined there are very artificial, and doesn't say anything about more natural problems we would like to solve. For these more natural problems, such as finding the shortest loop between a set of cities, it is easy to *verify* that we have been given a correct solution, but seemingly hard to generate solutions from the problem instances. In seminal works by Cook and Levin, a class $\mathsf{NP}$ was introduced that captured this notion of easy verification [26, 67]. It has since been the central question of complexity theory to relate $\mathsf{P}$ and $\mathsf{NP}$.

**Open Problem 1.** Is $\mathsf{P}$ equal to $\mathsf{NP}$?

The research problems above have so far been posed in terms of determining how (in)efficient a program is at computing various functions. This makes sense for designing

algorithms with step-by-step instructions, run on some sort of general-purpose computer. But to actually make use of these algorithms, we need to build such a computer, and this requires discussing *circuits* and how to implement functions using them. Circuits, in this context, are not physical circuits with resistors, capacitors, or other such components, but are instead a network of gates (logical connectives such as AND, OR, NOT, XOR, etc.) and wires connecting inputs and outputs of each gate to each other, with one of the component gates labelled the output of the circuit. Owing to their concrete nature, they have certain differences from more abstract programs. The first difference is that a circuit accepts a fixed number of input bits. Thus, in order to compute functions over arbitrary-length inputs, we are required to talk about *families of circuits* $\{C_n\}$, where $n$ is a parameter indexing each circuit by the number of inputs it takes. From a complexity standpoint, we can analyze the complexity of these circuit families by looking at how the number of gates grows as $n$ increases, which we call the *circuit size*, and by how the depth (the longest path from any of the circuit's inputs to the output gate) grows, the *circuit depth*. Another difference from the usual model of computation is the *non-uniformity*; whereas a program only has one set of instructions that it uses on all possible inputs, a circuit family can use completely different circuits for each input length. This allows sufficiently-large circuits to compute all possible binary functions, whereas there are well-known binary functions which cannot be computed by a single program no matter how complex (see [94] for an example). Finally, we know that all circuits have a non-trivial (i.e. smaller than $2^n$) upper bound on their size [71].

As a circuit analogue to $\mathsf{P}$, we can define the class $\mathsf{P/poly}$ of circuit families where the size is polynomially-bounded relative to the input length. Similar to uniform algorithms, we have a hierarchy of size classes for circuits. Unlike uniform algorithms, however, we also know that most boolean functions require exponentially-large circuit sizes to compute [92]. While it is useful to know most functions need large circuits, ultimately we want to find an explicit function that is hard. As an example, if we can find an explicit function that takes $2^{O(n)}$ time to compute, but no polynomial-sized circuit family can compute it, then using randomness is no

more powerful than time for polynomial-time computation.

**Theorem 2** (Hardness-to-Randomness [78, 58])**.** *If* $\mathsf{E} \not\subset \mathsf{P/poly}$*, then* $\mathsf{P} = \mathsf{BPP}$.

We can also reformulate the P versus NP question using our new circuit complexity classes.

**Open Problem 3.** Is NP contained in P/poly?

A great deal of work has been done in attempting to resolve this question, with larger and larger lower bounds being proven against stronger and stronger subclasses of P/poly [39, 41, 63, 68, 55, 93, 42, 97]. However, many of the techniques used to achieve these results were found to be unable to separate P/poly and NP, due to Razborov and Rudich [86]. Informally, this barrier states that any proof of super-polynomial circuit size lower bounds that uses a "natural property," i.e. a property of the function that is easy to test given the truth-table of the function and is true of many functions, implies that cryptography cannot exist. As it is widely believed that we can do cryptography, this result forms a barrier which many known techniques seem unable to cross, although there has been some recent progress in circumventing it [23, 29].

Focusing on the first part of the natural property definition, the ability to test whether a function has the property or not, we can observe an interesting kind of "meta-computational" problem emerge. We use the term meta here because we are designing an algorithm that computes some computational property of a given input function, i.e. computation to decide computational properties. These meta-computational problems have a rich history in the area; the Halting problem and Rice's theorem both explicitly concern meta-computation when given a program as input [24, 94, 87]. The first NP-complete problem, Circuit Satisfiability, is also of a meta-computational nature: given a circuit as input, decide if the circuit ever outputs TRUE on any input.

The Minimum Circuit Size problem (MCSP) is another problem in this family: given a function, represented by its truth-table, and a size $s$, decide if the function has a circuit computing

it of size at-most $s$. While the problem is in NP,[1] it has been notoriously difficult to determine whether it is NP-hard, in P, or perhaps somewhere in-between. In recent years, spurred on by Kabanets and Cai, we've seen a flurry of work studying MCSP, with many papers examining why it has been so difficult to prove hard [61, 76, 47, 9, 46, 8, 45, 89]. In the remainder of the introduction, we will be explaining what our results are, and further situate them within the existing literature.

## 1.1 Switching Lemmas

As part of the initial work in investigating NP versus P/poly, subclasses defined by restricting the circuit depth were looked at. Furst, Saxe, and Sipser showed that circuit families with a fixed constant depth could not compute the parity of $n$ bits using a polynomial number of gates [34]. A key part of their proof was showing that given a depth-$d$ circuit for $n$-bit parity, if they fixed a random subset of the bits to random constants, this fixing being called a *random restriction*, then with non-zero probability they would get a depth-$d$ circuit for $m$-bit parity, where $m \in \Omega(\sqrt{n})$ where each conjunctive normal form formula (CNF) or disjunctive normal form formula (DNF) at the bottom of the circuit has sized bounded by a constant. By doing this, they could "switch" the bottom formulas from CNFs to DNFs and vice versa, and thereby decrease the overall depth of the circuit by 1 without inflating the size too much. By repeatedly doing this, they would arrive at a "small" depth-2 circuit for parity, contradicting a known exponential lower bound from Lupanov [72]. A similar lemma can be found in [2], in the context of determing the power of first- and second-order formulas. This style of lemma was formally named as a *switching lemma* by Håstad, when improving on the prior result by giving a fully exponential lower bound on the size of parity circuits for any constant depth [41].

This version of the switching lemma has been used in multiple other results: Rossman proved an $\Omega(n^{k/4})$ lower bound on the size of constant-depth circuits for deciding if a graph

[1] Because we're given the $2^n$-bit truth table as input, given a candidate circuit we can check in linear time that it agrees with each possible input to the function.

contains a $k$-clique [88], and Agrawal, Allender, and Rudich were able to prove that problems complete using fixed-depth-circuit computable reductions are also isomorphic using such circuits [1]. However, what is more common is the use of variants of the Switching Lemma. In [54], Impagliazzo, Matthews, and Paturi gave a variant showing that applying a random restriction to a set of DNFs over the same variables results in a small number of high fan-in DNFs or all high fan-in DNFs depending on an identical smaller subset of variables. And in [91], Segerlind, Buss, and Impagliazzo introduce a switching lemma that allows for a polynomially-small fraction of the inputs to be fixed, rather than the $\Omega(1)$ fraction needed for Håstad's version. This comes at a cost of requiring an $\Omega(\sqrt{n})$ fan-in for the resulting formula, however.

Our contribution is a *block-wise* switching lemma, where the the input is broken into equally sized blocks and the restrictions fix all but a single bit in each block. Another version of a block-wise switching lemma has appeared in previous work [41, 43], but differs from ours in that it attempts to use groups that match the structure of Sipser's function, whereas our groups are meant to preserve the structure of a constituent function in a composition. This results in very different restriction distributions being used, as well as a quantitative difference in the number of unset variables and the probability of having a good restriction.

**Lemma 4.** *Let $\varphi$ be a $k$-CNF over $nl$ variables, and let the variables be grouped into $n$ arbitrary disjoint blocks of $l$ bits each. Let $\mathscr{D}$ be the uniform distribution over restrictions which leave exactly one variable in each block unset. The probability that $\varphi$ cannot be written as a $t$-DNF after applying a random restriction selected from $\mathscr{D}$ is at most $\left(\frac{8k}{l}\right)^t$.*

## 1.2 Lifting Theorems

A recurring theme in computational complexity is the differing "power" of various models of computation. As a concrete example, consider the parity function; whereas there is a simple linear time algorithm for computing it, any constant-depth circuit requires exponential size to even well-approximate it. Given this state, we could imagine finding models of computation in

which it is easy to prove lower bounds, and then attempt to construct related functions which "lift" the lower bounds into a different, stronger model. A common scheme for this is using function composition to lift. Under this scheme, a suitable "gadget" function $g$ is identified, and it is shown that for any function $f$, the complexity of $f$ in the weaker model is (approximately) equal to the complexity of $f \circ g$ in the stronger model. First shown by Raz and McKenzie [84], a number of lifting theorems can now be observed in the literature [20, 38, 82, 37, 28, 83]. The majority of these theorems use a query model as the weaker model, where computation is done by examining a single bit of the input at a time and branching depending on the value, and a communication model as the stronger model. They also often use depth as the measure for complexity, as opposed to size.

Recently, in a breakthrough result, Ilango showed that using randomized quasi-polynomial time Turing reductions,[2] MCSP for constant-depth formulas is NP-hard [50]. The main lemma used in this a type of lifting argument, in which the depth-$(d-1)$ size of a function $f$ is lifted to lower bound the depth-$d$ size of the function $f \wedge g$, where $g$ is a random function selected from a carefully designed distribution. This breaks with more standard lifting theorems in a number of ways; one is that there is a distribution over gadgets, as opposed to a single fixed choice. In fact, this distribution itself depends on the function $f$ being lifted. In addition, they are lifting between two models of the same type, in the sense that both models are formulas over unbounded fan-in AND and OR with the output gate being an OR, but with different restrictions on depth. This lifting theorem works by first showing that $g^{-1}(1)$ is covered redundantly by the subformulas under the output gate, and then using this redundancy to give a smaller approximation for $g$ than is assumed possible. Such an approach does seem to require formulas as the model, leveraging the fact that subformulas must be disjoint from each other to achieve the contradiction above; subcircuits can reuse most of the gates beneath the top gate of each subcircuit, in effect amortizing the complexity of computing a "hard core" over multiple subcircuits.

[2] Turing reductions are stronger than the many-one reductions used as the standard definition for NP-hardness. As one example, under linear-time Turing reductions NP = coNP.

Our lifting theorem is more in line with previous lifting theorems, using the composition of an arbitrary $f$ with the parity function to lift the size complexity of $f$ for *bounded bottom fan-in* depth-$d$ circuits to bounded bottom fan-in depth-$(d+1)$ circuits. Here, both the bottom fan-in bounds and the number of input bits for the parity gadget are logarithmic in the depth-$d$ size of $f$. Such bounded bottom fan-in models have been studied before in the literature, with fixed-constant bounds appearing in [16, 80, 81, 57] in the context of both finding circuit lower bounds and designing algorithms, and logarithmic bounds similar to our own appearing in [41].

**Theorem 5.** *Let $f$ be a function with bounded bottom fan-in depth-$d$ complexity $s$. Then $f$ composed with the parity function over $\Theta(\log s)$ bits has bounded bottom fan-in depth-$(d+1)$ complexity $s^{\Theta(1)}$.*

The two main ingredients of this theorem are our switching lemma, which we use to transform a depth-$(d+1)$ circuit for the composition into a depth-$d$ circuit computing a function that can be projected into $f$, and what we call "clever brute force." What we mean by this is the following: if we use an optimum CNF or DNF for parity, and wire the outputs of each parity group into the inputs for $f$, we would either have too large a depth or the bottom fan-in would be too large for our switching-based approach to work. So instead, we view the single parity as a composition of two separate parities, and distribute the number of inputs between them such that the overall bottom fan-in meets our bound (the clever part). Then, we use distributivity to take what would be a depth-$(d+3)$ circuit and instead collapse two layers, giving us a depth-$(d+1)$ circuit at some cost to the size (the brute force part).

## 1.3 Reductions between MCSP Variants

When we demonstrate a lifting theorem, we can view it through two lenses. The standard lens is that of lower bounds; we can lift a lower bound in one model up into lower bounds in another. The alternative lens is viewing the lifting as a reduction. As an example, consider decision tree to communication complexity lifting, where we show that the decision tree complexity of a

function $f$ is roughly equivalent to the communication complexity of a composed function $f \circ g$ [20]. If we view it through the alternative lens of a reduction, we can see that we are reducing "finding the decision tree complexity of $f$" to "finding the communication complexity of $f \circ g$." Considering decision tree complexity can be approximated to within a polynomial factor in polynomial time given the full truth table of $f$,[3] this reduction on its own isn't very interesting.

However, the lifting theorem present in [50] is much more interesting to us. In particular, we know that computing the minimum DNF size of function is NP-complete [73], while hardness for other constant depth formula classes is unknown. Thus such lifting theorems actually do present us with new hardness results! In a similar vein, we transform our lifting theorem into an algorithmic reduction, and obtain a quasi-polynomial Karp reduction from approximating MCSP for small depth bounded bottom fan-in circuits to approximating MCSP for large depth bounded bottom fan-in circuits, where the approximation factor shrinks by a polynomial factor.

**Theorem 6.** *Let $k > 0$ be a sufficiently large constant. Approximating* MCSP *for depth-$d$ bounded bottom fan-in circuits to within a factor of $s^k$ can be reduced in quasi-polynomial time to approximating* MCSP *for depth-$(d+1)$ bounded bottom fan-in circuits to within a factor of $s^{\Theta(k)}$.*

We can then combine this reduction with a size versus bounded bottom fan-in trade-off, and another lift from bounded bottom fan-in depth-$d$ to unbounded bottom fan-in depth-$d$, to obtain a quasi-polynomial time reduction from approximating MCSP for depth-$d$ circuits to approximating MCSP for depth-$(d+1)$ circuits.

**Theorem 7.** *Let $0 < \alpha < \beta < \gamma < \delta < 1$ be constants, with a sufficiently large gap between $\gamma$ and $\delta$ and between $\alpha$ and $\beta$. Distinguishing between $n$-bit functions which have $2^{n^\alpha}$-sized depth-$d$ circuits and those which do not have any $2^{n^\delta}$-sized depth-$d$ circuit can be reduced in quasi-polynomial time to distinguishing between $n^{\Theta(1)}$-bit functions which have $2^{n^\beta}$-sized depth-$(d+1)$ circuits and those which do not have any $2^{n^\gamma}$-sized depth-$(d+1)$ circuit.*

[3] Block sensitivity, a measure of complexity for boolean functions, can be brute forced in time $2^{O(n)}$ on an $n$-bit function, and is polynomially related to decision tree complexity [77].

We can also consider a *tolerant* version of MCSP, where instead of finding the smallest circuit exactly computing a function $f$, we want the smallest circuit that approximates $f$ to within an error $\varepsilon$. Just as we can view MCSP as a generalization of natural properties versus worst-case complexity classes, we can view tolerant MCSP as being a generalization of natural properties versus *average-case* complexity classes, as is done in [19] to construct agnostic learning algorithms. In the tolerant case, we can use a well-known trade-off between the bottom fan-in of a circuit and the error to obtain a reduction for unbounded bottom fan-in circuits, instead of resorting to the exponential size blow-up of the size versus bottom fan-in trade-off.

**Theorem 8.** *Let $k > 0$ be a sufficiently large constant. Approximating tolerant* MCSP *for depth-$d$ circuits to within a factor of $s^k$ can be reduced in quasi-polynomial time to approximating tolerant* MCSP *for depth-$(d+1)$ circuits to within a factor of $s^{\Theta(k)}$, with a $1/\mathsf{poly}(n)$ additive loss in the tolerance.*

## 1.4 Hardness for MCSP Variants

As mentioned earlier in the introduction, determining the hardness of MCSP has been an elusive problem within the field. While we have seen some limited success in showing MCSP is hard, under various types of reductions and for smaller classes than NP [6, 79], and have some lower bounds for MCSP in subclasses of P/poly [5, 49, 36, 22], we also have a number of barrier results that either rule out certain classes of reductions for proving hardness ([8, 76, 10]) or show that hardness for MCSP leads to resolving longstanding open questions (we'll discuss this line in the next section of the introduction). In light of that, instead of attempting to show results about MCSP itself, we can think about variants to the problem.

Until very recently, the only variant with unconditional NP-hardness was DNF-MCSP, where given a function we want to find the minimum number of terms in any DNF computing the function. This was originally due to Masek [73], with simplified proofs appearing later [27, 95, 7]. Focusing on the proof in Allender *et al.* [7], it proceeds in two stages: first, a

reduction is given from 3-Partite Set Cover (closely related to 3D Matching) to DNF-MCSP for *partial* functions; then a reduction from DNF-MCSP for partial functions to DNF-MCSP for total functions. The first stage uses the fact that each term of a DNF corresponds to a projection of the boolean hypercube, and constructs a mapping from universe elements and sets to boolean vectors and projections such that the membership relation is unchanged. The second stage uses a clever technique where they introduce two new bits, used to select whether the function should accept: all non-rejecting inputs for $f$; all indeterminate inputs of even parity; all indeterminate inputs of odd parity, or; no inputs. They then argue that any DNF computing this new function must contain one term for each indeterminate input and a DNF for $f$, with the two sets of terms being disjoint.

In 2018, Hirahara, Oliveira, and Santhanam demonstrated that MCSP for OR-AND-$\text{MOD}_2$ circuits is NP-complete [45]. Such circuits consist of an output OR gate of unbounded fan-in, with unbounded fan-in AND gates underneath it, and finally unbounded fan-in $\text{MOD}_2$ gates beneath those. Their reduction follows a similar two-stage approach as Allender *et al.*, reducing from 2-approximating $r$-Bounded Set Cover to the partial function version of MCSP, and from the partial function version to the total function version. The main difference is that the reductions here are *zero-error randomized* reductions, rather than deterministic. This is resolved in a third stage, where they construct a pseudorandom generator with sufficient strength to derandomize their reductions. Here too, they use a geometric property of AND-$\text{MOD}_2$ circuits, namely that all such circuits only accept affine subspaces of $\mathbb{Z}_2^n$. Similar in spirit to Allender *et al.*, they map the elements of the universe uniformly at random onto a sufficiently large hypercube, and map the sets onto the span of the contained elements. In the second stage, they map each input $x$ onto a function $f_x(y)$, with all accepting inputs mapping to the function $y = 0$, all rejecting inputs mapping to the constant FALSE function, and all indeterminate inputs being mapped to the indicator for the linear span of a random vector set. Again, in a similar argument to Allender *et al.*, they show that each AND gate either computes $f_x(y)$ for a single indeterminate input $x$, or is part of computing the original partial function.

Both of these results leverage geometric properties of their circuit classes when viewed as subsets of the hypercube. Such leverage disappears once we consider OR-AND-OR circuits, as now the depth-2 AND-OR subcircuits form a complete class, i.e. any function can be expressed as an AND-OR circuit. As such, we are forced to examine other techniques for hardness. In [50], Ilango gives super-polynomial time lower bounds under the Exponential Time Hypothesis (ETH) for partial function MCSP,[4] and unconditional NP-hardness under quasi-polynomial time randomized reductions for depth-$k$ formula MCSP, for all constants $k$. The former result uses the Bipartite Permutation Independent Set (BPIS) problem, proved hard under the ETH in [70], and shows that given a suitably defined function, any small circuit computing the function has a canonical form from which a permutation solving the original BPIS problem can be derived. This result also carries over to the partial-function version of MFSP, the Minimum *Formula* Size Problem. A similar technique of forcing all minimum formulas into a canonical form can be seen in [51], where ETH hardness for total-function MFSP is proven. In that case, they first prove a lower bound assuming the formula is already in canonical form, and then demonstrate that any general formula embeds the canonical form after a projection that kills a small number of leaves, using a technique from [14]. For the fixed-constant-depth formula MCSP result, a lifting approach is taken instead, as outlined previously.

In Chapter 6, we demonstrate a reduction from total-function DNF-MCSP to bounded bottom fan-in total-function DNF-MCSP, with the hope of possibly using this as a starting point for hardness of higher-depth circuits. Our reduction uses a padding argument to blow up the required size in a controlled way, allowing formerly unbounded bottom fan-in circuits to be recast as bounded bottom fan-in circuits.

**Theorem 9.** DNF-MCSP *for bounded bottom fan-in circuits is* NP*-hard.*

Unfortunately for us, the gap for which we can possibly expect the DNF version to be hard is much smaller than needed, having an efficient algorithm for distinguishing functions with

[4]The Exponential Time Hypothesis, informally stated, claims that the Circuit Satisfiability problem requires deterministic time $2^{\Omega(n)}$ to compute [56, 57].

size-$s$ bounded bottom fan-in DNFs versus functions with no size-$sn$ bounded bottom fan-in DNFs.

**Theorem 10.** *Distinguishing between n-bit functions having size-s bounded bottom fan-in DNFs versus n-bit functions with no size-sn bounded bottom fan-in DNFs can be done in polynomial time.*

## 1.5 Meta-algorithms versus Circuit Lower Bounds

Consider meta-computational problems at their most abstract: an algorithm is given a function, or a circuit, or a program, and is asked to determine if some computational property is true of the given object. Informally, being able to design such an algorithm should entail a sufficient degree of understanding about the property in question. However, designing reductions to such a problem would *also* seem to require such understanding, in order to prove the reduction correct. Thus being able to make statements about meta-computational problems would seem to go hand-in-hand with our ability to prove statements regarding the underlying property. Such an intuition can be made explicit, as was done with the Natural Proofs barrier [86]. A good example of this phenomenon is the Easy Witness line of papers, which tie together circuit upper bounds for various non-deterministic time classes and the complexity of generating witnesses for problems in those classes [60, 53, 96, 97, 75, 21]. A particular highlight is the main result of [75], which shows that faster $\mathscr{C}$ Satisifiability algorithms, for any $\mathscr{C}$ a standard circuit class, yield $\mathscr{C}$-circuit lower bounds against $\mathsf{NP}$. We can also see this again in [18], in which the distinguisher-to-circuit algorithm present in the proof of the Nisan-Wigderson pseudorandom generator is leveraged to construct learning algorithms from natural properties.

Even more recently, a deep connection between the existence one-way functions (the core from which the cryptographic schemes in use today can be derived) and $\mathtt{MCSP}$ has been uncovered [69, 90, 52]. The latest in this sequence, from Ilango, Ren, and Santhanam, go beyond $\mathtt{MCSP}$, showing that one-way functions existing is in fact equivalent to showing that, for some "nice"

complexity measure $C$ and polynomial-time samplable distribution $D$ where the $C$ complexity of a function $f \in D$ is bounded by roughly $\log(1/D(f))$, approximating $C$ complexity is hard on average over $D$. Going back to natural properties, then, the existence of one-way functions would imply strong pseudorandom generators exist; these generators would rule out $\texttt{MCSP} \in \mathsf{P/poly}$, giving circuit lower bounds from a uniform average-case hardness result.

Our contribution in Chapter 7 can be viewed as an expansion of the "natural reductions imply circuit lower bounds" result in [61]. In their paper, they define the concept of a "natural" reduction to MCSP as one where the size parameter for the MCSP instance only depends on the length of the instance being reduced from. They then demonstrate in Theorem 15 how, if MCSP is NP-hard using such a reduction then there is a family of functions computable in time $2^{O(n)}$ that require super-polynomial sized circuits. We instead look at inter-reductions between MCSP variants, in particular constant-depth circuit variants, and so relax this notion of "natural" to instead allow for the size parameter of the output instance to vary depending on the length as well as the size parameter of the input instance. We can then apply a similar win-win argument to obtain new circuit lower bounds from efficient natural reductions.

**Theorem 11.** *Suppose there is a natural polytime reduction from* MCSP *for depth-$d$ circuits to* MCSP *for depth-$d'$ circuits, where $d' > d$. Then* MCSP *for depth-$d$ circuits can be solved surprisingly fast, or there is a function family computable in time $2^{O(n)}$ that does not have size-$2^{\Omega(n^{1/(d-1)})}$ depth-$d'$ circuits.*

## 1.6 Communication Complexity

In Yao's communication model of computation [98], we have two parties, Alice and Bob, who wish to compute some function $f : A \times B \to Z$. Alice is given the $A$ input, Bob is given the $B$ input, and both are assumed to have unbounded computational resources (besides knowing which input the other party received). Their task is to communicate bits to each other, one at a time, until the output of $f$ on their joint inputs is known to both parties. A *protocol* is a

binary tree with internal nodes each labelled with either a partition of $A$ or a partition of $B$ and leaves each labelled with an element of $Z$. The two parties communicate using the protocol as follows; starting at the root, check whether Alice's (Bob's) input is in the left or right partition, then traverse to the left or right child accordingly. Repeat this process until a leaf is reached, and then output the label of that leaf. Traversing to the left or right child corresponds to Alice (Bob) sending a 0 or 1 bit and both players updating their state accordingly. The *communication complexity* of a protocol is the depth of the tree, and the communication complexity of $f$ is the minimum communication complexity of any protocol computing $f$. We can extend this model to relations $R \subseteq A \times B \times Z$ and partial functions as well; for partial functions, we say a protocol computes the partial function if on all inputs for which the function is defined, the protocol outputs the correct value. Similarly, a protocol computes a relation if for every leaf, the label $z$ for the leaf and the input set $S \subseteq A \times B$ which can reach the leaf are such that $S \times \{z\} \subseteq R$.

In [63], Karchmer and Wigderson define a communication relation called the *Karchmer-Wigderson game for $f$*, $\mathtt{KW}_f$, where $f : \{0,1\}^n \to \{0,1\}$. In this relation, Alice is given an input in $x \in f^{-1}(1)$, Bob an input $y \in f^{-1}(0)$, and they wish to find an index $i \in [n]$ where $x_i \neq y_i$. As they show in their work, the communication complexity of this relation is exactly equal to the minimum formula depth needed to compute $f$. Using a closely related relation applicable to monotone functions and formulas, they manage to separate monotone $\mathsf{P}$ from logarithmic-depth fan-in-2 monotone formulas. Seeking to drop the monotone restriction and obtain a full separation between $\mathsf{P}$ and these low-depth formulas, Karchmer, Raz, and Wigderson introduce the following conjecture.

**Conjecture 12** ([62])**.** *There is some $1 \geq \varepsilon > 0$ such that for all functions $f$, the communication complexity of $\mathtt{KW}_{f \circ f}$ is at least $1 + \varepsilon$ times the complexity of $\mathtt{KW}_f$.*

If this conjecture were true, then we would have our separation. To see why, consider the hardest function $f^*$ on $\log n$ variables requiring formula depth $\Omega(\log n)$. If we compose $f^*$ with itself $\log n / \log\log n$ times, the resulting function would have $n$ variables, and by the conjecture

above would require formulas of depth $\log^{1+\varepsilon'} n / \log\log n \in \omega(\log n)$.

Instead of directly attempting to study the complexity of compositions for concrete functions, we can examine the complexity of the *Universal Composition Relation* and *Iterated Multiplexor function*. The former is a generalization of the KW game for compositions where each player is given a $k$-ary tree of depth $d$, with each node labeled 0 or 1, with the property that if the label given to each player for a particular internal node is different, then some child of that node is given different labels for each party's tree. Alice's root is labelled 1, Bob's root is labelled 0, and they wish to find a leaf which is labelled differently in each tree. Formalized in [63], lower bounds that are almost tight for the regimes of $k$ and $d$ we are concerned with were proven independently by Edmonds *et al.* and Håstad and Wigderson [33, 44]. The Iterated Multiplexor takes as input a $k^d$-bit vector and a $k$-ary depth-$d$ tree where the leaves are labelled with variable names and the internal nodes are labelled with $k$-ary functions, and outputs the natural composition computation indicated by the tree. If we restrict all internal nodes at the same depth to have the same function label, the Iterated Multiplexor can then be viewed as a single function through which we can study the behaviour of all function composition with respect to depth.

It was hoped that techniques used in proving the lower bounds for the Universal Composition Relation would be useful in demonstrating lower bounds for the Iterated Multiplexor, which would then give us P formula depth lower bounds. However, the techniques used in lower bounding the Universal Composition Relation rely on being able to maintain symmetry between the possible sets of Alice and Bob inputs as the protocol proceeds, whereas for the KW game for the Iterated Multiplexor the two parties have already broken symmetry completely. Moreover, the Iterated Multiplexor KW game is difficult to study due to how Alice and Bob may want to change who gets to speak at a given protocol node depending on what a particular function in the Multiplexor tree; such behaviour is not allowed in the standard communication model. Several results since have begun bridging this gap between known lower bounds for Universal Composition and the lack of such results for Iterated Multiplexor [31, 65, 35, 74].

In Chapter 8, we introduce a new model of communication through which we hope to better study the complexity of Iterated Multiplexor. In this *half-duplex* model, Alice and Bob are allowed to simultaneously send or simultaneously receive bits, in addition to one sending the other a bit. In addition to this new ability for the two players to conflict, it also allows for them to, e.g., change who is the speaker and who is the receiver based on the function label given to a Multiplexor tree node. Introducing alternative models of communication to facilitate better analysis of an existing problem has prior examples in the literature [59, 30, 11, 64]. In particular, Impagliazzo and Williams [59] use their model, defined somewhat similarly to ours, in order to obtain lower bounds for the communication analogue of $\mathsf{P}^{\mathsf{NP}}$, although they focus on the complexities of protocols where choosing to send, receive, or do neither all have different costs. We introduce three variants of the half-duplex model according to what happens when both parties choose to receive bits, termed a *silent* round, and give some upper and lower bounds of various communication problems within these models. Additionally, for one such variant we give an upper bound on the amount of information the two parties can exchange in a single round, and use it to derive lower bounds for parity that match the known lower bounds of standard communication.

# Chapter 2

# Preliminaries

We will follow the convention that capitalized Greek letters, e.g. $\Lambda$, $\Gamma$, refer to circuit classes; lowercase Greek letters, e.g. $\alpha$, $\beta$, refer to functions $\mathbb{N} \to \mathbb{N}$, and; constants are lowercase Latin letters.

## 2.1 General

**Definition 13** (Time Complexity)**.** Given a language $L \subseteq \{0,1\}^*$, we say $L \in \mathsf{TIME}[t(n)]$ if there is an algorithm running in asymptotic time $O(t(n))$ that accepts an input $x$ if and only if $x \in L$.

**Definition 14** (Standard Time Complexity Classes)**.** The class $\mathsf{P}$ is defined as $\mathsf{P} = \bigcup_{k\in\mathbb{N}} \mathsf{TIME}[n^k]$. The class $\mathsf{QuasiP}$ is defined as $\mathsf{QuasiP} = \bigcup_{k\in\mathbb{N}} \mathsf{TIME}[2^{\log^k(n)}]$. The class $\mathsf{E}$ is defined as $\mathsf{E} = \bigcup_{k\in\mathbb{N}} \mathsf{TIME}[2^{kn}]$.

**Definition 15** (Truth Table)**.** The *truth table* of a function $f : \{0,1\}^n \to \{0,1\}$ is the $2^n$-bit string $z$ where $z_i$ is equal to $f$ on the binary representation of $i$.

**Definition 16** (Composition)**.** Given two boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, we say that the *composition* of $f$ with $g$ is the function $f \circ g : \{0,1\}^{nm} \to \{0,1\}$ defined by breaking the inputs into $n$ blocks $[\vec{x_1}, \ldots, \vec{x_n}]$ of $m$ bits each, computing $g$ for each block, and then outputting $f(g(\vec{x_1}), \ldots, g(\vec{x_n}))$.

**Definition 17** (Promise Problems)**.** A *promise problem* is defined by two disjoint sets $Y, N \subset$

$\{0,1\}^*$. An algorithm *computes* a promise problem if for every $x_Y \in Y$ the algorithm accepts $x_Y$ and for every $x_N \in N$ it rejects $x_N$.

We will sometimes call the sets $Y$ and $N$ the YES and NO sets, respectively.

**Definition 18** (Many-One Reductions)**.** A $t(n)$*-many-one reduction from a language A to a language B* is a function $f : \{0,1\}^* \to \{0,1\}^*$ computable in time $O(t(n))$ such that for any $x$, $x \in A \Leftrightarrow f(x) \in B$. We denote this as $A \leq_m^{t(n)} B$. We will use $\leq_m^{\mathsf{poly}}$ to allow any polynomial be used for $t$, and $\leq_m^{\mathsf{qpoly}}$ for any quasi-polynomial.

**Definition 19** (Hardness)**.** For any standard complexity class $\mathsf{C} \supseteq \mathsf{P}$, we say that a language $L$ is $\mathsf{C}$*-hard* if for every language $L' \in \mathsf{C}$, $L' \leq_m^{\mathsf{poly}} L$. We say that $L$ is $\mathsf{C}$*-complete* if $L \in \mathsf{C}$ as well.

**Definition 20** (Parity)**.** The *n-bit parity function*, which we denote as $\oplus_n : \{0,1\}^n \to \{0,1\}$, is defined as outputting 1 if and only if the number of 1s in the input is odd.

We will omit $n$ when it is clear from context what the value is.

## 2.2 Circuits

**Definition 21.** A *circuit over n-bit inputs* is a labelled directed acyclic graph, where all source vertices have labels from $\{x_i, \neg x_i \mid i \in [n]\} \cup \{0,1\}$, all non-source vertices $v$ have as a label a function $f_v : \{0,1\}^{\text{in-deg}(v)} \to \{0,1\}$, and there is a single sink vertex designated as the output. For our purposes, we will suppose that for each $i$, there is a source labelled with $x_i$ or $\neg x_i$, i.e. all circuits have at least $n$ vertices. We define the *size* of a circuit to be the number of vertices in the underlying graph. The *depth* of a vertex $v$ (also called a *gate*) is the edge-length of the longest path from any source that can reach $v$ to $v$. The depth of a circuit is the depth of the output gate.

Computation using a circuit given an input $x$ works by first replacing each source label $x_i$ (or $\neg x_i$) with the value (or negation) of the input at position $i$. Then, for each non-source $v$ with a function label, if all in-neighbours have boolean valued labels, we compute $f_v$ on those values

and replace $f_v$ with the resulting output. When the output vertex is given a boolean value, we take that as the result of the entire circuit.

As circuits are concrete objects that only ever accept $n$-bit inputs, in order to talk about circuits "computing" functions/languages over arbitrary input lengths we refer to circuit families instead.

**Definition 22.** A *circuit family* is a sequence of circuits $\mathscr{C} = \{C_n\}_{n\in\mathbb{N}}$ where each $C_n$ takes $n$ bits as input. We say the size and depth of a family are the asymptotic upper bounds of the size and depth of $C_n$ as $n$ goes to infinity.

We also have a subset of circuits where reusing computation from vertices within the circuit is not allowed.

**Definition 23** (Formulas)**.** A *formula* is a circuit where the underlying undirected graph is a tree. The *formula size* of such a circuit is the number of source vertices.

If we do not otherwise specify, we assume all formulas have function labels taken from (are over the basis) $\{\wedge : \{0,1\}^2 \to \{0,1\}, \vee : \{0,1\}^2 \to \{0,1\}, \neg : \{0,1\} \to \{0,1\}\}$.

**Definition 24** (Circuit Complexity Classes)**.** We define the class of *general circuits* to be all circuits where the function labels are taken from arbitrary arity $\{\wedge, \vee, \neg\}$, and define $\mathsf{SIZE}[t(n)]$ to be all functions computable by a family of general circuits of size $O(t(n))$. Similar to $\mathsf{P}$, we define $\mathsf{P/poly}$ as $\mathsf{P/poly} = \bigcup_{k\in\mathbb{N}} \mathsf{SIZE}[n^k]$.

We can also place other restrictions on the properties of the circuits, and define size classes in terms of those as well.

**Definition 25** ($\mathsf{AC}^0$)**.** We say a circuit is in $\mathsf{AC}^0_d$ if it is a general circuit of depth at most $d$, and for any two non-source vertices at the same depth, they are either both labelled $\wedge$ or both labelled $\vee$, possibly of differing arities. The $\mathsf{AC}^0_d$ size of a function $f$ is the size of the smallest family of $\mathsf{AC}^0_d$ circuits computing $f$. We defined $\mathsf{AC}^0_d$-$\mathsf{SIZE}[t(n)]$ to be the class of functions computable by size $O(t(n))$ families of $\mathsf{AC}^0_d$ circuits.

We will be analyzing a subclass of these $\mathsf{AC}^0$ circuits, where the in-degree of any gate neighbouring a source is bounded.

**Definition 26** (Bounded Bottom Fan-in)**.** A circuit is in $\mathsf{AC}^0_{d+1/2}$ if it is an $\mathsf{AC}^0_{d+1}$ circuit of size $s$ where all vertices neighbouring a source have in-degree at most $\log s$. We will also refer to these circuits as *bounded bottom fan-in* circuits.

For the case of $d = 2$, we have two special definitions.

**Definition 27** ($\mathsf{CNF}$ and $\mathsf{DNF}$)**.** A $\mathsf{CNF}$ is an $\mathsf{AC}^0_2$ circuit where the output gate is labelled $\wedge$, and a $\mathsf{DNF}$ is an $\mathsf{AC}^0_2$ circuit where the output gate is labelled $\vee$. The size of a $\mathsf{CNF}$ or $\mathsf{DNF}$ is the number of $\vee$ or $\wedge$ gates in the circuit, respectively.

**Definition 28** ($\mathsf{NC}$)**.** A general circuit is in $\mathsf{NC}^i$ if all vertices have in-degree at most 2, and the depth of the circuit is $O(\log^i n)$. We define $\mathsf{NC}$ as $\mathsf{NC} = \bigcup_{i\in\mathbb{N}} \mathsf{NC}^i$. The $\mathsf{NC}$ depth of a function $f$ is the smallest depth over any family of polynomially-sized $\mathsf{NC}$ circuits computing $f$ (or $\infty$ if no such family exists), and will say $f \in \mathsf{NC}^i$ if the $\mathsf{NC}$ depth of $f$ is $O(\log^i n)$.

Note that for $\mathsf{NC}^1$ in particular, all such circuits will be polynomially-sized. Moreover, we can also assume all such circuits are formulas, as duplicating gates with multiple out-neighbours only increases the size by a polynomial factor with such a low depth.

**Definition 29** (Tolerance)**.** For a complexity class $\mathsf{C}$, we say that a function $f$ is in $\widehat{\mathsf{C}}[\varepsilon]$ if there is a function $f' \in \mathsf{C}$ such that $\Pr_{x\sim\mathscr{U}}[f(x) \neq f'(x)] \leq \varepsilon$.

## 2.3 Minimum Circuit Size Problem

**Definition 30** ($\Lambda$-MCSP)**.** The *Minimum $\Lambda$-Circuit Size Problem* is the language $\Lambda$-MCSP defined by

$$\{(f,s) \mid f \in \Lambda\text{-}\mathsf{SIZE}[s]\},$$

where $f$ is given in the form of its truth-table.

When $\Lambda$ is unspecified, the problem is for general circuits.

**Definition 31** (Fixed-Parameter MCSP)**.** For a fixed $s \in \mathbb{N}$, $\Lambda$-MCSP$[s]$ is the language defined by

$$\{f \mid f \in \Lambda\text{-SIZE}[s]\}.$$

The difference between this definition and the one previous is that $s$ isn't given as part of the input, but is instead fixed ahead of time, similar to the difference between Clique and $k$-Clique.

**Definition 32** (Gap MCSP)**.** For fixed $s_{yes}, s_{no} \in \mathbb{N}$, $\Lambda$-GapMCSP$[s_{yes}, s_{no}]$ is the promise problem defined by

$$Y = \{f \mid f \in \Lambda\text{-SIZE}[s_{yes}]\}$$
$$N = \{f \mid f \notin \Lambda\text{-SIZE}[s_{no}]\}.$$

These can be combined with the tolerance operator $\widehat{\bullet}$ introduced above to obtain tolerant versions of all the above problems. For GapMCSP, we can also introduce a gap in the tolerance allowed, leading to the following definition.

**Definition 33** (Tolerant Gap MCSP)**.** For fixed $s_{yes}, s_{no} \in \mathbb{N}$ and $\varepsilon_1, \varepsilon_2 \in [0,1]$, $\widehat{\Lambda}[\varepsilon_1, \varepsilon_2]$-GapMCSP$[s_{yes}, s_{no}]$ is the promise problem defined by

$$Y = \{f \mid f \in \widehat{\Lambda}[\varepsilon_1]\text{-SIZE}[s_{yes}]\}$$
$$N = \{f \mid f \notin \widehat{\Lambda}[\varepsilon_2]\text{-SIZE}[s_{no}]\}.$$

We can restrict the problems further by only accepting (or in the case of GapMCSP, accepting or rejecting) functions over $n$ bits; we will denote this as $\Lambda$-MCSP$_n$, etc. We can then view our above definitions as the union of these fixed-bit versions where $n$ ranges over $\mathbb{N}$. By

doing this, we can also generalize our gap definition to allow for size *functions* instead of fixed size constants.

**Definition 34.** Given size functions $\alpha, \beta : \mathbb{N} \to \mathbb{N}$,

$$\Lambda\text{-}\mathtt{GapMCSP}[\alpha, \beta] = \bigcup_{n \in \mathbb{N}} \Lambda\text{-}\mathtt{GapMCSP}_n[\alpha(n), \beta(n)],$$

where for promise problems we define the union as being a promise problem where $Y$ and $N$ are defined as the unions of the constituent $Y$ and $N$ sets.

## 2.4 Communication Complexity

**Definition 35** (Rectangles)**.** A *rectangle* $R \subseteq X \times Y$ is a set that can be written as $X' \times Y'$ for $X' \subseteq X$ and $Y' \subseteq Y$.

**Definition 36** (Communication Protocol)**.** A *communication protocol* over $X \times Y \to Z$ is a rooted labelled proper binary tree where all leaves are given a label from $Z$, and all internal vertices are given a label from $\{\langle A, f : X \to \{0,1\}\rangle\} \cup \{\langle B, g : Y \to \{0,1\}\rangle\}$. The *communication complexity* of a protocol is the edge-length of the longest root-leaf path in the tree.

To evaluate a communication protocol on an input pair $(x, y)$, we recursively perform the following from the root: if the current vertex is a leaf, output its label. If it is not, evaluate $f(x)$ or $g(y)$, depending on the type of label the vertex was given, and proceed to the left child if the value is 0 or the right child otherwise. We say a protocol *computes* a function $f : X \times Y \to Z$ if for all pairs $(x, y)$, evaluating the protocol on $(x, y)$ outputs $f(x, y)$. For relations $R \subseteq X \times Y \times Z$, we say a protocol computes the relation if for all $(x, y)$ such that there is a $z$ for which $(x, y, z) \in R$, the protocol evaluates to some $z_P$ such that $(x, y, z_P) \in R$.

**Definition 37** (Communication Complexity)**.** The *communication complexity* of a function $f$ or relation $R$, denoted $D(f)$ or $D(R)$, is the minimum communication complexity of any protocol computing $f$ or $R$.

Similarly to circuit complexity, we can define families of protocols and talk about asymptotic communication complexity in terms of them.

**Definition 38** (Karchmer-Wigderson Games [63])**.** For a boolean function $f : \{0,1\}^n \to \{0,1\}$, the *Karchmer-Wigderson Game for $f$* is the relation $KW_f \subseteq \{0,1\}^n \times \{0,1\}^n \times [n]$ defined by

$$KW_f = \{(x,y,i) \mid f(x) = 1, f(y) = 0, x_i \neq y_i\}.$$

**Theorem 39** ([63])**.** *For any function $f$, $D(KW_f)$ is equal to the formula depth of $f$.*

## 2.5 Information Theory

We will slightly abuse notation, here and throughout, by referring to random variables from a distribution as being the distribution itself when no other variables are sampled from the same distribution.

**Definition 40** (Entropy)**.** The *entropy* of a distribution $X$ is the quantity

$$H(X) = \sum_{x \in X} -\Pr_X[x] \log\left(\Pr_X[x]\right).$$

The *joint entropy* of two distributions $X, Y$ is the quantity

$$H(X,Y) = \sum_{x \in X, y \in Y} -\Pr_{X,Y}[x,y] \log\left(\Pr_{X,Y}[x,y]\right).$$

**Definition 41** (Conditional Entropy)**.** Given two distributions $X, Y$, the *conditional entropy of $X$ given $Y$* is the quantity

$$H(X \mid Y) = H(X,Y) - H(X).$$

**Definition 42** (Mutual Information)**.** Given two distributions $X, Y$, the *mutual information*

between the two is the quantity

$$I(X;Y) = H(X) - H(X \mid Y).$$

**Definition 43** (Conditional Information)**.** Given three distributions $X, Y, Z$, the *mutual information between $X$ and $Y$ conditioned on $Z$* is the quantity

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z).$$

We will now give some basic equalities and relations relating to mutual information.

**Fact 44** (Symmetry)**.** $I(X;Y) = I(Y;X)$.

**Fact 45** (Non-negativity)**.** $I(X;Y \mid Z) \geq 0$.

**Fact 46** (Chain Rule for Mutual Information)**.** $I(X;Y,Z) = I(X;Y) + I(X;Y \mid Z)$.

**Corollary 47.** $I(X;Y) \leq I(X;Y,Z)$.

These will suffice to prove Theorem 104.

# Chapter 3

# Blockwise Switching Lemma

We will need a strengthening of Håstad's Switching Lemma [41] for the case of *structured* random restrictions that leave exactly one variable unset in every block of variables.

**Definition 48** (Blockwise Restrictions, $\mathscr{B}_n^l$)**.** A binary string of length $n \cdot l$ can naturally be divided into $n$ consecutive "blocks" of $l$ bits each. Variables $\{y_{i,j} : i \in [n],\ j \in [l]\}$ index into these strings. Denote by $\mathscr{B}_n^l$ the set of all restrictions $\rho$ that place **exactly one** $\star$ in each block of an $n$-block, $l$-block-size string. Formally, we have $\rho : [n] \times [k] \to \{0,1,\star\}$ and $\forall i \in [n]\ \exists! j \in [k]$ such that $\rho(i,j) = \star$.

**Lemma 49** (Blockwise Switching Lemma)**.** *Let $\varphi$ be a $k$-CNF on $n \cdot l$ variables. For any $s \geq 0$,* $\Pr_{\rho \sim \mathscr{B}_n^l}[\varphi \restriction_\rho$ *cannot be expressed as an $2^s$-term $s$-DNF$] \leq \left(\frac{8k}{l}\right)^s$.*

**Remark 50.** While the proof of Lemma 49 is actually slightly simpler than that of the standard Switching Lemma [85, 12], this Blockwise Switching Lemma implies the standard Switching Lemma (as stated in [12]). A uniformly random subset of $pn$ out of $n$ variables can be chosen as follows: Randomly uniformly permute the $n$ variables, then partition them into $pn$ consecutive disjoint blocks of size $1/p$ each, and, finally, randomly uniformly choose exactly one variable from each of the $pn$ blocks. For each fixed permutation of $n$ variables, Lemma 49 applies with $l = 1/p$. We get that the probability that a given $k$-CNF fails to simplify to an $s$-DNF when hit with a random restriction that leaves exactly $pn$ variables unset is upper-bounded by $(8pk)^s$.

We prove the Switching Lemma (Lemma 49) below, via a modification of the "compression" based proof of the Switching Lemma due to Razborov [15, 85, 12].

**Canonical Decision Trees & Notation.**

We write assignments to a set of variables $\{x_i | i \in [n]\}$ as functions $\alpha : [n] \to \{0,1\}$. A $k$-CNF $\varphi(x_1, \ldots, x_n)$ is a conjunction of $m$ clauses, where each clause is a disjunction over at most $k$ literals. A **Decision Tree** is a binary tree where nodes are labeled by variables $x_1, \ldots, x_n$, and leaves and edges are labeled by constants $\{0,1\}$.

To evaluate a Decision Tree on an assignment $\alpha$, begin at the root, labeled by some $x_i$. Move down the edge labeled by $\alpha(i)$. Repeat until you arrive at a leaf and report the constant labeling that leaf as the value of the tree.

Given a $k$-CNF $\varphi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$ we can create a ***Canonical*** **Decision Tree**. Fix a lexical ordering on variables and use it to sort and de-duplicate clauses; let $i \in [m]$ index the clauses of $\varphi$ in this sorted order. We define $\mathrm{CDT}(\varphi)$ recursively:

1. Transform $C_1 \in \varphi$ to a depth $\leq k$ tree $T$ querying all variables of $C_1$ in lexical order
2. for each branch $b$ of $T$:
   (a) follow $b$ to induce a partial assignment $\alpha_b$
   (b) $\varphi_b \leftarrow$ Simplify $\varphi / \alpha_b$
   (c) Case Analysis:
      i. $\varphi_b$ is empty: terminate $b$ with leaf labeled 1
      ii. $\varphi_b$ is falsified: terminate $b$ with leaf labeled 0
      iii. $\varphi_b$ is undetermined: extend $b$ with $\mathrm{CDT}(\varphi_b)$

A restriction is a *partial assignment*: a map $\rho : [n] \to \{0,1,\star\}$. The result of applying a restriction to a Boolean function $f$ is written $f \restriction_\rho$ where we substitute each occurrence of $x_i$ by $\rho(i)$ for every $\rho(i) \neq \star$. We will need to define restrictions that *extend* other restrictions. Let

$\mathscr{E}_D(\rho)$ denote the set of restrictions that are identical to $\rho$, except for replacing $D$ star locations with constants. Let $\mathscr{E}(\rho)$ denote the set of restrictions that replace *all* $\star$-locations of $\rho$ with constants. We will be concerned with the blockwise restrictions of Definition 48.

**Coding and Decoding Large-Depth Restrictions.**

Suppose all we know about $\rho$ is that it produces a large-depth canonical decision tree when applied to $\varphi$. We can witness this with some "long" path $\sigma$ through the tree. Our code will consist of a restriction $\tilde{\rho}_c$ that extends $\rho$ and a short bitstring "hint" that allows us to implicitly navigate "down" a long path of the CDT and guess $\rho$ by un-setting variables of $\rho_c$.

**Algorithm 1.** `ENC`

- Let $\sigma$ be a long path ($\geq$ depth $D$) through $T$
- For each clause along $\sigma$, $C_i^\sigma$:
  1. For each variable $\eta_{ij}$ appearing in $C_i^\sigma$, record a hint:
     (a) $\eta_{ij}$ as an index into $C_i^\sigma$ ($\log k$ bits)
     (b) Assignment to $\eta_{ij}$ along $\sigma$ (single bit)
     (c) Is this the *last* variable queried in $C_i^\sigma$? (single bit)
  2. Record $\tau_i$ as an assignment to $\eta_i$ that falsifies $\mathsf{C}_i^\sigma$
- $\rho_c = \rho \circ \tau_1 \circ \cdots \circ \tau_D$
- Return:
  1. $\tilde{\rho}_c \leftarrow \rho_c$ completed to a full assignment uniformly at random
  2. all hints concatenated together

**Claim 51** (Decoding from `ENC` output)**.** *Suppose $\rho \in \mathscr{B}_n^l$ fails to simplify a particular k-CNF $\varphi$, so that* $\mathrm{CDT}(\varphi \restriction_\rho) \geq D$. *Then,* $\Pr_{\tilde{\rho}_c \sim \mathtt{ENC}(\rho)}[\mathtt{DEC}(\tilde{\rho}_c) = \rho] \geq \left(\frac{1}{l}\right)^{(n-D)}$.

*Proof of Claim 51.* Fix $\rho \in \mathscr{B}_n^l$ and suppose $T = \mathrm{CDT}(\varphi \restriction_\rho)$ has depth $\geq D$. Let $\sigma$ be a witnessing path of length at least $D$ through $T$. We'll require some notation; denote by $C_i^\sigma$ the $i$th clause traversed along the path $\sigma$, in the sense that the recursive CDT construction worked on

**Algorithm 2.** DEC

1. Initialize: $\rho_1 \leftarrow \rho_c = \rho \circ \tau_1 \circ \cdots \circ \tau_D$ and $i \leftarrow 1$
2. for $i = 1$ to $D$
3. Simplify $\varphi_i \restriction_{\rho_i}$
4. Find first falsified clause of $\varphi_i = C_i^\sigma$
5. Read hint to find $\tau_i$ and $\sigma_i$ (stop-bit tells you when to stop).
6. $\rho_{i+1} \leftarrow \rho_i$ with $\tau_i$ replaced by $\sigma_i$ (so $\rho_{i+1} = \rho \circ \sigma_1 \circ \sigma_{i-1} \circ \sigma_i \circ \tau_{i+1} \ldots \tau_D$)
7. return $\rho_D$ with $\sigma_1 \circ \ldots \sigma_D$ unset, and $\star$'s *guessed* uniformly at random for all other blocks

clause $C$ to produce that section of the decision tree. Note, this may well be smaller than $m$, due to simplifications applied during construction of the CDT. Further, let $\sigma_i$ be the section of $\sigma$ that traverses $C_i^\sigma$ and let $\eta_i$ be the variables queried along $\sigma_i$. We can think of $\sigma_i$ as a sequence of assignments to these variables.

Now, consider the operation of DEC on $\tilde{\rho}_c \sim \mathtt{ENC}(\rho)$. First observe that no clauses of $\varphi$ were falsified by $\rho$ alone, by our assumption that $\mathrm{CDT}(\varphi \restriction_\rho) \geq D$ — a falsified clause would give a depth-1 decision tree with a single 0 leaf. Therefore, any falsified clause is due to variables set by some $\tau_i$ or a randomly set variable.

Because the CDT is constructed in lexical-clause-order and ENC follows this order, the *first* falsified clause of $\varphi \restriction_{\tilde{\rho}_c}$ must be $C_1^\sigma$. We wish to recover which variables $\tau_1$ set; the trick is that now we know they must reside in a uniquely identified clause of at most $k$ variables. So, we spend $\log k$ bits of the hint per variable to name *which* variables of $C_1^\sigma$ were along $\sigma$ and thus set in $\tau_1$.

Iterating this argument, we see that lexical ordering of the canonical decision tree ensures recovery of $D$ $\star$ locations of $\rho$. So, after running the main loop of DEC on $\tilde{\rho}_c$ we have a candidate that matches $\rho$ exactly in $D$ blocks. For each remaining block, DEC will simply guess at random which variable in the block was a $\star$ in $\rho$. Each block has $l$ bits, so we have a $(1/l)$ chance of

guessing correctly — that is, in agreement with the original location of the $\star$ in $\rho$. The number of blocks that must be guessed (instead of recovered using deterministic decoding, DEC) is $(n-D)$. Every guess must be correct to successfully decode $\rho$. This gives the claimed probability of decoding. □

Given instead a *random* completion $\rho_r$ of blockwise restriction $\rho$ and a *random* hint $h_r$, can any algorithm decode $\rho$? We can upper bound this probability.

**Claim 52** (Decoding from random information)**.** *For any algorithm $\mathscr{A}$, for every blockwise restriction $\rho$,* $\Pr_{\rho_r \sim \mathscr{E}(\rho)}[\mathscr{A}(\rho_r, h_r) = \rho] \leq \left(\frac{1}{l}\right)^n$.

*Proof of Claim 52.* The hint $h_r$ is clearly useless, because it is a random string. Furthermore, the random variables $\rho_r$ and $\rho$ are conditionally independent, given that $\rho_r$ is a randomly sampled completion of $\rho$. This means that observing $\rho_r$ provides *no information* regarding the $\star$-locations of $\rho$. Therefore, no algorithm can do better than to randomly guess which location, in each block, was a star, for every block of the received $\rho_r$. There are $n$ blocks of $l$ bits each and every guess must be correct, for the overall probability $(1/l)^n$. □

**Completing the proof.**

**Lemma 53** (Blockwise Switching Lemma)**.** *Let $\varphi$ be a k-CNF. Pick $\rho$ from $\mathscr{B}_n^l$ uniformly at random. Then* $\Pr[\mathrm{CDT}(\varphi \restriction_\rho) \geq D] \leq \left(\frac{8k}{l}\right)^D$.

*Proof.* We can lower-bound the probability of decoding from random completion $\rho_r$: if we are lucky enough that the randomly sampled completion agrees with $\rho_c$ in the "special" blocks set by ENC, then we can significantly narrow down the number of blocks whose $\star$ must be guessed at random! That is, the non-trivial probability of recovery for DEC can be exploited. Formally,

$$\begin{aligned}\left(\frac{1}{l}\right)^n &\geq \Pr[\mathtt{DEC}(\rho_r, h_r) = \rho] && \text{(by Claim 52)}\\ &\geq \Pr[\mathrm{CDT}(\varphi \restriction_\rho) \geq D] \times \Pr[h_r = h] \times \Pr[\rho_r \text{ extends } \rho_c] \times \Pr[\mathtt{DEC} \text{ decodes} \star\text{'s}]\end{aligned}$$

Taking each event in turn:

1. $|h| = D(\log(k)+2)$ so there are $(4k)^D$ possible strings. Flipping $h_r$ uniformly at random, $\Pr[h_r = h] = (4k)^{-D}$.

2. To extend $\rho_c$, the randomly chosen $\rho_r$ must agree in $D$ locations. One of these settings is correct, so $\Pr[\rho_r \text{ extends } \rho_c] = 2^{-D}$.

3. Given a correct hint and randomly completed $\rho_c$, the probability of DEC recovering $\rho$ is $\left(\frac{1}{l}\right)^{(n-D)}$ by Claim 51.

Plugging in, we get

$$\left(\frac{1}{l}\right)^n \geq \Pr[\mathrm{CDT}(\varphi \restriction_\rho) \geq D] \times (4k)^{-D} \times 2^{-D} \times \left(\frac{1}{l}\right)^{(n-D)}.$$

The proof of the lemma follows. □

Chapter 3, in part, is based on material as it appears in "Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for Constant-Depth Circuits and Applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

# Chapter 4

# Constant-Depth GapMCSP Reductions

The focus of this chapter is "hardness lifting" for circuits of depth $(d+1)$ to circuits of depth $(d+2)$, and its applications to GapMCSP for the respective classes. Theorem 56 shows how to lift hardness for bounded-fan-in $\mathsf{AC}^0$ circuits from depth $d$ to depth $d+1$ (also bounded fan-in). Here, a function of a not much larger size yet higher depth is constructed by replacing input variables of the original function by disjoint relatively small parities. This theorem is then applied to reduce GapMCSP for $\mathsf{AC}^0_d$ circuits to GapMCSP for $\mathsf{AC}^0_{d+1}$ circuits.

The reduction proceeds in three steps, with the middle step potentially repeated multiple times for a larger depth increase. The first step converts unbounded bottom fan-in circuits of depth $(d+1)$ to bounded (by log of the circuit size) fan-in circuits of the same depth, at the cost of increasing the size from $s$ to $2^{O(\sqrt{n \log n \log s})}$; see Corollary 55. This rebalancing only needs to be done once.

The second step, which relies on the hardness lifting theorem, is the quasi-polynomial time reduction from GapMCSP for bounded bottom fan-in circuits of depth $(d+1)$, to GapMCSP for bounded bottom fan-in circuits of depth $(d+2)$. The quasi-polynomial running time of this reduction comes from the blow-up in the size of the output truth table of the new function. Then, we show that for this setting GapMCSP for depth $d+1$ circuits reduces to GapMCSP for depth $d+2$ circuits (both bounded bottom fan-in), with a small loss in the gap size. See Theorem 57 for the exact statement.

The last step is a polytime reduction from GapMCSP for bounded bottom fan-in circuits of depth $(d+2)$, to GapMCSP for unbounded bottom fan-in circuits of the same depth; see Theorem 59.

## 4.1 Depth $d+1$ to $d+1/2$

**Lemma 54** (Fanin vs Size Tradeoff)**.** *For any $d \geq 3$, let $C$ be any depth-$d$ size-$s$ circuit over $n$ inputs. Then, for any $w \geq 1$, there is an equivalent depth-$d$ circuit $C'$ with bottom fan-in at most $w$, and the size at most $s^{(4n\log n)/w}$.*

*Proof.* Assume WLOG that all the bottom gates of $C$ are disjunctions. We will recursively define a decision tree $T$ such that each leaf $l$ is associated with a restriction $\rho_l$ resulting in $C\restriction_{\rho_l}$ having bottom fan-in at most $w$.

Initially, $T$ consists of a single leaf node corresponding to the empty restriction. While there is a leaf $v$ in $T$ corresponding to a restriction $\rho$ such that $C\restriction_\rho$ has some nonempty set $S$ of bottom gates of fan-in greater than $w$, do the following. Let $t = |S| \leq s$. Let $z$ be the literal that occurs in the most gates of $S$. Since there are more than $tw$ literal occurrences among the gates in $S$ and there are $2n$ literals, $z$ must appear in more than $(tw)/(2n)$ bottom gates. Branch on $z$, with the left child $v_1$ of $v$ corresponding to $z=1$, and the right child $v_0$ to $z=0$. Note that the restriction corresponding to $v_1$ satisfies all bottom gates containing $z$, and the restriction corresponding to $v_0$ reduces their fan-in by 1.

Every left branching we take in the decision tree results in $t$ shrinking by more than a factor $\left(1-\frac{w}{2n}\right)$. So after $k$ left branchings, there are fewer than $t\left(1-\frac{w}{2n}\right)^k$ large fan-in gates left. Setting $k = (2n/w)\ln s$, we have that after $k$ left branchings there are no large fan-in gates left. If $k \geq n/2$ (i.e., $w \leq 4\ln s$), then we can use the trivial upper bound $2^n$ on the size of $T$; note that, in this case, $2^n \leq 2^{4n\cdot(\ln s)/w}$, as required.

Otherwise, for $k < n/2$, we can upper-bound the size of $T$ as follows. Since each branch of $T$ is of length at most $n$, and it may contain at most $k$ left branchings, we get that the size of $T$

is at most

$$\sum_{r=0}^{k}\binom{n}{r} \leq k\cdot\binom{n}{k} \leq k\cdot\left(\frac{ne}{k}\right)^k \leq k\cdot\left(\frac{we}{2\ln s}\right)^k \leq 2^{(1.5)k\cdot\log(w/\ln s)} \leq s^{\frac{3n\log n}{w}},$$

where for the last inequality we used the definition of $k$ and the bound $(w/\ln s) \leq w \leq n$.

Suppose without loss of generality that the top gate of $C$ is a disjunction, i.e., $C = \bigvee_i \bigwedge_j g_{i,j}$. We can rewrite $C(x)$ as

$$\bigvee_{\text{leaves } l\in T} \left(\phi(x,\rho_l) \wedge C\restriction_{\rho_l}(x)\right), \tag{4.1}$$

where, for a fixed restriction $\rho_l$, the formula $\phi(x,\rho_l)$ indicates whether $x$ is consistent with $\rho_l$ (i.e., whether $x$ ends up at leaf $l$ of our decision tree $T$). It is easy to see that $\phi(x,\rho_l)$ can be written as a conjunction of at most $n$ literals.

As written, the circuit above is a depth-$(d+2)$ size at most $(1+|T|+|T|\cdot s)$ circuit with fan-in at most $w$. By distributivity, we can rewrite each $\phi(x,\rho_l)\wedge C\restriction_{\rho_l}(x)$ as $\bigvee_i\left(\phi(x,\rho_l)\wedge\bigwedge_j\left(g_{i,j}\restriction_{\rho_l}(x)\right)\right)$. Plugging this into equation (4.1), we obtain a depth-$d$ circuit $C'$ with fan-in at most $w$, computing the same function as $C$, and the size of $C'$ is at most $|T|\cdot s \leq s^{(4n\log n)/w}$, as required. □

**Corollary 55** (Depth $(d+1)\to(d+1/2)$)**.** *For any $d\geq 2$, $n$, and $s_{yes}, s_{no}$ such that $\log s_{yes} \leq \frac{\log^2(s_{no}/4)}{n\log n}$, we have*

$$\mathsf{AC}^0_{d+1}\text{-}\mathtt{GapMCSP}_n[s_{yes},s_{no}] \leq^{\mathsf{poly}}_m \mathsf{AC}^0_{d+1/2}\text{-}\mathtt{GapMCSP}_n[2^{4\cdot\sqrt{n(\log n)(\log s_{yes})}}, s_{no}]$$

*with the identity functions as a reduction.*

*Proof.* The "NO→NO" case is immediate: if $f\colon\{0,1\}^n\to\{0,1\}$ doesn't have size-$s_{no}$ circuits with no restriction on the bottom fan-in, then $f$ doesn't have size-$s_{no}$ circuits with restricted bottom fan-in.

For the "YES→YES" case, we apply Lemma 54 to a depth-$(d+1)$ size-$s_{yes}$ circuit for $f$, with $w = \sqrt{n(\log n)(\log s_{yes})}$. This results in a circuit for $f$ of size at most $2^{4\cdot\sqrt{n(\log n)(\log s_{yes})}}$, with bottom fan-in at most $\sqrt{n(\log n)(\log s_{yes})}$. □

## 4.2 Depth $d + 1/2$ to $(d+1) + 1/2$

**Theorem 56** (Hardness lifting)**.** *Let $f$ have $\mathsf{AC}^0_{d+1/2}$ circuit complexity $s$. Fix $s_0 > 0$. Then there is a function $f'$ on $n' = n \cdot 16\log s_0$ inputs with $\mathsf{AC}^0_{(d+1)+1/2}$ circuit complexity $s'$ where $s' \leq 2s2^{\sqrt{16\log s\log s_0}}\sqrt{16\log s\log s_0}$. Moreover, if $s_0 < \sqrt{s/3}$, then $s_0 \leq s'$.*

*Proof.* The construction is as follows: given the truth table of $f\colon \{0,1\}^n \to \{0,1\}$, output the truth table of $f' = f \circ \oplus_l$ for $l = 16\log s_0$. This takes time $2^{nl} = N^{16\log s_0} \leq N^{O(\log N)}$, quasi-polynomial in $N$ since $s_0 \leq N$. We argue the correctness next.

**Bounding $s'$ from below:**

Note that the parameter $l$ must be sufficiently larger than $\log s_0$ so that we can apply the Blockwise Switching Lemma to a depth-$(d+2)$ size-$s_0$ circuit with bottom fan-in $\log s_0$ that presumably computes $f \circ \oplus_l$ to obtain a depth-$(d+1)$ size-$s$ circuit with bottom fan-in $\log s$ that computes $f$. We prove that if $f'$ has a $\mathsf{AC}^0_{(d+1)+1/2}$ of size $s_0$, then $f$ has a $\mathsf{AC}^0_{d+1/2}$ circuit of size $s \leq 3(s_0)^2$.

Suppose $f \circ \oplus_l$ has a depth-$(d+2)$ circuit $C'$ of size $s_0$ and bottom fan-in at most $\log s_0$. We shall hit $C'$ with a blockwise random restriction $\rho$, where the blocks are the inputs to each $\oplus_l$. Since exactly one bit is left unset in each block, $C' \restriction_\rho$ computes $f$ with some of the input bits potentially negated. For $C' \restriction_\rho$ to simplify to a depth-$(d+1)$ circuit with bottom fan-in at most $k \leq \log(3s_0^2) \leq \log s$, we need to argue that there exists a blockwise restriction $\rho$ which makes every depth-2 bottom circuit of $C'$ into a decision tree of depth at most $k$. By the Blockwise Switching Lemma (Lemma 49), this is implied if $s_0\left(\frac{8\log s_0}{l}\right)^k < 1$, which is equivalent to $2^{\log s_0 - k} < 1$, for our choice of $l = 16\log s_0$. Thus, setting $k = \log s_0 + 1$ satisfies

this inequality. Moreover, each bottom CNF or DNF of $C'$ is turned into a DNF or CNF with $2^k$ clauses. So the size of $C' \restriction_\rho$ is at most $s_0 + s_0 \cdot 2^k \leq 3(s_0)^2 \leq s$, as required.

**Bounding $s'$ from above**

Next we need to show that if $f$ has a small depth-$(d + 1/2)$ circuit, then $f \circ \oplus_l$ has a small depth-$(d + 1 + 1/2)$ circuit. Note that computing the $l$-bit parities by naive depth-2 circuits of size $2^l$ is prohibitively expensive, as this would make the size of the new circuit for $f \circ \oplus_l$ at least $(s_0)^{16} > s'$, for our choice of $l = 16 \log s_0$ (which was dictated by the "NO→NO" case analysis above). Instead we will compute each $\oplus_l$ by a depth-3 circuit, as a parity of parities, adapting the standard construction of optimal size-$(l2^{\sqrt{l}})$ depth-3 circuits. To get a final circuit for $f \circ \oplus_l$ to be of depth $d + 1 + 1/2$, we will need to carefully balance the parameters of our partition of $l$ bits into $l_1$ blocks of size $l_2$ each, for $l_1$ and $l_2$ such that $l = l_1 \cdot l_2$.

Suppose $f$ has a depth-$(d+1)$ circuit $C$ of size $s$ and bottom fan-in at most $\log s$, with all negations at the leaves; this at most doubles the size. Without loss of generality, assume that the bottom layer of gates consists of disjunctions with fan-in $\log s$. To obtain a circuit for $f \circ \oplus_l$, we will compose $\oplus_l$ with each of the bottom CNFs of $C$. Consider a particular CNF $h_i = \bigwedge_{j=1}^{k} g_{i,j}$ at the bottom of $C$, where $k \leq s$ and each $g_{i,j}$ is a disjunction of at most $\log s$ literals.

For $l_1$ to be chosen later, let $l_2 = l/l_1$. Using the trivial $2^{l_1}$-size CNF for computing $\oplus_{l_1}$, we can compute each $g_{i,j} \circ \oplus_{l_1}$ by an OR-AND-OR circuit, where the top OR gate has fan-in $\log s$ and the AND gates each have fan-in $2^{l_1}$. By distributivity, we can rewrite $g_{i,j} \circ \oplus_{l_1}$ as a CNF with $2^{l_1 \log s}$ clauses, each of width at most $l_1 \log s$.

Since $C$ is a layered circuit, we can merge this CNF into $h_i$ to obtain a depth-2 circuit computing $h_i \circ \oplus_{l_1}$. Finally, composing this with the DNF for $\oplus_{l_2}$, we get a depth-3 circuit with bottom fan-in $l_2$ computing $h_i \circ \oplus_l$. Replacing each $h_i$ in $C$ with circuits constructed in this way, we obtain a depth-$(d+2)$ circuit for $f \circ \oplus_l$ with bottom fan-in $l_2$. The subcircuit for computing each $g_{i,j} \circ \oplus_l$ is of size at most $\sigma = 1 + 2^{l_1 \log s} + 2^{l_2} \cdot l_1 \log s$. So the total size of the circuit for $f \circ \oplus_l$ is at most $s + s \cdot \sigma = s(\sigma + 1)$. If we set $l_2 = \sqrt{l \log s}$ and $l_1 = \sqrt{\frac{l}{\log s}}$, then the total size

is at most

$$s\left(2+2^{\sqrt{l\log s}}+2^{\sqrt{l\log s}}\cdot\sqrt{l\log s}\right)\leq(2s)\cdot 2^{\sqrt{l\log s}}\cdot\sqrt{l\log s}\leq s'.$$

Since the bottom fan-in is at most $\sqrt{l\log s}\leq\log s'$, this concludes the proof. □

**Theorem 57** (Depth $(d+1/2)\to((d+1)+1/2)$ GapMCSP)**.** *For any $d\geq 1$, $n, s_{yes}$, $s'_{yes}$, $s'_{no}$, and $s_{no}$ such that $s_{yes}<s_{no}$, $s'_{yes}<s'_{no}$, $s_{no}\geq 3(s'_{no})^2$ and*

$$s'_{yes}\geq 2(s_{yes})2^{\sqrt{16(\log s_{yes})(\log s'_{no})}}\sqrt{16(\log s_{yes})(\log s'_{no})},$$

*we have*

$$\mathsf{AC}^0_{d+1/2}\text{-}\mathtt{GapMCSP}_n[s_{yes},s_{no}]\ \leq_m^{\mathsf{qpoly}}\ \mathsf{AC}^0_{(d+1)+1/2}\text{-}\mathtt{GapMCSP}_{n'}[s'_{yes},s'_{no}],$$

*where $n'=16n\log s'_{no}\in O(n^2)$.*

*Proof.* We use the construction in Theorem 56 as the reduction function, with $s_0=s'_{no}$. For the YES $\to$ YES side, if $AC^0_{d+1/2}(f)\leq s_{yes}$, then

$$\mathsf{AC}^0_{d+1+1/2}(f')\leq 2(s_{yes})2^{\sqrt{16(\log s_{yes})(\log s'_{no})}}\sqrt{16(\log s_{yes})(\log s'_{no})}$$

as desired. For the NO $\to$ NO side, if $AC^0_{d+1/2}(f)>s_{no}$, then $\mathsf{AC}^0_{d+1+1/2}(f')\geq s_0=s'_{no}$. □

**Remark 58.** If we apply this to succinct MCSP, we actually get a polytime reduction instead; constructing the naïve $f\circ\oplus_l$ circuit given a circuit for $f$ takes polytime, it just makes the truth table too large.

## 4.3 Depth $d+1/2$ to $d+1$

**Theorem 59** (Depth $(d+1/2)\to(d+1)$)**.** *For any $d\geq 1$, $n, s_{yes}, s_{no}, s'_{yes}, s'_{no}$, such that $s_{yes}<s_{no}$, $s'_{yes}<s'_{no}$, $s_{no}\geq(s'_{no})^5$ and $s'_{yes}\geq 2(s_{yes})^3$, we have*

$$\mathsf{AC}^0_{d+1/2}\text{-}\mathtt{GapMCSP}_n[s_{yes},s_{no}]\ \leq_m^{\mathsf{poly}}\ \mathsf{AC}^0_{d+1}\text{-}\mathtt{GapMCSP}_{2n}[s'_{yes},s'_{no}]$$

*Proof.* The reduction is as follows: given the truth table of $f\colon \{0,1\}^n \to \{0,1\}$, output the truth table of $g = f \circ \oplus_2$. The size of the input for $g$ is $2n$. The runtime of the reduction is $\mathsf{poly}(N)$. Next we argue the correctness of this reduction.

**NO $\to$ NO:**

Suppose $f \circ \oplus_2 \colon \{0,1\}^{2n} \to \{0,1\}$ is computable by a size-$s'_{no}$ circuit $C'$ of depth $d+1$. Without loss of generality, we may assume that the bottom gates of $C'$ are ANDs. We will hit $C'$ with a random blockwise restriction $\rho$. Consider a particular bottom AND-gate of fan-in $t$, for some $1 \leq t \leq n$. Since each block in a blockwise restriction is of size two, there must be at least $t/2$ variables from distinct blocks that feed into this AND gate. Each one of these variables will be chosen as a non-star variable by $\rho$ with probability $1/2$, and then independently set to 0 with probability $1/2$. This would simplify the AND gate to the constant 0, with probability $1/4$. This happens independently for each of these $t/2$ variables. Thus the probability that the AND gate of fan-in at least $t$ survives a random restriction is at most $(3/4)^{t/2}$. By the union bound, the probability that any such AND gate survives is at most $s'_{no} \cdot (3/4)^{t/2}$, which is less than 1 for $t = 5(\log s'_{no})$. Thus there exists a blockwise restriction $\rho$ which simplifies $C'$ to a depth-$(d+1)$ circuit computing $f$, with size at most $s'_{no} \leq s_{no}$ and bottom fan-in at most $5(\log s'_{no}) \leq \log s_{no}$.

**YES $\to$ YES:**

Suppose $f\colon \{0,1\}^n \to \{0,1\}$ is computable by a size-$s_{yes}$ circuit $C$ of depth $d+1$, with bottom fan-in at most $\log s_{yes}$. WLOG, assume the bottom gates of $C$ are ANDs. Note that we can express the XOR and the negated XOR of two variables as the following 2-CNFs:

$$y \oplus z = (\bar{y} \vee \bar{z}) \wedge (y \vee z) \qquad \text{and} \qquad \neg(y \oplus z) = (\bar{y} \vee z) \wedge (y \vee \bar{z}).$$

Replacing the input literals of $C$ by these circuits for (possibly negated) $\oplus_2$, and merging the bottom AND gate of $C$ with the top AND gate of these parity circuits, we get a depth-$(d+2)$ circuit $C'$ for $f \circ \oplus_2$, with 2-CNFs on $t = (2\log s_{yes})$ clauses as the bottom depth-2 sub-circuits.

By distributivity, we can rewrite each 2-CNF on $t$ clauses as a $t$-DNF on $2^t$ terms. Then merge the OR gates of these DNFs with the OR gates at the preceding level in $C'$, obtaining an equivalent depth-$(d+1)$ circuit $C''$ for $f \circ \oplus_2$, of size at most $s_{yes} + s_{yes} \cdot (s_{yes})^2 \leq 2(s_{yes})^3 \leq s'_{yes}$ (and bottom fan-in at most $(2 \log s_{yes}) \leq \log s'_{yes}$). □

## 4.4 Combining the steps: Depth $d+1$ to $d+c$ for any constant $c > 1$

The reduction in Theorem 57 can be repeated multiple times, resulting in the overall reduction lifting hardness to constantly many levels. The following theorem shows how the parameters evolve over all steps of the reduction.

**Theorem 60** (Depth $(d+1) \to (d+c)$)**.** *For any $d \geq 2$, $c > 1$, $n \geq n_0(\alpha, \delta, c)$, and $0 < \alpha < \delta < 1$ where $1 + \alpha < 2\delta$, we have*

$$\mathsf{AC}^0_{d+1}\text{-}\mathtt{GapMCSP}_n[2^{n^\alpha}, 2^{n^\delta}] \leq_m^{\mathsf{qpoly}} \mathsf{AC}^0_{(d+c)}\text{-}\mathtt{GapMCSP}_{n'}[2^{(n')^\beta}, 2^{(n')^\gamma}],$$

*where $n' = n^{(c-1)\delta+1}$, $\gamma \approx \frac{1}{c-1}$, $\beta \approx \frac{1}{c-1} - \frac{1}{(c-1)2^{c-1}} \cdot (1 - \frac{1+\alpha}{2\delta})$.*

*Proof.* As outlined at the beginning of the chapter, we will create this reduction via composing the reductions in Corollary 55 and Theorems 57 and 59. Let $a = \frac{1+\alpha+\frac{\log\log n+4}{\log n}}{2}$.

**Step 1:** $\mathsf{AC}^0_{d+1}\text{-}\mathtt{GapMCSP}_n[2^{n^\alpha}, 2^{n^\delta}] \leq_m^{\mathsf{poly}} \mathsf{AC}^0_{d+1/2}\text{-}\mathtt{GapMCSP}_n[2^{n^a}, 2^{n^\delta}]$

This follows immediately from Corollary 55 with $s_{yes} = 2^{n^\alpha}$ and $s_{no} = 2^{n^\delta}$.

**Step 2:** $\mathsf{AC}^0_{d+1/2}\text{-}\mathtt{GapMCSP}_n[2^{n^a}, 2^{n^\delta}] \leq_m^{\mathsf{qpoly}}$

$$\mathsf{AC}^0_{(d+c-1)+1/2}\text{-}\mathtt{GapMCSP}_{n^{(c-1)\delta+1}/2}[\exp_2\left(5n^{\frac{a+(2^{c-1}-1)\delta}{2^{c-1}}}\right), \exp_2(\tfrac{n^\delta}{2^{c-1}} - \tfrac{2^{c-1}-1}{2^{c-1}}\log 3)]$$

We will show each of $n$, $s_{yes}$, and $s_{no}$ map to the corresponding values after $c-1$ applications of the reduction in Theorem 57. Define $n^{(i)}$, $s^{(i)}_{yes}$, and $s^{(i)}_{no}$ to be each value after applying $i$ iterations of the reduction, with $n^{(0)}$, $s^{(0)}_{yes}$, and $s^{(0)}_{no}$ set to the initial values.

We will first show that $s_{no}^{(i)} = 2^{\frac{n^\delta}{2^i} - \frac{2^i-1}{2^i}\log 3}$; this is true for $i = 0$, and for larger $i$ we have $s_{no}^{(i+1)} = \sqrt{\frac{s_{no}^{(i)}}{3}} = 2^{\frac{n^\delta}{2^{i+1}} - \frac{2^i-1}{2^{i+1}}\log 3 - \frac{\log 3}{2}} = 2^{\frac{n^\delta}{2^{i+1}} - \frac{2^{i+1}-1}{2^{i+1}}\log 3}$.

Next, we show that $n^{(i)} = 16^i n \prod_{j=1}^{i} [\frac{n^\delta}{2^j} - \frac{2^j-1}{2^j}\log 3]$; via padding, we can increase the number of variables to $n^{(c-1)\delta+1}/2$ at the end. Again, this is true for $i = 0$. For larger $i$,

$$n^{(i+1)} = 16n^{(i)} \log s_{no}^{(i+1)} = 16^{i+1} n \prod_{j=1}^{i+1} \left[ \frac{n^\delta}{2^j} - \frac{2^j - 1}{2^j} \log 3 \right].$$

Finally, for $s_{yes}^{(i)}$, we show that after $i$ iterations of the Theorem 57 reduction, $s_{yes} = 2^{n^a}$, $s_{no} = 2^{n^\delta}$ would be mapped to at most $s'_{yes} = \exp_2(5n^{\frac{a+(2^i-1)\delta}{2^i}})$. For $i > 0$, assuming $s_{yes}^{(i)} \leq \exp_2(5n^{\frac{a+(2^i-1)\delta}{2^i}})$, we have $s_{yes}^{(i+1)}$ is at most

$$\exp_2\left(1 + 5n^{\frac{a+(2^i-1)\delta}{2^i}} + \sqrt{\frac{80}{2^i} n^{\frac{a+(2^i-1)\delta}{2^i}} (n^\delta - \Theta(2^i))} + O(\log n)\right) \leq \exp_2\left(5n^{\frac{a+(2^{i+1}-1)\delta}{2^{i+1}}}\right),$$

fixing $n_0$ sufficiently large. For $i = 0$, note that $n^a < 5n^{\frac{a+(2^0-1)\delta}{2^0}}$.

**Step 3:** $\mathsf{AC}^0_{(d+c-1)+1/2}$-$\mathtt{GapMCSP}_{n^{(c-1)\delta+1}/2}[\exp_2\left(5n^{\frac{a+(2^{c-1}-1)\delta}{2^{c-1}}}\right), \exp_2(\frac{n^\delta}{2^{c-1}} - \frac{2^{c-1}-1}{2^{c-1}}\log 3)] \leq_m^{\mathsf{poly}}$ $\mathsf{AC}^0_{d+c}$-$\mathtt{GapMCSP}_{n^{(c-1)\delta+1}}[2^{n^\beta}, 2^{n^\gamma}]$

This follows immediately from Theorem 59, setting

$$s_{yes} = \exp_2\left(5n^{\frac{a+(2^{c-1}-1)\delta}{2^{c-1}}}\right) \text{ and } s_{no} = \exp_2\left(\frac{n^\delta}{2^{c-1}} - \frac{2^{c-1}-1}{2^{c-1}}\log 3\right).$$

□

Chapter 4, in part, is based on material as it appears in "Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for Constant-Depth Circuits and Applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics*

*(LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

# Chapter 5

# Constant-Depth Tolerant GapMCSP Reductions

We will show an analogous "hardness lifting" reduction from the GapMCSP problem for average-case circuits of depth $d$ to depth $d+1$. In this average case setting, instead of applying the machinery of Lemma 54, we can make use of the observation that bottom gates of large fan-in are almost always equal to their bias; see Theorem 61. Thus we get smaller gaps on the output side of the reduction, at a small cost to the tolerance parameter.

## 5.1 Tolerant depth $d+1$ to $d+1/2$ and reverse

**Theorem 61** (Tolerant depth $(d+1) \to (d+1/2)$)**.** *For any $0 \leq \varepsilon_1, \varepsilon_2 < 1/2$, $d \geq 1$, $n \geq 1$, and $s_{yes} < s_{no}$, we have*

$$\widehat{\mathsf{AC}}^0_{d+1}[\varepsilon_1, \varepsilon_2]\text{-}\mathtt{GapMCSP}_n[s_{yes}, s_{no}] \leq^{\mathsf{poly}}_m \widehat{\mathsf{AC}}^0_{d+1/2}[\varepsilon_1 + 1/n, \varepsilon_2]\text{-}\mathtt{GapMCSP}_n[(s_{yes})^2, s_{no}]$$

*with the identity functions as a reduction.*

*Proof.* The "NO"→"NO" case is obvious. For the "YES"→"YES" case, suppose $C$ is a depth $d+1$ circuit of size $s_{yes}$ that disagrees with $f$ on at-most an $\varepsilon_1$-fraction of inputs. For each bottom gate of $C$ with fan-in larger than $2\log|C|$, replace the gate with a 1 if it is an OR, or a 0 if it is an AND. Call this new circuit with the replaced gates $C'$. For a uniformly-random

sampled input, any of the replaced gates would disagree with this bit with probability at most $|C|^{-2}$, and so the probability $C'$ disagrees with $C$ on a uniformly-random input is at most $1/|C|$, via a union bound. Since $|C| \geq n$, this is at most $1/n$, and so $C'$ disagrees with $f$ on at most an $(\varepsilon_1 + 1/n)$-fraction of inputs. Note that $|C'| \leq |C| \leq (s_{yes})^2$ and the bottom fan-in of $C'$ is at most $2\log s_{yes} \leq \log(s_{yes})^2$, as required. □

**Theorem 62** (Tolerant depth $(d+1/2) \to (d+1)$)**.** *For any $0 \leq \varepsilon_1, \varepsilon_2 < 1/2$, $d \geq 1$, $n \geq 1$, $s_{yes}, s_{no}, s'_{yes}, s'_{no}$ such that $s_{yes} < s_{no}$, we have, via the the identity functions as a reduction,*

$$\widehat{\mathsf{AC}}^0_{d+1/2}[\varepsilon_1, \varepsilon_2 + 1/n]\text{-}\mathtt{GapMCSP}_n[s_{yes}, s_{no}] \leq^{\mathsf{poly}}_m \widehat{\mathsf{AC}}^0_{d+1}[\varepsilon_1, \varepsilon_2]\text{-}\mathtt{GapMCSP}_n[s_{yes}, \sqrt{s_{no}}].$$

*Proof.* The "YES"→"YES" case is obvious. For the "NO"→"NO" case, let $C'$ be depth-$(d+1)$ circuit of size at most $s'_{no} = \sqrt{s_{no}}$ that $\varepsilon_2$-approximates $f$. As in the proof of Theorem 61 above, we replace by constants all bottom gates of $C'$ that have fan-in larger than $2\log|C'|$, getting a new circuit $C$ that computes $f$ on all but at most $\varepsilon_2 + (1/n)$ fraction of inputs. The size of $C$ is at most $s'_{no} \leq s_{no}$, and the bottom fan-in is at most $2\log s_{no}^{1/2} = \log s_{no}$, as required. □

## 5.2 Tolerant depth $d + 1/2$ to $(d+1) + 1/2$

**Theorem 63** (Tolerant depth $(d+1/2) \to ((d+1)+1/2)$)**.** *For any $d \geq 1$, $n \geq 1$, $0 \leq \varepsilon_1, \varepsilon_2 < 1/2$, $s_{yes}$, $s'_{yes}$, $s'_{no}$, and $s_{no}$ such that $s_{yes} < s_{no}$, $s'_{yes} < s'_{no}$, $s_{no} \geq 3(s'_{no})^2(\varepsilon_2 n + 1)$ and $s'_{yes} \geq 2(s_{yes})2^{\sqrt{16(\log s_{yes})(\log s'_{no})}}\sqrt{16(\log s_{yes})(\log s'_{no})}$, we have*

$$\widehat{\mathsf{AC}}^0_{d+1/2}[\varepsilon_1, \varepsilon_2 + 1/n]\text{-}\mathtt{GapMCSP}_n[s_{yes}, s_{no}] \leq^{\mathsf{qpoly}}_m \widehat{\mathsf{AC}}^0_{(d+1)+1/2}[\varepsilon_1, \varepsilon_2]\text{-}\mathtt{GapMCSP}_{n'}[s'_{yes}, s'_{no}],$$

*where $n' = 16n\log s'_{no} \leq O(n^2)$.*

*Proof.* We shall use the same reduction as in Theorem 57, outputting $f \circ \oplus_l$ on input $f$, where $l = 16\log s'_{no}$.

**NO $\to$ NO:**

Let $C'$ be a depth-$(d+2)$ circuit of size $s'_{no}$ and bottom fan-in at most $\log s'_{no}$ that $\varepsilon_2$-approximates $f \circ \oplus_l$. We shall hit $C'$ with a blockwise random restriction, as before. Here, we simultaneously require that $C' \restriction \rho$ simplifies to a depth-$(d+1)$ circuit with bounded bottom fan-in, and that its truth table is $(\varepsilon_2 + 1/n)$-close to (some fixed shift of) $f$.

For any $x \in \{0,1\}^n$ and a blockwise restriction $\rho$, we denote by $\langle x, \rho \rangle$ the $(n \cdot l)$-tuple of bits obtained by placing $x$ in the star positions of $\rho$. Clearly, picking $x$ and $\rho$ uniformly at random results in $\langle x, \rho \rangle$ being the uniform distribution on $\{0,1\}^{n \cdot l}$. By our assumption on $C'$, we have $\mathsf{Exp}_{x,\rho}\left[C'(\langle x, \rho \rangle) \neq (f \circ \oplus_l)(\langle x, \rho \rangle)\right] \leq \varepsilon_2$. By Markov's Inequality,

$$\Pr_{\rho}\left[\mathsf{Exp}_x\left[C'(\langle x, \rho \rangle) \neq (f \circ \oplus_l)(\langle x, \rho \rangle)\right] > \varepsilon_2 + \frac{1}{n}\right] < \frac{\varepsilon_2}{\varepsilon_2 + (1/n)}.$$

Hence, with probability at least $(\varepsilon_2 \cdot n + 1)^{-1}$, for a randomly chosen blockwise restriction $\rho$

$$\begin{aligned}\mathsf{Exp}_x\left[C'(\langle x, \rho \rangle) \neq (f \circ \oplus_l)(\langle x, \rho \rangle)\right] &= \mathsf{Exp}_x\left[C' \restriction_\rho (x) \neq (f \circ \oplus_l) \restriction_\rho (x)\right] \\ &= \mathsf{Exp}_x\left[C' \restriction_\rho (x) \neq f(x \oplus b^\rho)\right] \leq \varepsilon_2 + \frac{1}{n},\end{aligned}$$

for $b^\rho = b_1 \ldots b_n \in \{0,1\}^n$ such that $b_i$ is the parity of assigned values in the $i$th block of $\rho$.

So, if $C' \restriction_\rho$ fails to simplify with probability less than $(\varepsilon_2 \cdot n + 1)^{-1}$, then we are guaranteed there is some $\rho$ such that $C' \restriction_\rho (x)$ agrees with $f(x \oplus b^\rho)$, a shift of $f$, on all but at most $(\varepsilon_2 + (1/n))$-fraction of inputs $x \in \{0,1\}^n$, and is a depth-$(d+1)$ circuit with bounded bottom fan-in.

By the Blockwise Switching Lemma (Lemma 49), the probability that $C' \restriction \rho$ fails to simplify to depth $(d+1)$ circuit with bottom fan-in at most $k$ is at most $s'_{no}\left(\frac{8 \log s'_{no}}{l}\right)^k = 2^{\log s'_{no} - k}$, which is less than $(\varepsilon_2 \cdot n + 1)^{-1}$ if we choose $k = \log(2s'_{no}(1 + \varepsilon_2 n))$.

Thus, there must exist a blockwise restriction $\rho$ such that $C' \restriction_\rho$ is simplified and agrees with $f(x \oplus b^\rho)$ on all but at most $(\varepsilon_2 + (1/n))$ fraction of inputs. We have that $C' \restriction_\rho$ is of size at

most $s'_{no}(1+2^k) \leq s'_{no}(1+2s'_{no}(1+\varepsilon_2 n)) \leq 3(s'_{no})^2(1+\varepsilon_2 n) \leq s_{no}$. Also, the bottom fan-in is at most $k \leq \log s_{no}$ for our choice of $k$. Then the circuit $C(x) = C' \restriction_\rho (x \oplus b^\rho)$ agrees with $f(x)$ on all but at most $(\varepsilon_2 + (1/n))$ fraction of inputs, and $C$ has depth $(d+1)$, size at most $s_{no}$, and bottom fan-in at most $\log s_{no}$, as required.

**YES $\to$ YES:**

Suppose $f$ is $\varepsilon_1$-approximated by a depth-$(d+1)$ circuit $C$ with size $s_{yes}$ and bottom fan-in $\log s_{yes}$. Let $g$ be the Boolean function computed by $C$. Using the same techniques as in the "YES$\to$YES" case analysis in the proof of Theorem 57, we construct a depth-$(d+1)$ circuit $C'$ computing $g \circ \oplus_l$, with size at most $s'_{yes}$ and bottom fan-in at most $\log s'_{yes}$.

We will argue that $C'$ computes $f \circ \oplus_l$ on all but at most $\varepsilon_1$ fraction of inputs. Indeed, since the parity of a uniformly random string of bits is a uniformly random bit, we get that

$$\Pr_{z \in \{0,1\}^{nl}}[(f \circ \oplus_l)(z) = (g \circ \oplus_l)(z)] = \Pr_{x \in \{0,1\}^n}[f(x) = g(x)],$$

which is at most $\varepsilon_1$ by our assumption. This concludes the proof. □

## 5.3 Combining the steps: Tolerant depth $d+1$ to $d+2$

Using the above reductions, we can obtain a reduction from tolerant depth $d+1$ gap-MCSP to tolerant depth $d+2$ gap-MCSP. Extending this to depth $d+c$ can be done via repeatedly composing this reduction with itself.

**Corollary 64.** *For any $d \geq 1, 0 \leq \varepsilon_1, \varepsilon_2 < 1/2, s_{yes}, s_{no}, s'_{yes}$, $s'_{no}$ where $s_{no} \geq (2\varepsilon n+1){s'_{no}}^4$ and $s'_{yes} \geq 2{s_{yes}}^2 2^{\sqrt{16\log({s_{yes}}^2)\log({s'_{no}}^2)}} \sqrt{16\log({s_{yes}}^2)\log({s'_{no}}^2)}$, we have*

$$\widehat{\mathsf{AC}}^0_{d+1}[\varepsilon_1, \varepsilon_2 + \frac{2}{n}]\text{-}\mathtt{GapMCSP}_n[s_{yes}, s_{no}] \leq_m^{\mathsf{qpoly}} \widehat{\mathsf{AC}}^0_{d+2}[\varepsilon_1 + \frac{1}{n}, \varepsilon_2]\text{-}\mathtt{GapMCSP}_{32n\log s'_{no}}[s'_{yes}, s'_{no}].$$

*Proof.* We obtain the desired reduction by composing the reductions from Theorems 61, 63, and 62. Using $\langle \varepsilon_1, \varepsilon_2, s_{yes}, s_{no} \rangle_{d,n}$ as a shorthand for $\widehat{\mathsf{AC}}^0_d[\varepsilon_1, \varepsilon_2]\text{-}\mathtt{GapMCSP}_n[s_{yes}, s_{no}]$, the reduc-

tions operate as follows:

$$\begin{aligned}
\langle \varepsilon_1, \varepsilon_2 + \frac{2}{n}, s_{yes}, s_{no} \rangle_{d+1,n} &\mapsto \langle \varepsilon_1 + \frac{1}{n}, \varepsilon_2 + \frac{2}{n}, {s_{yes}}^2, s_{no} \rangle_{d+1/2,n} && \text{Theorem 61} \\
&\mapsto \langle \varepsilon_1 + \frac{1}{n}, \varepsilon_2 + \frac{1}{n}, s'_{yes}, {s'_{no}}^2 \rangle_{d+1+1/2,32n\log s'_{no}} && \text{Theorem 63} \\
&\mapsto \langle \varepsilon_1 + \frac{1}{n}, \varepsilon_2, s'_{yes}, s'_{no} \rangle_{d+2,32n\log s'_{no}} && \text{Theorem 62}
\end{aligned}$$

□

Chapter 5, in part, is based on material as it appears in "Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for Constant-Depth Circuits and Applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

# Chapter 6

# NP-hardness and Approximation Algorithms for bounded fan-in DNF-MCSP

As was done in [50], we would like to use depth-2 hardness for our bounded fan-in model to bootstrap our reductions to higher depths. While we can obtain hardness for bounded fan-in DNF-MCSP, we end up showing that approximating the size to within a factor of $n$ (which is much smaller than the gaps used in our reductions) is solvable in polynomial time. We begin with our hardness result, and introduce the following operator for combining multiple functions.

**Definition 65.** The *k-wise multiplexing* of $k$ partial binary functions $\{f_i : \{0,1\}^{n_i} \to \{0,1,*\}$ is the partial function $M_{f_1,\ldots,f_k}(x_1,\ldots,x_k,b_1,\ldots,b_k) : \{0,1\}^{\Sigma_i(n_i+1)} \to \{0,1,*\}$ defined by

$$M_{f_1,\ldots,f_k}(x_1,\ldots,x_k,b_1,\ldots,b_k) = \bigvee_{i=1}^{k} f_i(x_i) \wedge b_i,$$

where $* \wedge 1 = * \wedge * = *$, $* \wedge 0 = 0$, $* \vee 1 = 1$, and $* \vee 0 = * \vee * = *$.

Observe that if each input function is total, the resulting multiplexed function is also total. We will use this multiplexing operator to increase the DNF size required to compute a given input function, such that any DNF for the function can be regarded as a bounded fan-in DNF for the multiplexed function.

**Lemma 66.** *Let $f : \{0,1\}^n \to \{0,1,*\}$ be a partial function, and let $\oplus_1, \oplus_2, \oplus_3, \oplus_4 : \{0,1\}^n \to \{0,1\}$ be the n-bit parity function over 4 distinct sets of input variables. Then $f$ has a size m DNF*

*if and only if the 5-wise multiplexing $M_{f,\oplus_1,\ldots,\oplus_4}$ has a bounded fan-in DNF of size $m+4(2^{n-1})$.*

*Proof.* For the forward direction, let $\varphi = \vee_{i=1}^{k} C_i$ be a size $m$ DNF for $f$ and let $\psi_i$ be the standard DNF for $\oplus_i$ obtained by OR-ing together all satisfying assignments. To construct a DNF for the multiplexed function, take each $C_i$, and generate a new term $C_i' = C_i \wedge b_1$. Similarly, for each term of each $\psi_i$, add the corresponding $b_{i+1}$ to the term. Note that the resulting DNF is equivalent to

$$(\varphi \wedge b_1) \vee (\psi_1 \wedge b_2) \vee (\psi_2 \wedge b_3) \vee (\psi_3 \wedge b_4) \vee (\psi_4 \wedge b_5),$$

which is exactly the definition of $M_{f,\oplus_1,\ldots,\oplus_4}$. Each term of the DNF has width at-most $n+1$, whereas the size is $m+4(2^{n-1}) \geq 2^{n+1}$, so this DNF is actually a bounded fan-in DNF of the correct size.

For the reverse direction, suppose we have a minimum-sized bounded fan-in DNF $\varphi$ for the multiplexed function, with at-most $m+4(2^{n-1})$ terms. For succinctness, we'll rename the functions such that $g_1 = f$, and $g_{i+1} = \oplus_i$. First, we can assume that all occurrences of the $b_i$ variables are as positive literals, since $\varphi$ is monotonic in them. Second, note that every term must contain at least one $b_i$ literal, as otherwise fixing all the $b_i$ to 0 would still leave $\varphi$ as satisfiable. Moreover, each term must contain at most one $b_i$ as well. To see why, consider the restrictions $\rho_i$ which set $b_i = 1$ and the remaining $b_j = 0$. The claim is that any term which is fixed under every $\rho_i$ is redundant and can be removed. A term $\tau$ being redundant means there is no assignment $\alpha \in M^{-1}_{g_1,g_2,g_3,g_4,g_5}(1)$ under which $\tau$ is the sole satisfied term. For contradiction, let $\tau$ be a term which is fixed under each $\rho_i$. This means there is some multi-element set $I \subseteq \{1,2,3,4,5\}$ such that $\{b_i \mid i \in I\} \subseteq \tau$. Now suppose $\alpha$ is an assignment that makes $\tau$ non-redundant. Pick an $i \in I$ such that $g_i$ is not satisfied by $\alpha$; if all $g_i$ are, pick an arbitrary one. Since $\tau$ is satisfied by $\alpha$, we have that $b_i = 1$. If we now set $b_i = 0$, $\tau$ becomes falsified by the new assignment $\alpha'$. However, $\varphi(\alpha)$ remains 1, so some other term $\tau'$ must be satisfied by $\alpha'$. And since the $b_i$ only appear as positive literals, this $\tau'$ was also satisfied by $\alpha$, contradicting $\tau$ being the only term satisfied by $\alpha$.

With this, we can now partition the terms into disjoint sets $T_i$ according to which $b_i$ appears in them. Moreover, we have that $\varphi \restriction_{\rho_i} ([x_i],[b_i]) = g_i(x_i)$, and the terms in $T_i$ are the only non-fixed terms in $\varphi \restriction_{\rho_i}$. Thus, $|T_i| \geq \mathsf{DNF}(g_i)$, and in particular the four parity functions each require $2^{n-1}$ terms, leaving at-most $m$ terms for $f$. So $\varphi \restriction_{\rho_1}$ is a DNF computing $f$, and it contains at most $m$ terms. □

This multiplexed function also has other nice properties. Its truth-table length is polynomial in the truth-table length of $f$, and generating any bit of the multiplexed truth-table is a poly-time operation given the truth-table for $f$. Also, if $f$ is a total function, then as observed above the lemma, the multiplexed function is also total. This gives as immediate corollaries NP-hardness for the partial function and total function versions of MCSP for bounded fan-in DNFs.

**Corollary 67.** $\mathtt{MCSP}^*$ *for bounded fan-in* DNF*s is* NP*-hard.*

**Corollary 68.** MCSP *for bounded fan-in* DNF*s is* NP*-hard.*

Unfortunately for our hopes of bootstrapping, we can also show that GapMCSP is easy for a gap smaller than the minimum gap needed in the Chapter 4 reductions.

**Theorem 69.** $\mathtt{GapMCSP}_n[s,sn]$ *for bounded fan-in* DNF*s is solvable in polynomial time.*

*Proof.* We will use the $\log n$-factor approximation for Set Cover, on an $O(2^n)$-sized instance. Given an $n$-bit function $f$ as input, let the universe $U$ to cover be all strings in $f^{-1}(1)$. The set family $\mathscr{S}$ is obtained as follows. Iterate over all fan-in $\log s$ conjunctions over the $n$ input variables. For each conjunction, let $S$ be the strings it accepts. If all these strings are in $U$, add $S$ to $\mathscr{S}$, otherwise continue. Each $S$ can be generated in $O(2^n)$ time, and there are at-most $\sum_{i=0}^{\log s} \binom{n}{i} \leq 2^n$ such sets. After generating $U$ and $\mathscr{S}$, we run the greedy Set Cover approximation, and accept if and only if that approximation outputs something less than $sn$.

For correctness, if the bounded fan-in DNF complexity of $f$ is at-most $s$, then there is an $s$-cover of $U$ by sets in $\mathscr{S}$, obtained by taking all the sets which correspond to terms in the small

DNF for $f$. Each accepted input of $f$ must be accepted by at least one of those terms, and so the string will be contained in the corresponding set in $\mathscr{S}$. Similarly, if there is an $s$-cover of $U$ by sets in $\mathscr{S}$, then each set in that cover corresponds to a particular bounded fan-in term that only accepts strings in the 1-set of $f$, and all strings in the 1-set are accepted by some such term. So the disjunction over all the corresponding terms is a bounded fan-in DNF for $f$. □

# Chapter 7

# Barriers to More Efficient Natural Reductions

Our reductions are deterministic, many-one, and "simple" in the original size parameter. However, they require quasi-polynomial time. Here, we give evidence that improving such "nice" reductions to run in polynomial time for the *exact* MCSP is difficult: such reductions would immediately give breakthrough circuit lower bounds or non-trivial MCSP algorithms, and either outcome seems like dramatic progress.[1] To begin, observe that every reduction we present is qpoly-Natural in the following sense.

**Definition 70** (Natural Reductions between Parametric Problems)**.** Let $A$ and $B$ be *parametric problems,* that is, inputs are of the form: $\{\langle x,s\rangle : x \in \{0,1\}^n,\ s \in \mathbb{N}\}$. We call a *parametric reduction* $R = \langle R_I, R_P\rangle$ where $R_I$ outputs instances and $R_P$ outputs parameters, $t(\cdot)$*-natural* if it is:

- **Parametric Many-one:** $\langle x,s\rangle \in A \iff \langle R_I(x,s),\ R_P(x,s)\rangle \in B$
- **Parameter-Value Uniform:** $R_P(x,s)$ depends **only** on the size of the input and value of the parameter; we will treat $R_P$ as a function from $\mathbb{N}\times\mathbb{N}$ in this case.
- $t(\cdot)$**-Efficient:** The combined runtime of $R_I$ and $R_P$ is bounded by $t(|x|,s)$.

[1] Similar arguments apply to the gap-versions of the problem that we study above, but we argue about the exact version here to facilitate exposition.

A natural reduction $R$ from $\Lambda$-MCSP to $\Gamma$-MCSP is many-one, so a $\Lambda$-MCSP algorithm follows by brute-force search through $\Gamma$-circuits, and $\Lambda$-to-$\Gamma$ lifting follows by mapping a $\Lambda$-hard function $h$ through $R$. This gives the next two lemmas. Kabanets and Cai used the same reasoning to prove that NP-hardness of MCSP under poly-time natural reductions would imply breakthrough circuit lower bounds (Theorem 15 of [61]). Removing NP-hardness from the picture, we instead obtain the following:

**Lemma 71** (Black-Box MCSP Algorithms from Natural MCSP-Redux)**.** *If there is a* poly*-Natural Reduction from* $\Lambda$-MCSP *to* $\Gamma$-MCSP*, then there is a fixed constant* $k \in \mathbb{N}$ *such that* $\Lambda\text{-MCSP}_n \in \mathsf{TIME}[\mathsf{poly}(nk) \times \Gamma\text{-count}(R_P(2^n, s))]$

*Proof.* Fix a reasonable encoding of $\Gamma$-circuits that admits efficient evaluation. Then write $\Gamma$-count$(s)$ for the total number of circuits so encoded that witness $\Gamma$-measure at most $s$. On input $(f,s)$ to $\Lambda\text{-MCSP}_n$ we first run $(f,s)$ through the natural reduction $R$ to obtain $(f',s')$. Just as above, because $R$ is poly-time, there is a fixed $k$ such that $t(n) = 2^{kn}$. This means $|f'| \leq 2^{kn}$, so we obtain an instance of $\Gamma$-MCSP with new size parameter $s' = R_P(2^n, s)$ on at most $kn$ input variables.

Then, because $R$ is parametric many-one, a (yes, no)-instance of $\Lambda\text{-MCSP}_n$ becomes a (yes, no)-instance of $\Gamma\text{-MCSP}_{kn}$ (respectively). So, we can solve the resulting instance of $\Gamma$-MCSP by brute-force search over the set of all $s'$-measure-witnessing $\Gamma$-circuits, and answer accordingly. We must evaluate a $s'$-size $\Gamma$-circuit on $\leq kn$ bits at most $\Gamma$-count$(s')$ times. This takes $\mathsf{poly}(nk) \cdot \Gamma$-count$(s')$ time in total. □

Lifting begins with pre-existing lower bounds for $\Lambda$, which we formalize below. Many concrete circuit lower bounds are *far more* explicit, but this weak notion will suffice for lifting via natural and efficient inter-MCSP reductions.

**Definition 72** (Explicit Complexity Lower Bounds)**.** Let $H = \{h_n\}_{n \in \mathbb{N}}$ be a sequence of Boolean functions in E, and let $s_\Lambda : \mathbb{N} \to \mathbb{N}$ be a function in FP. We call the pair $\langle H, s_\Lambda \rangle$ an *explicit* $\Lambda$*-complexity lower bound* if $\forall n\ \Lambda(h_n) > s_\Lambda(n)$.

**Lemma 73** (Black-Box Lifting from Natural MCSP-Redux)**.** *Let $\langle H, s\rangle$ be a $\Lambda$-complexity lower bound. If there is a* poly*-Natural Reduction $R$ from $\Lambda$-*MCSP *to $\Gamma$-*MCSP*, then there exists a constant $k$ and sequence of $m$-input Boolean functions $H'$ such that $\langle H', R_P(2^{m/k}, s(m/k))\rangle$ is an explicit $\Gamma$-complexity lower bound.*

*Proof.* Fix an explicit $\Lambda$-complexity lower bound $\langle H, s\rangle$ and poly-natural reduction $R = \langle R_I, R_P\rangle$ from $\Lambda$-MCSP to $\Gamma$-MCSP. Now run the reduction: let $H'$ be the sequence $h'_n = R_I(h_n, s(n))$ and let $s'(n) = R_P(h_n, s(n))$. We know $(h_n, s(n)) \notin \Lambda$-MCSP by the hardness assumption about $H$. Then, because $R$ is parametric many-one, $(h'_n, s'(n)) \notin \Gamma$-MCSP and thus $\Gamma(h'_n) > s'(n)$. To make this explicit, we bound the runtime of answering queries according to $h'$ on inputs $x$ of $m$ bits. This amounts to re-indexing the sequence $H'$ to ensure that a $\Gamma$-hard function is defined everywhere and computable in E.

First, because $R$ is poly-time, there is a fixed $k$ such that $t(n) = 2^{kn}$. This means $|h'_n| \leq 2^{kn}$, so we send each input length $n$ through the reduction to a new input length of at most $kn$. We evaluate $h$ at $m/k$ input bits and pad to fill in the gaps. Propagating this padded sequence of functions through the parameter-map $R_P$, we obtained the claimed $\Gamma$-complexity lower bound. □

## 7.1 Efficient Natural Reductions Between $\mathsf{AC}^0_d$-, $\mathsf{AC}^0_{d+1}$-MCSP: Win/Win

Notice how both applications of poly-Natural reductions depend quantitatively on $R_P$, the size parameter of the reduction. For lifting, we want $R_P(\cdot)$ *large enough* to improve the best known $\Gamma$-complexity lower bound by starting with a stronger lower bound for $\Lambda$. For solving $\Lambda$-MCSP by brute-force on $\Gamma$-MCSP, we want $R_P(\cdot)$ *small enough* such that searching all relevant $\Gamma$-circuits is faster than trivial brute-force over all relevant $\Lambda$-circuits. This observation suggests a case analysis of the function $R_P$, to obtain either a non-trivial MCSP algorithm or improved circuit lower bounds. For poly-Natural reductions from $\mathsf{AC}^0_d$-MCSP to $\mathsf{AC}^0_{d+1}$-MCSP, such a win/win

argument succeeds. Informally, we have the following:

**Theorem 74** (poly-Natural MCSP Reduction Win/Win)**.** *Suppose there is a* poly*-Natural reduction from* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ *to* $\mathsf{AC}^0_{d'}$-$\mathtt{MCSP}$*, for* $d' > d$*. Then, either:*

- *There is a surprisingly fast algorithm for* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$*, or*
- *There are breakthrough explicit circuit lower bounds against* $\mathsf{AC}^0_{d'}[2^{\Omega(n^{1/d})}]$ *for* $d < d'$ *!*

We spend the remainder of this chapter formalizing and proving variations on the above.

## 7.2 Quantitative Consequences of a Hardness Hypothesis for MCSP

We first formulate an appropriate hypothesis about the hardness of $\mathtt{MCSP}$.

**Definition 75** (Weak Exponential Time Hypothesis (WETH) for $\Lambda$-$\mathtt{MCSP}$)**.** There exists an $\varepsilon > 0$ such that for all "nice" size functions $s(n)$, $\Lambda$-$\mathtt{MCSP}_n[s(n)] \notin \mathsf{TIME}[2^{s(n)^{\varepsilon}}]$.

For the general $\mathtt{MCSP}$ (when $\Lambda$ is the class of unrestricted Boolean circuits), it can be shown that the WETH for $\mathtt{MCSP}$ is implied by the cryptographic conjecture that exponentially-strong one-way functions exist (using the ideas of [86, 61, 5]). One can also show that if WETH for general $\mathtt{MCSP}$ is false, then $\mathsf{NEXP} \not\subset \mathsf{P/poly}$ (using the ideas of [53]). For every $d \geq 2$, the WETH for $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ is also reasonable to assume, although we don't seem to have any strong evidence to support it yet (see [7] for some cryptographic hardness of $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ for large $d$).

Under this hypothesis, we establish barriers to giving poly-Natural reductions from $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ to $\mathsf{AC}^0_{d+c}$-$\mathtt{MCSP}$. We begin by recalling the best-known $\mathsf{AC}^0_d$ circuit lower bounds.

**Theorem 76** (Håstad [41])**.** *Any depth* $(d+1)$ *alternating circuit computing* $\oplus_n$ *requires* $2^{\Omega(n^{1/d})}$ *gates. Furthermore, this bound is clearly* explicit *as in Definition 72.*

**Theorem 77.** *Suppose there is a* poly*-Natural reduction from* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ *to* $\mathsf{AC}^0_{d'}$-$\mathtt{MCSP}$*, for* $d' > d$*. Then, either:*

- *The WETH for* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ *is* ***false****, or*
- *There is an explicit circuit lower bound with* $s(n) = 2^{\Omega(n^{1/(d-1)})}$ *against* $\mathsf{AC}^0_{d'}$.

*Proof.* Assume such a poly-Natural reduction $R = \langle R_I, R_P \rangle$ exists, with run-time $2^{kn}$. We reason by cases on bounds for $R_P$.

Suppose $R_P$ is small. That is, $\forall \varepsilon. R_P(2^n, s(n)) < s(n)^\varepsilon$. Substituting into the black-box MCSP algorithm above, we have that $\forall \varepsilon. \mathsf{AC}^0_d\text{-}\mathtt{MCSP}_n \in \mathsf{TIME}[\mathsf{poly}(nk) \times \mathsf{AC}^0_{d'}\text{-}\mathsf{count}(s(n)^\varepsilon)] \in \mathsf{TIME}[2^{s(n)^{2\varepsilon}}]$, where the first inclusion is by Lemma 71, and second by counting $\mathsf{AC}^0_{d'}$ circuits. This contradicts the MCSP-WETH for $\mathsf{AC}^0_d$.

Suppose $R_P$ is large. That is, $\exists \varepsilon. R_P(2^n, s(n)) > s(n)^\varepsilon$. Lifting $\oplus$ through $R$ we have that there is an explicit sequence of Boolean functions $H$ on $m$-bit inputs such that we have the following explicit $\mathsf{AC}^0_{d'}$-complexity bounds: $R_P(2^{m/k}, s(m/k)) > s(m/k)^\varepsilon > 2^{\Omega(m^{1/(d-1)})}$. Here, the lower bound is by Lemma 73, the first inequality by size assumption about $R_P$, and the last by application of Håstad's bound. □

When $d' > d$, the lower-bound case above would be a breakthrough in circuit complexity.

**Corollary 78** (Breakthrough Circuit Lower Bounds for Alternating Constant-Depth)**.** *Suppose the WETH for* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ *holds, for every* $d \geq 2$*. Then, if* $\forall d > d_0$ *we have a* poly*-Natural reduction* $R_d$ *from* $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ *to* $\mathsf{AC}^0_{(d+1)}$-$\mathtt{MCSP}$*, then there is a fixed constant* $\alpha$ *such that, for each depth* $d > d_0$*, there is a Boolean function* $f^d \in E$ *such that any depth-*$d$ *alternating circuit computing* $f^d_n$ *requires* $2^{\Omega_d(n^\alpha)}$ *gates.*

*Proof.* Fix any constant $d > d_0$. We first compose $R_d$ with itself sufficiently many times to obtain a many-one reduction $R'_d$ all the way from $\mathsf{AC}^0_{d_0}$-$\mathtt{MCSP}$ to $\mathsf{AC}^0_d$-$\mathtt{MCSP}$. Observe that $R'_d$ remains poly-Natural, because all the polynomial resource bounds are closed under a constant number of compositions — though the leading constant exponent of runtime for $R'_d$ certainly increases proportional to the gap between $d$ and $d_0$; this is precisely what is hidden by $\Omega_d$ in the bound.

To conclude, we apply black box lifting (Lemma 73) to the composed poly-Natural reduction $R'_d$, with Håstad's lower bound for $\oplus$ at depth $d_0$, getting $\alpha = 1/d_0$ in the theorem. □

Combining with a simulation of shallow formulas by constant-depth circuits, we get

**Lemma 79** (Folklore)**.** *Any sequence $f_n$ of Boolean functions on n inputs computable by formulas of depth $c\log(n)$ is computable by depth-d alternating* circuits *of size $2^d \times 2^{n^{(c/d)}}$.*

**Theorem 80** (Breakthrough Circuit Lower Bounds for Formulas)**.** *Suppose the WETH for $\mathsf{AC}^0_d$-*MCSP *holds, for every $d \geq 2$. Then, if $\forall d > d_0$ we have a* poly*-Natural reduction $R_d$ from $\mathsf{AC}^0_d$-*MCSP *to $\mathsf{AC}^0_{(d+1)}$-*MCSP*, for every fixed k there exists $f^k$ a sequence of Boolean functions in* E*, such that $f^k$ does not have size-$n^k$ formulas.*

*Proof.* Fix constant $k$, and let $c \in \mathbb{N}$ be the leading constant that results from re-balancing an arbitrary $n^k$-size formula to log-depth. Any function computed by such a formula will have — for every $d$ — $\mathsf{AC}^0_d$ circuits of size $\approx 2^{n^{c/d}}$ by Lemma 79. Therefore, if we choose $d$ such that $1/d_0 > c/d$, the size bound that results from lifting $\oplus$ through iterated composition of $R_d$ exceeds the constant-depth simulation-size of any $n^k$-size formula. The rest of this argument is identical to the proof of Corollary 78 above. □

Chapter 7, in part, is based on material as it appears in "Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for Constant-Depth Circuits and Applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

# Chapter 8

# Half-Duplex Communication

**Definition 81.** Let $X$, $Y$ and $Z$ be some finite sets. We say that two players, Alice and Bob, are solving *half-duplex communication problem* for function $f : X \times Y \to Z$ if sets $X$, $Y$, $Z$ and function $f$ are known by both players, Alice is given some $x \in X$, Bob is given some $y \in Y$, and players want to compute the value of $f(x,y)$ by communicating to each other. The communication is organized in rounds. At every round, each player decides (depending only on its own input and previous communication) to do one of three available actions: send 0, send 1 or receive. If one player sends some bit $b \in \{0,1\}$ and the other one receives then the latter gets bit $b$, we call such rounds *normal*. If both players send bits at the same time then these bits get lost, we call such rounds *spent* (it is important that the player that is sending can not distinguish whether this round is normal or spent). If both players receive at the same time, we call such rounds *silent*. There are three variants of half-duplex communication problem depending on how silent rounds work.

- In a silent round both players receive *nothing*, so it is possible for both players to distinguish a silent round from a normal one, the corresponding problem is called *half-duplex communication problem with silence*.
- In a silent round both players receive 0, i.e., players cannot distinguish a silent round from a normal round where the other player sends 0, the corresponding problem is called *half-duplex communication problem with zero*;

- In a silent round each player receives some arbitrary bit, not necessarily the same as the other player; the corresponding problem is called *half-duplex communication problem with adversary*.

We say that half-duplex communication problem is *solved* if at the end of communication both players know $f(x,y)$.

Note that solving half-duplex communication problem with zero there is no need to send zeros — player can receive instead and the other player will not notice the difference.

**Definition 82.** *Half-duplex communication protocol with silence (with zero)* for function $f : X \times Y \to Z$ is a rooted tree that describes how Alice and Bob solve communication problem using half-duplex channel on all possible inputs. Every leaf $l$ of the protocol is labeled with $z_l \in Z$. Let $\mathscr{A} = \{\text{send } 0, \text{send } 1, \text{receive}\}$ be the set of possible actions. Every internal node $v$ of the protocol is labeled with three functions $g_v^A : X \to \mathscr{A}$, $g_v^B : Y \to \mathscr{A}$, and $h_v : \mathscr{A} \times \mathscr{A} \to C(v)$, where $C(v)$ is a set of child nodes of $v$. Root node corresponds to the initial state of communication. If the current state of communication corresponds to a node $v$, then Alice does action $g_v^A(x)$, Bob does action $g_v^B(y)$, and the next node is defined by $h(g_v^A(x), g_v^B(y))$.

The protocol definition for half-duplex communication problems with an adversary is a little bit more complicated.

**Definition 83.** *Half-duplex communication protocol with adversary* for function $f : X \times Y \to Z$ is a rooted tree that describes how Alice and Bob solves communication problem over half-duplex channel on all possible inputs and for any *strategy of adversary* $w \in \{0,1\}^*$. Every leaf $l$ of the protocol is labeled with $z_l \in Z$. Let $\mathscr{A} = \{\text{send } 0, \text{send } 1, \text{receive}\}$ be the set of possible actions, and $\mathscr{E} = \{\text{send } 0, \text{send } 1, \text{receive } 0, \text{receive } 1\}$ be the set of all possible events. Every inner node $v$ of the protocol is labeled with three functions $g_v^A : X \to \mathscr{A}$, $g_v^B : Y \to \mathscr{A}$, and $h_v : \mathscr{E} \times \mathscr{E} \to C(v)$, where $C(v)$ is a set of child nodes of $v$. Root node corresponds to the initial state of communication. If the current state of communication corresponds to a node $v$, then

Alice does action $g_v^A(x)$, Bob does action $g_v^B(y)$. If at least one of players decides to send then corresponding events are defined in a natural way. If both players decide to receive, i.e., this is a silent round, then Alice receives bit $w_{2i-1}$ and Bob receives bit $w_{2i}$. The next node of the protocol is defined by function $h$.

**Definition 84.** We say that half-duplex communication protocol *computes* function $f : X \times Y \to Z$ if for all $(x,y) \in X \times Y$, every leaf $l$ of the protocol labeled with $z_l$ corresponds to a state where both players know $z_l = f(x,y)$.

The arity of half-duplex communication protocols with silence and with zero is at most nine. The arity of half-duplex communication problems with adversary is at most 12: there are four possible events for each player, 16 options in total, but four of them are prohibited (e.g., if Alice sends 0 and Bob receives 1).

The classical communication complexity of a communication problem for function $f$, $D(f)$, is defined in terms of the minimal depth of a protocol solving it. Analogously, we define communication complexity for half-duplex communication problems.

**Definition 85.** The minimal depth of a communication protocol solving half-duplex communication problem for function $f$ with silence, with zero, with adversary, defines *half-duplex communication complexity* of function $f$ with silence, denoted $D_s^{hd}(f)$, with zero, denoted $D_0^{hd}(f)$, with adversary, denoted $D_a^{hd}(f)$, respectively.

We will focus on half-duplex communication complexity for a special case of Boolean functions $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ (i.e., $X = Y = \{0,1\}^n$, $Z = \{0,1\}$).

## 8.1 Trivial bounds

As far as half-duplex communication generalizes classical communication the following upper bound is immediate.

**Theorem 86.** *For every function* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$,

$$D_s^{hd}(f) \leq D_0^{hd}(f) \leq D_a^{hd}(f) \leq D(f).$$

*Proof.* Every classical communication protocol can be embedded in half-duplex communication protocol that does not use spent and silent rounds. □

Next theorem shows that every half-duplex protocol with zero or with adversary can be transformed in a classical communication protocol of double depth.

**Theorem 87.** *For every function* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$,

$$\frac{D(f)}{2} \leq D_0^{hd}(f) \leq D_a^{hd}(f).$$

*Proof.* Every $t$-round half-duplex communication protocol with silence or with adversary can be transformed into $2t$-round classical communication protocol. Every round of the original protocol corresponds to two consecutive rounds of the new one: at first round Alice sends a bit she was sending in the original protocol or sends 0 if she was receiving, at second round Bob does the same thing. □

As we will see later, half-duplex protocols with silence can use silent rounds as an additional third symbol and hence not every $t$-round half-duplex protocol with silence can be embedded in $2t$ classical protocol. The following theorem shows that instead we can embed every such protocol in a classical protocol with $3t$ rounds.

**Theorem 88.** *For every function* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$, $D_s^{hd}(f) \geq \frac{D(f)}{3}$.

*Proof.* Every $t$-round half-duplex communication protocol with silence can be transformed into $3t$-round classical communication protocol. Every round of the original protocol corresponds to three consecutive rounds of the new one: at first round Alice sends 1 to indicate if she was sending a bit in the original protocol, or sends 0 otherwise, at second round Bob does the same

thing symmetrically. After that they are both aware of the intentions of each other. If they were both planning to send, they can skip the third round. If they were both planning to receive, then they can just assume that they heard silence. If one player was planning to send and the other one was planning to receive they can perform such an action on third round. □

## 8.2 Rectangles

Many lower bounds on classical communication complexity were proved by considering combinatorial rectangles that are associated with the nodes of communication protocol [66]: it's easy to see that every node $v$ of the (classical) protocol corresponds to a combinatorial rectangle $R_v = X_v \times Y_v$, where $X_v \subseteq X$, $Y_v \subseteq Y$, such that if Alice and Bob are given an input from $R_v$ then their communication will necessarily pass through node $v$. This implies that the rectangles associated with the child nodes of $v$ define a subdivision of $R_v$.

There is a general technique [66] for proving lower bounds using associated combinational rectangles in: if for some sub-additive measure $\mu$ defined on combinatorial rectangles we show both

1. a lower bound on the measure of $X \times Y$, the rectangle in the root node, i.e., $\mu(X \times Y) \geq \mu_r$ for some $\mu_r$, and

2. an upper bound on the measure of rectangles in leaves, i.e., for every leaf $l$ the measure of the corresponding rectangle $R_l$ is at most $\mu_\ell$ for some $\mu_\ell$,

then we can claim lower bound of $\log_2(\mu_r/\mu_\ell)$ on the depth of the protocol.

One of the most studied sub-additive measure on rectangles is $\mu_M(R)$ that is equal to the minimal number of *monochromatic* rectangles that covers $R$. Rectangle $R$ is *z-monochromatic* in respect to function $f$ for some $z \in Z$ if for all $(x,y) \in R$, $f(x,y) = z$. As far as both players have to come up with the same answer at the end of communication every rectangle in leaves is monochromatic, thus for this measure $\mu_\ell = 1$.

Almost the same technique can be used for half-duplex protocols. There are some technical differences that we have to keep in mind. First of all, as we have already mentioned above, half-duplex protocol trees has different arities. Secondly, we should be careful while defining associated combinatorial rectangles for half-duplex protocols with adversary — in case of silent rounds the next node of the protocol depends also on a strategy $w$ of adversary, so we have to formally consider $w$ it as a part of input. This leads to the following lower bound for equality function $\mathrm{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, such that $\mathrm{EQ}_n(x,y) = [x = y]$.

**Theorem 89.**

- $D_s^{hd}(\mathrm{EQ}_n) > \log_9 2^n = n/\log 9$,
- $D_0^{hd}(\mathrm{EQ}_n) > \log_9 2^n = n/\log 9$,
- $D_a^{hd}(\mathrm{EQ}_n) > \log_{12} 2^n = n/\log 12$.

*Proof.* Let $\mu = \mu_M$. All rectangle in leaves are monochromatic, $\mu_\ell = 1$. Every 1-monochromatic rectangle is of size one: if some rectangle contains two elements, say $(x,x)$ and $(x',x')$, then it also contains $(x,x')$ and $(x',x)$, so it is not 1-monochromatic. Thus, the root rectangle has measure at least $\mu_r = 2^n + 1$ [66]. □

Unlike the classical communication in half-duplex communication players do not always know what was the other's player action — the information about it can be "lost" i.e., in spent rounds player do not know what was that other's player action. It means that a player might not know what node of the protocol corresponds to the current state of communication. Keeping this in mind, we can give an alternative definition of half-duplex protocols.

**Definition 90.** *Internal half-duplex communication protocol* for function $f : X \times Y \to Z$ is a pair $(T_A, T_B)$ of rooted trees that describe how Alice and Bob solve half-duplex communication problem on all possible inputs (and for any strategy of adversary $w \in \{0,1\}^*$). Every node of $T_A$ corresponds to a state of Alice, every node of $T_B$ — to a state of Bob. Every leaf $l$

is labeled with $z_l \in Z$. Let $\mathscr{A} = \{\text{send } 0, \text{send } 1, \text{receive}\}$ be the set of possible actions, and $\mathscr{E} = \{\text{send } 0, \text{send } 1, \text{receive } 0, \text{receive } 1\}$ be the set of all possible events. Every node $v$ of $T_A$ (of $T_B$) is labeled with two functions $g_v : X \to \mathscr{A}$ ($g_v : Y \to \mathscr{A}$) and $h_v : \mathscr{E} \to C(v)$, where $C(v)$ is a set of child nodes of $v$. Root nodes of $T_A$ and $T_B$ correspond, respectively, to the initial states of Alice and Bob. If Alice (Bob) is in a state that corresponds to node $v \in T_A$ ($v \in T_B$), then she does action $g_v(x)$ (he does action $g_v(y)$). The next node of the protocol is defined by the function $h$ (and also by strategy $w$ in case of silent round).

Trees $T_A$ and $T_B$ have smaller arity than protocol trees we defined earlier. In fact,

- arity is 5 for half-duplex communication with silence (send 0 or 1, receive 0 or 1, silence),
- arity is 3 for half-duplex communication with zero (send 1, receive 0 or 1),
- arity is 4 for half-duplex communication with adversary (send 0 or 1, receive 0 or 1).

For internal half-duplex protocols we still can define associated combinatorial rectangles and apply the same technique. This allows us to improve Theorem 89.

**Theorem 91.**

- $D_s^{hd}(\mathrm{EQ}_n) \geq \log_5 2^n = n/\log 5$,
- $D_0^{hd}(\mathrm{EQ}_n) \geq \log_3 2^n = n/\log 3$,
- $D_a^{hd}(\mathrm{EQ}_n) \geq \log_4 2^n = n/2$.

*Proof.* See the proof of Theorem 89. □

Surprisingly, as we will see later, first two result are sharp up to additive logarithmic term. We can get better bound if we improve this technique using *round elimination*.

### 8.2.1 Round elimination

Let us fix a protocol for some half-duplex communication problem and consider the first round. Let $R_c = X \times Y$ be the corresponding rectangle of all possible inputs. We can subdivide $R_c$ in nine rectangles, one for each possible combination of actions.

| **Alice\Bob** | send 0 | send 1 | receive |
|---|---|---|---|
| send 0 | $R_{00}$ | $R_{01}$ | $R_{0r}$ |
| send 1 | $R_{10}$ | $R_{11}$ | $R_{1r}$ |
| receive | $R_{r0}$ | $R_{r1}$ | $R_{rr}$ |

Consider two rectangles: $R_{good} = R_{00} \cup R_{01} \cup R_{0r}$ and $R_{bad} = R_{0r} \cup R_{1r}$. If we restrict $f$ to be a partial function defined only on $R_{good}$, i.e., players will always get some $(x,y) \in R_{good}$, then there is no need in the first round — the information the players get about the other part of the input is fixed: Alice does not get any information, Bob can receive 0 if he decide to receive. On the other hand if we restrict $f$ to $R_{bad}$ then the first round is still needed: Bob can receive both 0 and 1 and this information in necessary to proceed to the next round. Lets call a rectangle $R$ *good for functions* $f$ if restricting $f$ to $R$ makes the first round unnecessary (i.e., protocol without the first round is correct for all $(x,y) \in R$). The idea of this method is to consider some covering of $R_c$ with a set of *good* rectangles and prove that there is always a good rectangle of large enough measure. If we can show that there is always a rectangle of measure at least $\alpha \cdot \mu(R_c)$ then we can iterate this idea and claim that protocol depth is at least $\log_{1/\alpha}(\mu_r/\mu_\ell)$, where $\mu_r$ is a lower bound on the measure of the root rectangle and $\mu_\ell$ is an upper bound on the measure of leaf rectangles.

**Lemma 92.** *Let $\mu$ be some sub-additive measure on rectangles such that $\mu(X \times Y) \geq \mu_r$ and for any leaf rectangle $R_l$, $\mu(R_l) \leq \mu_\ell$. If for any rectangle $R$ there is always a good subrectangle for function $f \restriction R$ of measure at least $\alpha \cdot \mu(R)$ then the depth of the protocol is at least $\log_{1/\alpha} \frac{\mu_r}{\mu_\ell}$.*

*Proof.* We start with $R = X \times Y$. Every round restrict $f$ to some good $R_{good} \subseteq R$ such that

$\mu(R_{good}) \geq \alpha \cdot \mu(R)$, let $R$ to be $R_{good}$, and proceed to the next round. At the end we will reach some leaf. Thus there is at least $\log_{1/\alpha}(\mu_r/\mu_\ell)$ rounds. □

## 8.3 Half-duplex communication with silence

The main advantage of this model over the other models we consider is that whenever players have silent round, they learn about it. In some sense they have a third symbol in the alphabet — receiving player can get either $0/1$ or a special symbol corresponding to "silence". Next theorem shows how players can take the advantage of silence to transfer data.

**Theorem 93.** *For every* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $D_s^{hd}(f) \leq \lceil n/\log 3 \rceil + 1$.

*Proof.* Alice encodes $x$ in ternary alphabet $\{0,1,2\}$ and sends it to Bob: in order to send 0 or 1 Alice sends the corresponding bit, sending 2 is emulated by receiving (keeping silence). This requires $\lceil \log_3 2^n \rceil = \lceil n/\log 3 \rceil$ bits. At the last round Bob computes $f(x,y)$ and sends it back to Alice. □

Using the idea of encoding in a non-binary alphabet, we managed to prove a better upper bound for equality function.

**Theorem 94.** $D_s^{hd}(\mathrm{EQ}_n) \leq \lceil n/\log 5 \rceil + \lceil \log n/\log 3 \rceil + 2$.

*Proof.* Alice and Bob encode their inputs in alphabet of size five $\{0,1,2,3,4\}$. Then they process their inputs symbol by symbol sequentially in $\lceil n/\log 5 \rceil$ rounds. At round $i$ they process $i$th symbol in the following manner.

| Symbol | Alice | Bob |
|---|---|---|
| 0 | send 0 | receive |
| 1 | send 1 | receive |
| 2 | receive | send 0 |
| 3 | receive | send 1 |
| 4 | receive | receive |

If $i$th round is normal then one player can check whether $i$th symbols are different. If $i$th round is silent then again one player knows if $i$th symbols are different. If after $\lceil n/\log 5 \rceil$ rounds one of the players has already learned that the answer is 0, then he or she sends 0. If this round is not silent, then both players know that the answer is 0. Otherwise, Alice and Bob have to make sure that there were no spent rounds. In order to check it, Alice sends the number normal rounds she was receiving in encoded in ternary, that requires $\lceil \log n/\log 3 \rceil$ rounds. Bob checks whether this number is equal to the number of rounds he was sending in. If so, inputs are equal. In the last round, Bob sends the answer back to Alice. □

The next theorem shows better than $n/\log 3$ upper bound for disjointness function $\mathrm{DISJ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, such that $\mathrm{DISJ}_n(x,y) = \bigwedge_{i\in[n]} \neg(x_i \wedge y_i)$, which in classical case is one of the hardest functions of this type.

**Theorem 95.** $D_s^{hd}(\mathrm{DISJ}_n) \leq \lceil n/2 \rceil + 2$.

*Proof.* Alice and Bob process their inputs two bits per round, $\lceil n/\log 2 \rceil$ rounds. At round $i$ they process symbols $2i-1$ and $2i$ in the following manner.

| **Symbols** | **Alice** | **Bob** |
|---|---|---|
| 00 | send 0 | receive |
| 01 | receive | send 0 |
| 10 | receive | send 1 |
| 11 | receive | receive |

At the end of communication Bob tells Alice whether there was a silent round in which Bob's input was 11 (i.e., inputs are not disjoint). Alice tells Bob whether she ever received 0 having 01 or 11, or received 1 having 10 or 11 (again, inputs are not disjoint). □

The next function we have results for is the inner product function $\mathrm{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, such that $\mathrm{IP}_n(x,y) = \bigoplus_{i\in[n]} x_i y_i$. In the classical model, this function is one of the harder ones. This might also be the case for half-duplex models as the same time we do not

know efficient protocols for it, and this is the function we can prove the best lower bounds for. On the other hand, the best lower bound we can prove for it in this model is $n/2$.

**Theorem 96.** $D_s^{hd}(\mathrm{IP}_n) \geq n/2$.

For this theorem we need the following fact about inner product function.

**Lemma 97.** *Every leaf rectangle of a protocol solving communication problem for* $\mathrm{IP}_n$ *has size at most* $2^n$.

*Proof.* We start with proving it for leaves labeled with 0. Let $R_l = X_l \times Y_l$ be a rectangle of leaf $l$ labeled with 0, i.e., $R_l$ is 0-monochromatic. For every $x \in X_l$ and $y \in Y_l$, $\mathrm{IP}_n(x,y) = 0$, set $X_l$ must be contained in the orthogonal complement for span of $Y_l$. Thus, $\dim(\{X_l\}) + \dim(\{Y_l\}) \leq n$, and hence, $|R| = |X_l| \times |Y_l| \leq 2^n$.

If leaf is labeled with 1 then for every $x \in X_l$ and $y \in Y_l$, $\mathrm{IP}_n(x,y) = 1$. Let $y'$ be arbitrary element of $Y_l$. Consider a set $Y_l' = \{y \oplus y' \mid y \in Y_l\}$. It is easy to see that for every $x \in X_l$ and $y \in Y_l'$, $\mathrm{IP}_n(x,y) = 0$, so we can apply the argument above to show that $|X_l| \times |Y_l'| \leq 2^n$. It remains to notice that $|Y_l| = |Y_l'|$. □

*Proof of Theorem 96.* Let $R_c$ be the rectangle of all possible inputs and $\mu(R) = |R|$. Consider the following set of good rectangles: a rectangle $R_{silent} = R_{rr}$ where round is silent, four rectangles $R_{0*} = R_{00} \cup R_{01} \cup R_{0r}$, $R_{1*} = R_{10} \cup R_{11} \cup R_{1r}$, $R_{*0} = R_{00} \cup R_{10} \cup R_{r0}$, $R_{*1} = R_{01} \cup R_{11} \cup R_{r1}$, where one of players sends some bit, and a rectangle $R_{spent} = R_{00} \cup R_{01} \cup R_{10} \cup R_{11}$, where round is spent. We claim one of these good rectangles has measure at least $\mu(R_c)/4$.

For $\mu(R) = |R|$ we can use the following fact. Let $a_0$, $a_1$ and $a_r$ be the probability over all possible inputs that Alice sends 0, sends 1, and receives, respectively. Analogously, we define $b_0$, $b_1$ and $b_r$ to be the probability that Bob sends 0, sends 1, and receives. It is easy to see that $a_0 + a_1 + a_r = b_0 + b_1 + b_r = 1$ and for all $\alpha, \beta \in \{0,1,r\}$, $\mu(R_{\alpha\beta}) = a_\alpha \cdot b_\beta \cdot \mu(R_c)$.

We need to show that

$$\max\{\mu(R_{0*}), \mu(R_{1*}), \mu(R_{*0}), \mu(R_{*1}), \mu(R_{silent}), \mu(R_{spent}))\} \geq \mu(R_c)/4.$$

This is equivalent to showing that

$$\max\{a_1, a_0, b_1, b_0, a_r b_r, (1-a_r)(1-b_r)\} \geq 1/4$$

for any $a_0, a_1, a_r, b_0, b_1, b_r \in [0,1]$, such that $a_0 + a_1 + a_r = b_0 + b_1 + b_r = 1$. Let $\bar{a} = (a_1 + a_0)/2$, $\bar{b} = (b_1 + b_0)/2$. As far as $\max\{a_0, a_1\} \geq \bar{a}$ and $\max\{b_0, b_1\} \geq \bar{b}$,

$$\max\{a_1, a_0, b_1, b_0, a_r b_r, (1-a_r)(1-b_r)\} \geq \max\{\bar{a}, \bar{b}, a_r b_r, (1-a_r)(1-b_r)\}.$$

Note that $a_r + 2\bar{a} = 1$, $b_r + 2\bar{b} = 1$. Hence $\bar{a} = (1-a_r)/2$, $\bar{b} = (1-b_r)/2$,

$$\max\{\bar{a}, \bar{b}, a_r b_r, (1-a_r)(1-b_r)\} = \max\{(1-a_r)/2, (1-b_r)/2, a_r b_r, (1-a_r)(1-b_r)\}.$$

If $a_r \leq 1/2$ or $b_r \leq 1/2$ then one of first arguments is at least $1/4$. On the other hand if $a_r > 1/2$ and $b_r > 1/2$ then $a_r b_r > 1/4$. Now we apply Lemma 92 for $\mu_r = 4^n$, $\mu_\ell = 2^n$ (Lemma 97), $\alpha = 1/4$, and get the desired bound. □

## 8.4 Half-duplex communication with zero

As we have already mentioned before there are only two reasonable actions in this model: send 1 or receive. The following theorem shows that half-duplex communication with zero is more powerful than classical communication, namely, it is possible to solve communication problem for $\mathrm{EQ}_n$ in less than $n$ rounds of communication.

**Theorem 98.** $D_0^{hd}(\mathrm{EQ}_n) \leq \lceil n/\log 3 \rceil + 2\lceil \log n \rceil + 1$.

*Proof.* Alice and Bob encode their inputs in ternary. In the first phase of the protocol, they process their inputs sequentially symbol by symbol in $\lceil n/\log 3 \rceil$ rounds. At round $i$ they process $i$th symbol in the following manner.

| Symbol | Alice | Bob |
|---|---|---|
| 0 | receive | receive |
| 1 | send 1 | receive |
| 2 | receive | send 1 |

In the next $2\lceil \log n \rceil$ they send each other the number of ones they sent in the first phase. If inputs were different then one of players must have noticed it. At the first phase at round $i$ Alice learns if their corresponding symbols are $(0,2)$, $(2,0)$ or $(2,1)$, Bob learns if their symbols are $(0,1)$ or $(1,0)$. In the second phase, they can learn whether any of $(1,2)$ situation happened in the first phase. The last round players use to notify each other if somebody noticed a mismatch — in this case the player that noticed sends 1. □

Next theorem shows that there are functions of higher complexity than $\mathrm{EQ}_n$.

**Theorem 99.** $D_0^{hd}(\mathrm{IP}_n) \geq n/\log \frac{2}{3-\sqrt{5}} > n/\log 2.62$.

*Proof.* Let $R_c$ be the rectangle of all possible inputs and $\mu(R) = |R|$. Consider the following set of good rectangles: $R_{slilent} = R_{rr}$, $R_{spent} = R_{11}$, $R_{1*} = R_{11} \cup R_{1r}$ and $R_{*1} = R_{11} \cup R_{r1}$. We claim one of these good rectangles has measure at least $\frac{3-\sqrt{5}}{2} \cdot \mu(R_c)$. We need to show that

$$\max\{\mu(R_{1*}), \mu(R_{*1}), \mu(R_{silent}), \mu(R_{spent}))\} \geq \frac{3-\sqrt{5}}{2} \cdot \mu(R).$$

It is equivalent to showing that for any $a, b \in [0,1]$,

$$\max\{a, b, ab, (1-a)(1-b)\} \geq \frac{3-\sqrt{5}}{2},$$

where $a$ and $b$ denote the probabilities over all possible inputs that, respectively, Alice and Bob sends 1. It's easy to see minimum value of $\max\{a, b, ab, (1-a)(1-b)\}$ is at most $1/2$, so we can consider only $a \leq 1/2$ and $b \leq 1/2$. Thus,

$$\max\{a, b, ab, (1-a)(1-b)\} = \max\{a, b, (1-a)(1-b)\}.$$

Now we can argue that minimum of this max is achieved when $a = b = (1-a)(1-b)$: indeed, increasing or decreasing $a$ or $b$ increases one of the arguments. Solving corresponding quadratic equation $a = (1-a)^2$ we get $a = \frac{3-\sqrt{5}}{2}$, and hence

$$\max\{a, b, ab, (1-a)(1-b)\} \geq \frac{3-\sqrt{5}}{2}.$$

Applying Lemma 92 for $\mu_r = 4^n$, $\mu_\ell = 2^n$, and $\alpha = \frac{3-\sqrt{5}}{2}$ finishes the proof. □

## 8.5 Half-duplex communication with adversary

The main feature of this model is that receiving player can not be 100% sure that the received bit if in fact is "real", i.e., this bit originates from the other player, not from an adversary. But the protocol must be correct for any strategy of adversary. Our intuition prompts that in this setting silent and spent rounds would be useless. So we state a conjecture.

**Conjecture 100.** *There is function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that requires $n - o(n)$ rounds of half-duplex communication with an adversary.*

There is a common obstacle our methods faced when we were trying to prove this conjecture — it could be the case that players send different bits in spent rounds. For some reason, our methods do not work in this case which is strange because these spend rounds do not transmit any information. If we somehow forbid players to send different bits in spent rounds (e.g., in this case, we immediately terminate the communication and make players output 0) then we can prove that $\mathrm{EQ}_n$ requires $n$ rounds of communication. The same bound can be achieved if we allow such spent rounds only on distinct inputs. We suppose that this is an artifact of our methods and there is a way to overcome this obstacle. For unrestricted model, the best we can show is the following two theorems.

**Theorem 101.** $D_a^{hd}(\mathrm{EQ}_n) \geq n/\log 2.5$.

*Proof.* Let $R_c$ be the rectangle of all possible inputs and $\mu(R) = \big|\{(x,x) \in R\}\big|$. Consider the following set of 5 good rectangles:

$$R_{spent} = R_{00} + R_{01} + R_{10} + R_{11},$$

and four rectangles

$$R_{\bar{1}\bar{1}} = R_{00} \cup R_{0r} \cup R_{r0} \cup R_{rr}, \qquad R_{\bar{0}\bar{1}} = R_{10} \cup R_{1r} \cup R_{r0} \cup R_{rr},$$
$$R_{\bar{1}\bar{0}} = R_{01} \cup R_{0r} \cup R_{r1} \cup R_{rr}, \qquad R_{\bar{0}\bar{0}} = R_{11} \cup R_{1r} \cup R_{r1} \cup R_{rr},$$

where Alice does not send $\alpha$ and Bob does not send $\beta$ some fixed bits $\alpha, \beta$.

Now let us observe that together all these good rectangles cover the entire rectangle of possible input twice, and hence one of it has measure at least $2/5 \cdot \mu(R_c)$.

□

The last theorem of this section demonstrates the best known lower bound for this model.

**Theorem 102.** $D(\mathrm{IP}_n) \geq n/\log\frac{7}{3}$.

*Proof.* Let $R_c$ be the rectangle of all possible inputs and $\mu(R) = |R|$. We use a set of good rectangles consisted of rectangles $R_{spent}, R_{\bar{1}\bar{1}}, R_{\bar{0}\bar{1}}, R_{\bar{1}\bar{0}}, R_{\bar{0}\bar{0}}$ from the proof of Theorem 101 and four additional rectangles

$$R_{0*} = R_{00} \cup R_{01} \cup R_{0r}, \qquad R_{*0} = R_{00} \cup R_{10} \cup R_{r0},$$
$$R_{1*} = R_{10} \cup R_{11} \cup R_{1r}, \qquad R_{*1} = R_{01} \cup R_{11} \cup R_{r1},$$

where one of players sends some fixed bit. The following lemma shows that for this set of good rectangles and this specific measure we can prove a better bound.

**Lemma 103.** *For all half-duplex protocols with adversary*

$$\max\left\{\mu(R_{spent}),\mu(R_{0*}),\mu(R_{*0}),\mu(R_{1*}),\mu(R_{*1}),\mu(R_{\bar{1}\bar{1}}),\mu(R_{\bar{0}\bar{1}}),\mu(R_{\bar{1}\bar{0}}),\mu(R_{\bar{0}\bar{0}})\right\} \geq \frac{3}{7}\cdot\mu(R_c).$$

*Proof.* We use the idea we have already seen in the proof of Theorem 96. Let $a_0$, $a_1$ and $a_r$ be the probabilities over all possible inputs that Alice sends 0, sends 1 and receives, respectively. Analogously, we define $b_0$, $b_1$ and $b_r$ to be the probabilities that Bob sends 0, sends 1 and receives. It is easy to see that $a_0+a_1+a_r = b_0+b_1+b_r = 1$ and for all $\alpha,\beta \in \{0,1,r\}$, $\mu(R_{\alpha\beta}) = a_\alpha \cdot b_\beta \cdot \mu(R_c)$ (it is important here that $\mu(R) = |R|$). Minimization of maximum of linear functions with such constraints can be reduced to a semidefinite programming problem. Its solution gives us a desired bound. □

Application of the Lemma 92 for $\mu_r = 4^n$, $\mu_\ell = 2^n$ and $\alpha = 3/7$, finishes the proof. □

### 8.5.1 Upper-bound on internal information

A useful tool for proving lower bounds on the communication complexity of problems in the classical model is the upper bound on the information Alice and Bob have learned about the other's inputs, as a function of the number of rounds that have been run. Such tools allow for proving lower bounds, such as the $2\log n$-bit lower bound on the KW-game for parity.

**Theorem 104.** *Let $f$ be a partial function and $\mathscr{P}$ a half-duplex communication protocol with adversary computing $f$, and $\mathscr{D}$ an arbitrary distribution over the range of $f$. Let $\mathscr{X}$ and $\mathscr{Y}$ be the marginal distributions over inputs to Alice and Bob, and for any $k$ let $\Pi_A^k$ and $\Pi_B^k$ be the marginal distributions over Alice and Bob's partial transcripts after running $\mathscr{P}$ for $k$ rounds induced by $\mathscr{D}$, where on silent rounds the adversary picks whether to send 0 or 1 uniformly and independently at random for both players separately. Then for any $k$,*

$$I(\mathscr{X} : \Pi_B^k \mid \mathscr{Y}) + I(\mathscr{Y} : \Pi_A^k \mid \mathscr{X}) \leq k.$$

*Proof.* We will induct on $k$, the number of rounds that have been run. For $k = 0$, there is only one possible partial transcript for either player, the empty transcript, and thus the result is immediate. Now suppose that this is true in round $k$. Let $\mathscr{E}_A^{k+1}$ and $\mathscr{E}_B^{k+1}$ be the marginal distributions over which event each player will observe. Note that

$$\begin{aligned} I(\mathscr{X} : \Pi_B^{k+1} \mid \mathscr{Y}) &= H(\mathscr{X} \mid \mathscr{Y}) - H(\mathscr{X} \mid \mathscr{Y}, \Pi_B^{k+1}) \\ &= H(\mathscr{X} \mid \mathscr{Y}) - H(\mathscr{X} \mid \mathscr{Y}, \Pi_B^k) + H(\mathscr{X} \mid \mathscr{Y}, \Pi_B^k) - H(\mathscr{X} \mid \mathscr{Y}, \Pi_B^k, \mathscr{E}_B^{k+1}) \\ &= I(\mathscr{X} : \Pi_B^k \mid \mathscr{Y}) + I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y}, \Pi_B^k). \end{aligned}$$

Thus, it suffices to show that

$$I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y}, \Pi_B^k) + I(\mathscr{Y} : \mathscr{E}_A^{k+1} \mid \mathscr{X}, \Pi_A^k) \leq 1.$$

Let $(y, \pi_B^k)$ be a particular valid input-transcript pair for Bob. Consider $I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k)$; note that

$$\begin{aligned} I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) &\leq I(\mathscr{X}, \Pi_A^k : \mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) \\ &\leq H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) - H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k, \mathscr{X}, \Pi_A^k). \end{aligned}$$

Suppose Bob will be receiving in round $k+1$; otherwise

$$H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) = H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k, \mathscr{X}, \Pi_A^k) = 0.$$

Consider each $(x, \pi_A^k)$ input-transcript pair for Alice consistent with $(y, \pi_B^k)$. Note that $H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k, \mathscr{X} = x, \Pi_A^k = \pi_A^k)$ will either be 0, if Alice is sending a bit in round $k+1$, or 1, if she is receiving. The latter is because the adversary will choose whether Bob receives a 0 or 1

in round $k+1$ uniformly at random independent of Alice or Bob's transcripts or inputs. Thus

$$H(\mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k, \mathscr{X}, \Pi_A^k) = \Pr[\text{Alice receives} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k],$$

and thus

$$\begin{aligned} I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) &\leq 1 - \Pr[\text{Alice receives} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k] \\ &\leq \Pr[\text{Alice sends} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k]. \end{aligned}$$

We then have that

$$\begin{aligned} I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y}, \Pi_B^k) &= \sum_{(y,\pi_B^k)} \Pr[y, \pi_B^k] \cdot I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y} = y, \Pi_B^k = \pi_B^k) \\ &\leq \sum_{(y,\pi_B^k)} \Pr[\text{Alice sends}, \mathscr{Y} = y, \Pi_B^k = \pi_B^k] \cdot \mathbf{1}[\text{Bob receives}] \\ &\leq \Pr[\text{Alice sends, Bob receives}]. \end{aligned}$$

A symmetric argument holds for Alice, giving

$$\begin{aligned} &I(\mathscr{X} : \mathscr{E}_B^{k+1} \mid \mathscr{Y}, \Pi_B^k) + I(\mathscr{Y} : \mathscr{E}_A^{k+1} \mid \mathscr{X}, \Pi_A^k) \\ &\qquad \leq \Pr[\text{Alice sends, Bob receives}] + \Pr[\text{Alice receives, Bob sends}] \leq 1. \end{aligned}$$

□

As an immediate corollary we obtain a lower bound on the number of rounds needed to compute the Karchmer-Wigderson game for parity.

**Corollary 105.** *The Karchmer-Wigderson game for n-bit parity requires exactly* $2\log n$ *rounds of half-duplex communication with adversary.*

*Proof.* Take the uniform distribution over valid input pairs with a single bit of difference. Then

$$H(\mathscr{Y} \mid \mathscr{X}) + H(\mathscr{X} \mid \mathscr{Y}) = 2\log n$$

before any communication takes place, as $\mathscr{Y} \mid \mathscr{X}$ corresponds to the uniform distribution over the $n$ possible bit locations that are different, and

$$H(\mathscr{Y} \mid \mathscr{X}, \Pi_A) + H(\mathscr{X} \mid \mathscr{Y}, \Pi_B) = 0$$

at any leaf, as given each leaf corresponds to a single bit position, there is only one possible $(x, y)$ pair given $x$ or $y$. □

Chapter 8, in part, is based on material as it appears in "Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander V. Smal. Half-Duplex Communication Complexity. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation (ISAAC 2018)*, volume 123 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum für Informatik". The dissertation author was the primary investigator and author of that paper.

# Chapter 9

# Conclusions and Open Questions

We have given reductions from low-depth $\mathsf{AC}^0$-MCSP to high-depth $\mathsf{AC}^0$-MCSP for both bounded and unbounded bottom fan-in, demonstrated barriers to improving these reductions, and proven hardness for bounded fan-in DNF-MCSP. We have also given an approximation algorithm for the DNF case that prevents us from using it as the base case for bootstrapping hardness. On the communication complexity side, we have introduced the half-duplex model and its three variants, and proven tight upper and lower bounds for a variety of classic communication problems. We also demonstrated an upper bound on the information both parties exchange over the course of a protocol, for one of the variants.

One open question is whether we can reduce how much our depth-increase reduction shrinks the gap between the YES and NO sets of our GapMCSP problems. On the one side, we are leveraging a kind of "pick 2 out of 3" trade-off between size, fan-in, and depth for composing our parity circuits, and giving up size. We can't give up fan-in without finding an alternative to switching lemmas, and if we allow the depth to increase by 2 instead of 1, we would need to apply switching twice. This was attempted at one point while studying the reductions, and we could not go from $d+2$ to $d$ without losing the structure of $f$ in one of the two applications of switching. At this point, if we want to continue with the switching approach, it would be worthwhile to examine other choices of function than parity; perhaps the TRIBES function would be an interesting alternative?

On the other side, our switching lemma only examines the bottom DNFs/CNFs of the circuit one at a time. In [54], they present a switching lemma that looks at the probability of a collection of DNFs/CNFs over the same set of variables failing to simplify, with each contributing to the overall long CDT path. Such a global analysis could be useful for us as well, in allowing for multiple CNFs/DNFs in the switched circuit to share their clauses/terms.

Due to the depth $1+1/2$ case having efficient algorithms, any bootstrapping of hardness up from low depths to higher depths must be based on hardness at depth $2+1/2$ at minimum (assuming there aren't dramatic improvements in the minimum gap needed to run the reduction). Rather than immediately attempt to show depth $2+1/2$ is hard, we could instead look at even more restricted bottom fan-ins, e.g. bottom fan-in 2. Alternatively, we could try and base our hardness on ETH instead.

The WETH for $\mathsf{AC}^0_d$-$\mathtt{MCSP}$ seems like a reasonable assumption to make, being implied by one-way functions. It would be useful to give further arguments for/against this hypothesis. One possible avenue to explore would be tying it closer to one-way functions, as has already been done in the average case setting.

In the half-duplex model with adversary, demonstrating an explicit $n-o(n)$ lower bound remains an open question. The reason we cannot do this with our existing information upper bound is it refers to the information each party learns about the other's input. Therefore, if we want to say something like, e.g., "Take the uniform distribution over all inputs; at any leaf the rectangle can have at most $X$ elements. . . .", we need to ensure that these are leaves in each party's *local* protocol tree, not the global tree.

Another open question in the half-duplex models is characterizing the complexity of the universal composition relation, or $KW_{MUX}$. While this was the initial inspiration for studying half-duplex communication, we have no new results on that front so far. It would be interesting to see if the techniques from [33] generalize to this half-duplex setting, or if we can give non-trivial upper bounds on $KW_{MUX}$ within the half-duplex setting. One idea is relaxing the notion of "computing" a relation, i.e. allowing Alice and Bob to have different values of $z$.

# Bibliography


[1] Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*, 57(2):127–143, 1998.

[2] Miklós Ajtai. $\Sigma_1^1$-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[3] Eric Allender. The complexity of complexity. In Adam Day, Michael Fellows, Noam Greenberg, Bakhadyr Khoussainov, Alexander Melnikov, and Frances Rosamond, editors, *Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, pages 79–94. Springer International Publishing, Cham, 2017.

[4] Eric Allender. The new complexity landscape around circuit minimization. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications*, pages 3–16, Cham, 2020. Springer International Publishing.

[5] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006.

[6] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Computation*, 256:2–8, 2017.

[7] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and $AC^0$ circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008.

[8] Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. *ACM Transactions on Computation Theory (ToCT)*, 11(4):1–27, 2019.

[9] Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 21–33, 2015.

[10] Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size. In René van Bevern and Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2019.

[11] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. On the complexity of radio communication. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 274–285, New York, NY, USA, 1989. Association for Computing Machinery.

[12] Paul Beame. A switching lemma primer. 1994.

[13] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, s3-73(1):1–26, 1996.

[14] David Buchfuhrer and Christopher Umans. The complexity of boolean formula minimization. *Journal of Computer and System Sciences*, 77(1):142–153, 2011.

[15] Jin-Yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. *Journal of Computer and System Sciences*, 38(1):68–85, 1989.

[16] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation*, pages 75–85, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[17] Marco Carmosino, Kenneth Hoover, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Lifting for constant-depth circuits and applications to MCSP. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[18] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *31st Conference on Computational Complexity, CCC*, pages 1–24, 2016.

[19] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Agnostic learning from tolerant natural proofs. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 35:1–35:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[20] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *SIAM Journal on Computing*, 50(1):171–210, 2021.

[21] Lijie Chen and Hanlin Ren. Strong average-case circuit lower bounds from nontrivial derandomization. *SIAM Journal on Computing*, 51(3):STOC20–115–STOC20–173, 2022.

[22] Mahdi Cheraghchi, Valentine Kabanets, Zhenjian Lu, and Dimitrios Myrisiotis. Circuit lower bounds for MCSP from local pseudorandom generators. *ACM Transactions on Computation Theory*, 12(3), July 2020.

[23] Timothy Y. Chow. Almost-natural proofs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 86–91, 2008.

[24] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

[25] Alan Cobham. The intrinsic computational difficulty of functions. In Yehoshua Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science: Proceedings of the 1964 International Congress (Studies in Logic and the Foundations of Mathematics)*, pages 24–30. North-Holland Publishing, 1965.

[26] Stephen A. Cook. A hierarchy for nondeterministic time complexity. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, STOC '72, page 187–192, New York, NY, USA, 1972. Association for Computing Machinery.

[27] Sebastian Czort. The complexity of minimizing disjunctive normal form formulas. Master's thesis, University of Aarhus, 1999.

[28] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 24–30. IEEE, 2020.

[29] Anant Dhayal. *On Limiting & Limited Non-determinism in NEXP Lower Bounds*. PhD thesis, University of California, San Diego, 2021.

[30] Anand K. Dhulipala, Christina Fragouli, and Alon Orlitsky. Silence-based communication. *IEEE Transactions on Information Theory*, 56(1):350–366, 2010.

[31] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. *Computational Complexity*, 27(3):375–462, September 2018.

[32] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[33] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jirí Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001.

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

[35] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: The composition of a function and a universal relation. *SIAM Journal on Computing*, 46(1):114–131, January 2017.

[36] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $AC^0[p]$ lower bounds against MCSP via the coin problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 66:1–66:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[37] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM Journal on Computing*, 47(6):2435–2450, 2018.

[38] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 132–143, 2017.

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

[40] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[41] Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986.

[42] Johan Håstad. *Computational limitations for small-depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1987.

[43] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. *Journal of the ACM*, 64(5):35:1–35:27, 2017.

[44] Johan Håstad and Avi Wigderson. Composition of the universal relation. In Jin-Yi Cai, editor, *Advances In Computational Complexity Theory, Proceedings of a DIMACS Workshop, New Jersey, USA, December 3-7, 1990*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 119–134. DIMACS/AMS, 1990.

[45] Shuichi Hirahara, Igor C. Oliveira, and Rahul Santhanam. NP-hardness of minimum circuit size problem for OR-AND-MOD circuits. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2018.

[46] Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity, CCC*, pages 18:1–18:20, 2016.

[47] John M. Hitchcock and A. Pavan. On the NP-completeness of the minimum circuit size problem. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS*, pages 236–245, 2015.

[48] Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander V. Smal. Half-duplex communication complexity. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation (ISAAC 2018)*, volume 123 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[49] Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $AC^0[p]$. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:26, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[50] Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 424–433. IEEE, 2020.

[51] Rahul Ilango. The minimum formula size problem is (ETH) hard. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–432, 2022.

[52] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1575–1583, New York, NY, USA, 2022. Association for Computing Machinery.

[53] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

[54] Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for $AC^0$. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 961–972, USA, 2012. Society for Industrial and Applied Mathematics.

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

[56] Russell Impagliazzo and Ramamohan Paturi. Complexity of k-SAT. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*, pages 237–240, 1999.

[57] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[58] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '97, page 220–229, New York, NY, USA, 1997. Association for Computing Machinery.

[59] Russell Impagliazzo and Ryan Williams. Communication complexity with synchronized clocks. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 259–269, 2010.

[60] Valentine Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63(2):236–252, 2001.

[61] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.

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

[63] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.

[64] Gillat Kol and Ran Raz. Interactive channel capacity. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 715–724, New York, NY, USA, 2013. Association for Computing Machinery.

[65] Sajin Koroth and Or Meir. Improved composition theorems for functions and relations. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, volume 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[66] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[67] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973.

[68] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, July 1993.

[69] Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1243–1254, 2020.

[70] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM Journal on Computing*, 47(3):675–702, 2018.

[71] Oleg Lupanov. The synthesis of contact circuits. *Doklady Akademii Nauk SSSR*, 119(1):23–26, 1958. (In Russian).

[72] Oleg Lupanov. Implement the algebra of logic functions in terms of constant-depth formulas in the basis +, *, -. *Soviet Physics-Doklady*, 6(2), 1961.

[73] William J. Masek. Some NP-complete set covering problems. 1979.

[74] Or Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Computational Complexity*, 29(1), June 2020.

[75] Cody Murray and Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: An easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 890–901, New York, NY, USA, 2018. Association for Computing Machinery.

[76] Cody D. Murray and Ryan R. Williams. On the (non) NP-hardness of computing circuit complexity. In *30th Conference on Computational Complexity, CCC*, pages 365–380, 2015.

[77] Noam Nisan. CREW PRAMS and decision trees. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 327–335, New York, NY, USA, 1989. Association for Computing Machinery.

[78] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[79] Igor C. Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, Dagstuhl, DEU, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[80] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *Journal of the ACM*, 52(3):337–364, May 2005.

[81] Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth three boolean circuits. *Computational Complexity*, 9(1):1–15, Jan 2000.

[82] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 1207–1219, New York, NY, USA, 2018. Association for Computing Machinery.

[83] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1207–1219, 2018.

[84] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 234–243. IEEE, 1997.

[85] Alexander A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In Peter Clote and Jeffrey B. Remmel, editors, *Feasible Mathematics II*, pages 344–386, Boston, MA, 1995. Birkhäuser Boston.

[86] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[87] Henry G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.

[88] Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 721–730, New York, NY, USA, 2008. Association for Computing Machinery.

[89] Michael Saks and Rahul Santhanam. Circuit lower bounds from NP-hardness of MCSP under Turing reductions. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[90] Rahul Santhanam. Pseudorandomness and the Minimum Circuit Size Problem. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 68:1–68:26, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[91] Nathan Segerlind, Sam Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004.

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

[93] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[94] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[95] Christopher Umans, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Complexity of two-level logic minimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(7):1230–1246, 2006.

[96] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

[97] Ryan Williams. Nonuniform ACC circuit lower bounds. *Journal of the ACM*, 61(1), January 2014.

[98] Andrew C.-C. Yao. Some complexity questions related to distributive computing. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, page 209–213, New York, NY, USA, 1979. Association for Computing Machinery.